

# SILVERDEV

## RAD

**Copyright information**

The information contained in this document may be modified without notification and may not be considered binding in any manner upon **EXPERIA Europe**. Software packages are supplied under licence or confidentiality agreement. The software package may only be used, copied or reproduced onto any support in compliance with the terms of the said licence or confidentiality agreement. The buyer may only make copies for backup or archiving purposes.

No part of the manual or software package may be reproduced or transmitted via any means, electronic or otherwise, including photocopying, recording or any other storage, processing or information retrieval process for any purpose other than the buyer's personal use without express, written permission from **EXPERIA Europe**.

SilverDev is a registered trademark belonging to EXPERIA Europe.

IBM, AS/400, iSeries, System i, i5, Power i are registered trademarks belonging to International Business Machines Corporation.

Windows is a registered trademark belonging to Microsoft.

All other products are registered trademarks belonging to their respective companies.

[www.silverdev.com](http://www.silverdev.com)

EXPERIA Europe  
4, rue L.Beridot  
Les jardins d'Epione  
38500 Voiron

## Table des matières

---

### **Chapitre 1. Introduction ..... 5**

|                            |   |
|----------------------------|---|
| Automatic conversion ..... | 5 |
| Développer assistant.....  | 5 |
| Generator/context .....    | 5 |
| Models .....               | 5 |
| Wizard.....                | 5 |
| Screen conversion .....    | 5 |

---

### **Chapitre 2. Développer assistant..... 6**

|   |   |
|---|---|
| Drag and drop of file declaration .....     | 6 |
| Drag and drop of fields in rpg source ..... | 7 |
| Drag and drop fields in a form .....        | 8 |
| OptionsGeneration .....                     | 9 |

---

### **Chapitre 3. Modèles.....11**

|                        |    |
|------------------------|----|
| Model parameters. .... | 12 |
| Raw copy .....         | 14 |

---

### **Chapitre 4. Wizard .....16**

|  |    |
|--|----|
| Introduction. ....                                   | 16 |
| Création d'un programme de gestion de livres : ..... | 16 |
| Outil schéma .....                                   | 18 |
| Assistant base de données .....                      | 18 |
| Sauvegarde du schéma. ....                           | 19 |
| Paramétrage des tables .....                         | 20 |
| Créer une grille avec le Wizard .....                | 21 |
| Génération du source.....                            | 24 |
| Compilation du programme .....                       | 25 |
| Mise à disposition du programme.....                 | 25 |
| Exécution du programme .....                         | 28 |
| Wizard Sélection .....                               | 29 |
| Créer une fiche avec le Wizard .....                 | 31 |
| Pour aller plus loin : afficher une image .....      | 34 |
| Modifier le programme généré .....                   | 36 |
| Protection contre l'écrasement. ....                 | 36 |
| Outils.....  | 36 |

---

### **Chapitre 5. Screen conversion .....38**

|                                  |    |
|----------------------------------|----|
| Launch the conversion tool ..... | 38 |
| Add 5250 screens .....           | 38 |
| Record list .....                | 39 |
| Record Tabsheets.....            | 41 |
| Components Grid.....             | 41 |
| Container.....                   | 42 |
| 'Program to system' fields ..... | 43 |
| Components tabsheet .....        | 44 |
| Destination .....                | 45 |

|  |    |
|--|----|
| Dspf Size.....                                   | 45 |
| Dynamic properties .....                         | 46 |
| Links fields/Components/Dynamic properties ..... | 49 |
| 'Messages' tabsheet .....                        | 49 |
| Genération .....                                 | 50 |
| Save/Open .....                                  | 51 |
| Concrete cases .....                             | 51 |
| Several screens or several formats.....          | 51 |
| Indicators.....                                  | 55 |
| reference fields .....                           | 56 |
| function keys.....                               | 57 |
| Checkboxes.....                                  | 58 |
| Menus .....                                      | 62 |
| Subfiles .....                                   | 63 |
| Subfiles simple choice.....                      | 65 |
| Subfile multiple choice .....                    | 66 |
| window .....                                     | 67 |
| push buttons.....                                | 69 |
| Radio buttons .....                              | 72 |
| msgid/msgcon.....                                | 74 |

---

# Chapitre 1. Introduction

Silverdev contains several mecanism to accelerate development.  
These mecanism are the followings :

---

## Automatic conversion

Silverdev has a tool that allows to convert a 5250 program automatically. This Fonctionnality has its own document, « automatic conversion »

---

## Développeur assistant

Some fonctionnalités allows developpers to write code faster

---

## Generator/context

When using silverdev contexts, développeur writes code in QXTLESRC and final code with silverdev mecanism is generated in QRPGLSRC.

See chapter in document « Programming guide »

---

## Models

Allows to create programs automatically from customizable models

---

## Wizard

Allows to create automatically a simple program from a database description.

---

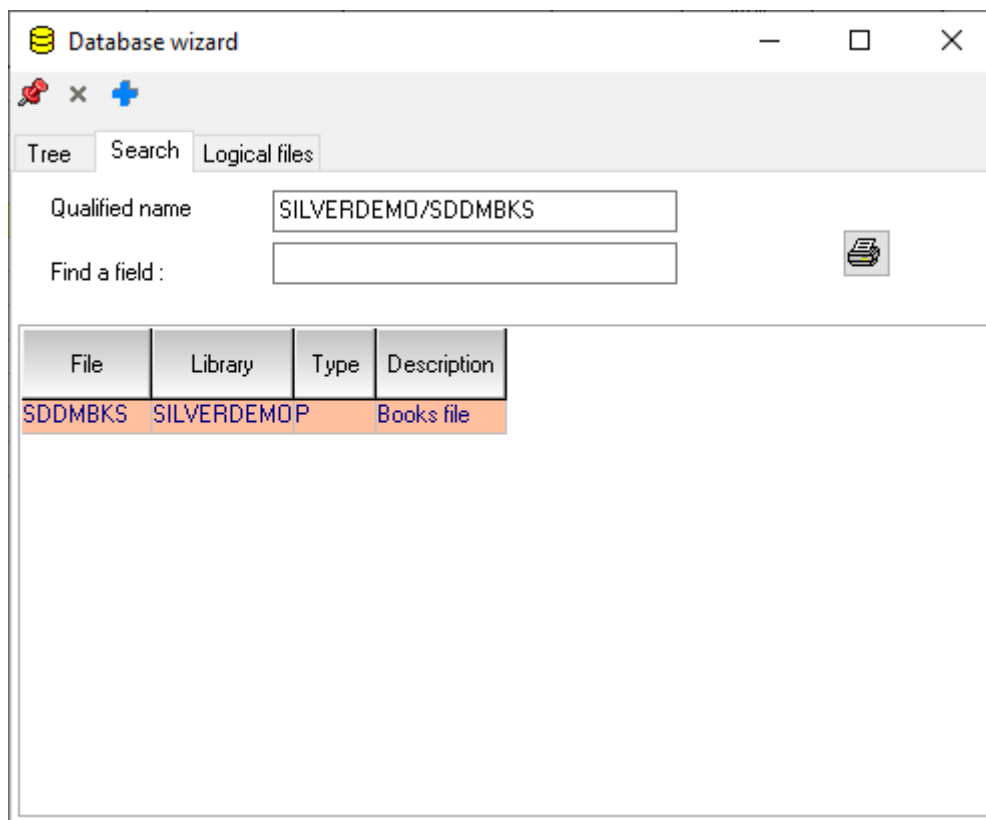
## Screen conversion

This fonctionnality allows to convert a 5250 screen in order to create a classic silverdev program based on this screen

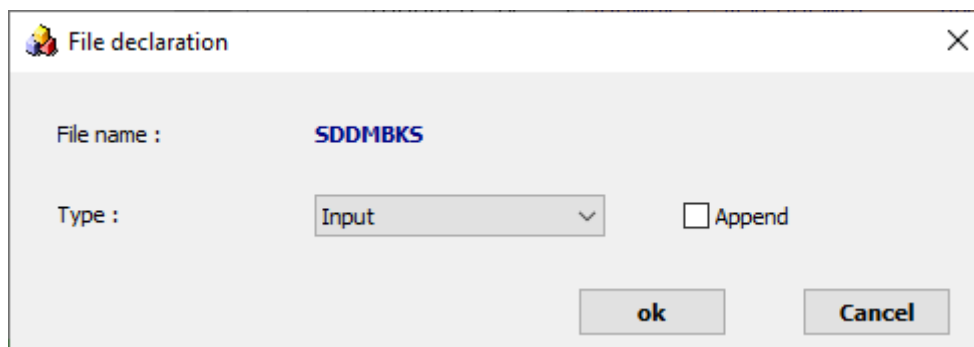
## Chapitre 2. Développeur assistant

### Drag and drop of file declaration

In database assistant, drag and drop a table name on the rpg source :



A dialog box is displayed :

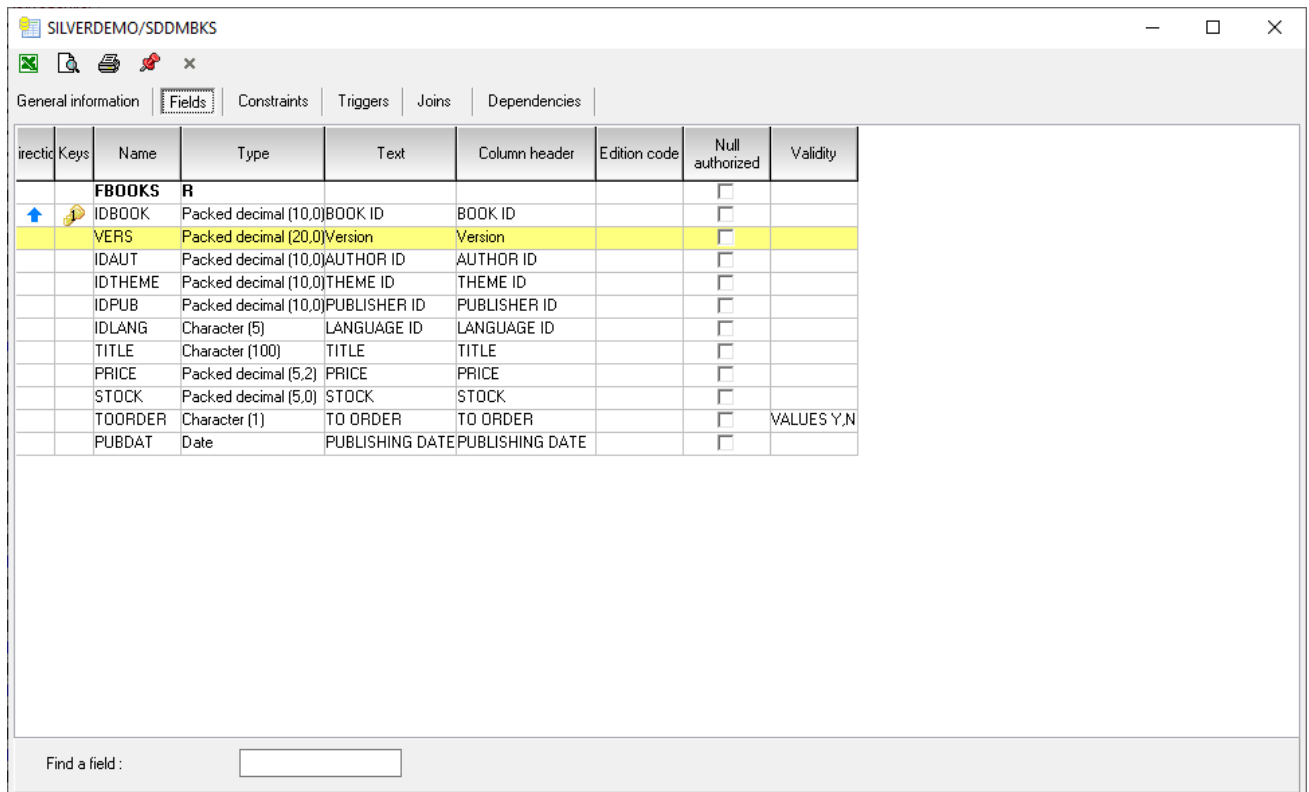


Declaration of the file is added :

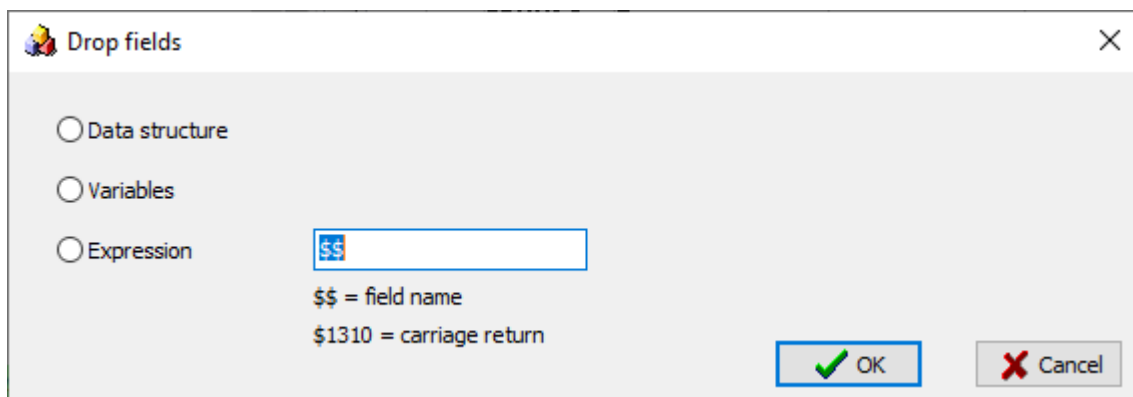
FSDDMBKS    uf A e                    k disk

## Drag and drop of fields in rpg source

Select fields in database assistant



Drag and drop on rpg code, a window is displayed :



Select one of the options. If you select expression option, you can customize the inserted expression

Example:

ds.\$\$ = \$\$;\$1310

will generate this code :

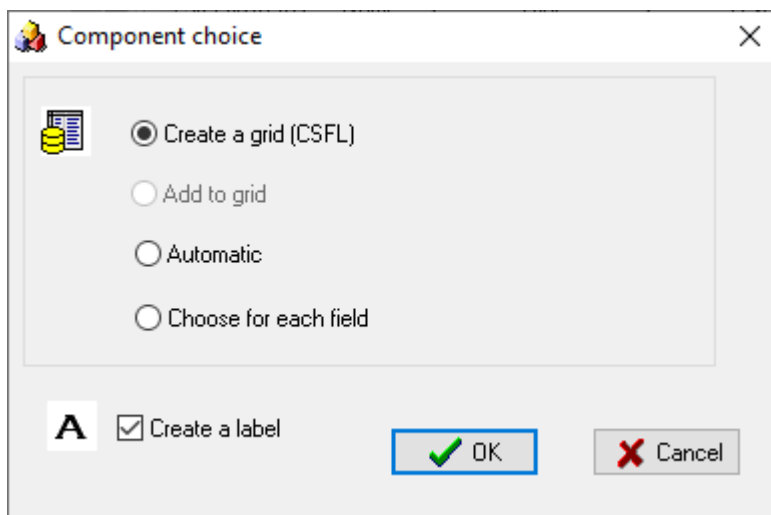
```
ds.IDBOOK = IDBOOK;  
ds.VERS = VERS;  
ds.IDAUT = IDAUT;  
ds.IDTHEME = IDTHEME;
```

---

## Drag and drop fields in a form

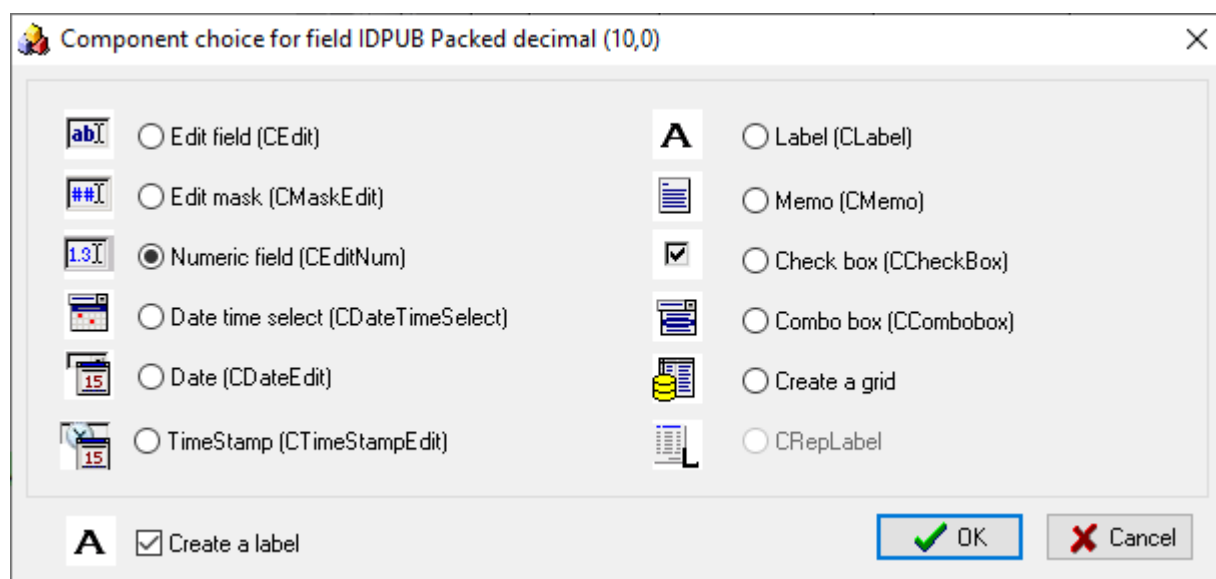
From database assistant, drag and drop fields on a form, a dialog box is displayed :

If you have selected several fields :



If you have selected only one field :





## OptionsGeneration

OptionsGeneration is a property that is specific to the generator.

This property is an object that has properties itself.

They differ according to the component.

|                    |  |
|--------------------|--|
| GenerateGet        | Determines whether or not the generator adds the code corresponding to this component to the read function.  |
| GenerateSet        | Determines whether or not the generator adds the code corresponding to this component to the write function.   |
| DefaultGenerateGet | Determines whether or not the generator adds the code corresponding to the subjacent components to the read function (or to the columns in the case of a CSFL).  |
| DefaultGenerateSet | Determines whether or not the generator adds the code corresponding to the subjacent components to the write function (or to the columns in the case of a CSFL). |

The values are set to true by default for all components, but to false for the window. The window's optionsGeneration properties must therefore be modified so that the generator can create read and write functions.

The generator creates the read and write instructions in specific procedures for all the components concerned (see the subjacent components). The name starts with WRITE or READ, followed by an underscore and the name of the parent component.

### For example:

```
d Write_SCREEN1 pr
```

```
d Read_SCREEN1      pr
```

```
p Write_SCREEN1      B
d                      PI
c                      Callp      sdSetString(F1
c                      : 'EDIT1 '
c                      : 'TEXT'
c                      :EDIT1)
c                      Callp      sdSetNum(F1
c                      : 'EDITNUM1 '
c                      : 'VALUE '
c                      :EDITNUM1)
p                      E
```

```
p Read_SCREEN1       B
d                      PI
c                      Eval        EDIT1=
c                      sdGet(F1
c                      : 'EDIT1 '
c                      : 'TEXT')
c                      Eval        EDITNUM1=
c                      sdGetNum(F1
c                      : 'EDITNUM1 '
c                      : 'VALUE ')
p                      E
```

This implies that fields Edit1 and Editnum1 exist.

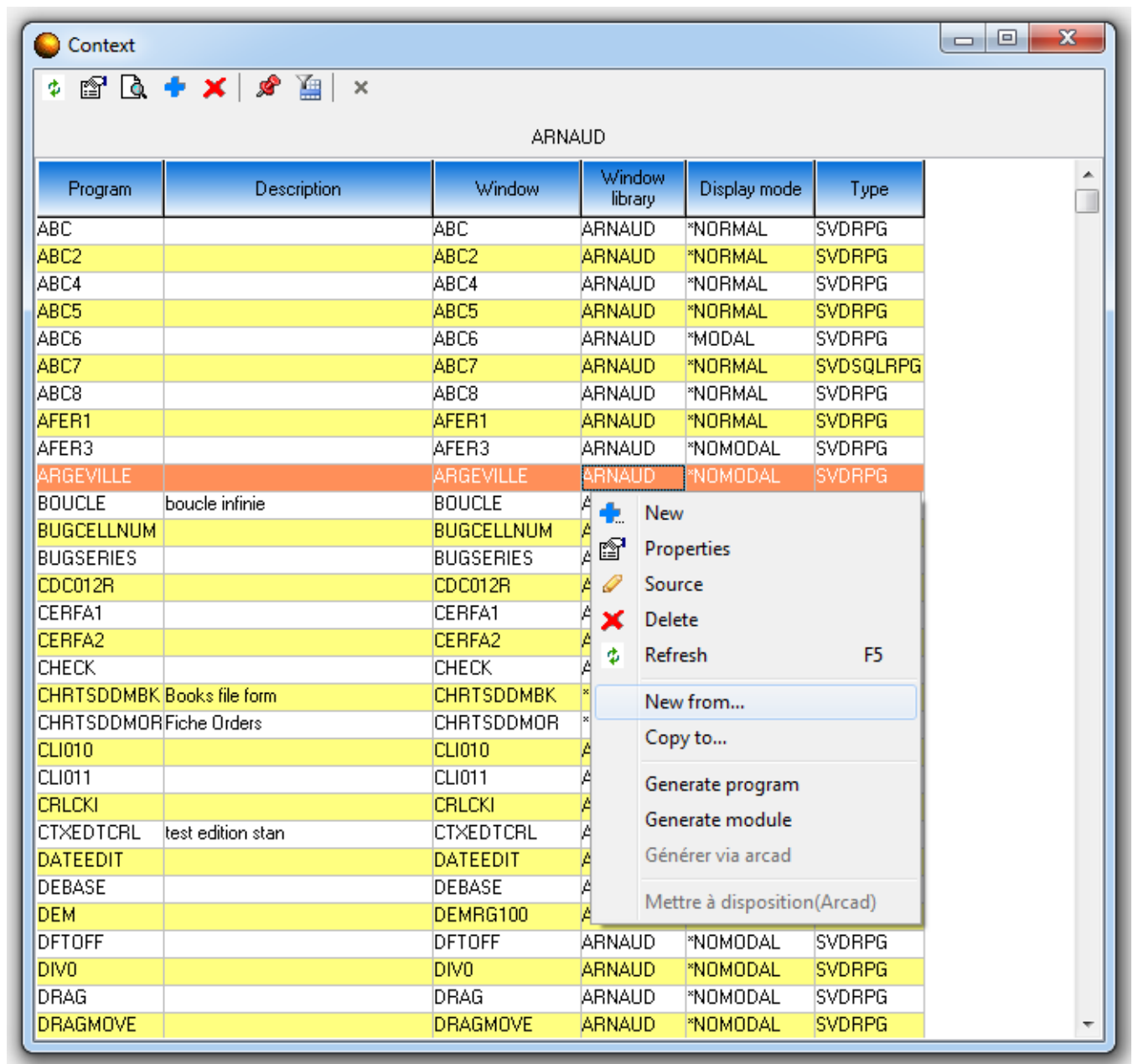
If you have created the components by dragging them from the database assistant, these fields exist once you have declared the files in your program.

For CSFL components, the procedures receive the number of lines to be processed as a parameter.

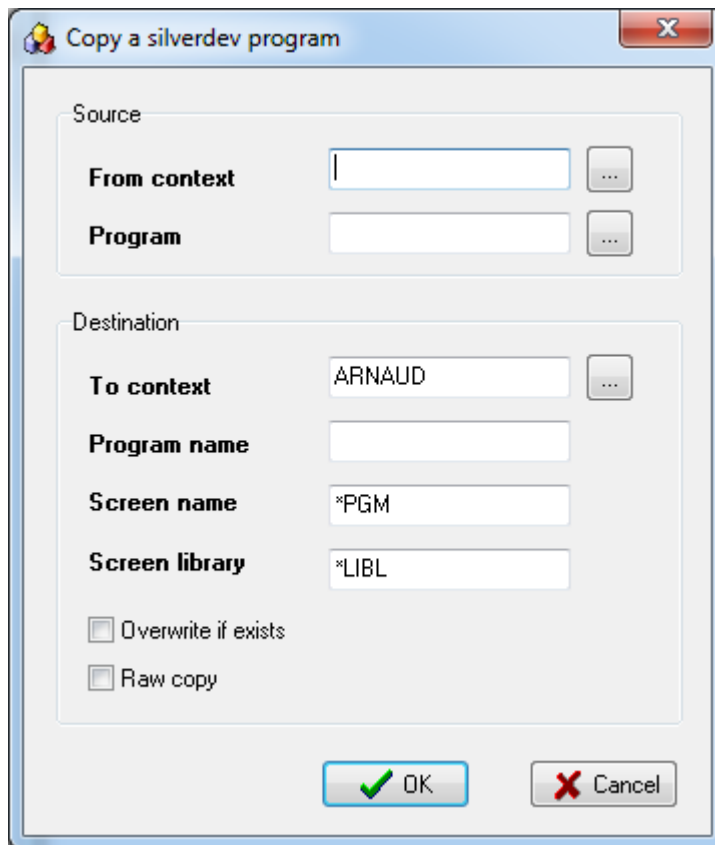
## Chapitre 3. Modèles

In order to create silverdev programs faster, you can create models and copy them.

To copy a model, use the popup menu of the context window, and click on "new from..."



In the displayed window, fill in the fields.



---

## Model parameters.

A model can have some parameters.

To add parameters in a model, add the following comments :

```
// $TPL_VAR  
// $END_TPL_VAR
```

Between these two tags, you can create some model parameters with the tag %subst.

Example :

```
// $SUBST
```

In the designer, you can press F4 to prompt this model parameter.

**\*LIBL/\$SUBST Substitution definition**

Prompt **Text**

DFT  
LEN  
MIN  
PROMPT  
TYPE  
VALUES  
VAR

**VAR**  
Variable  Name

**TYPE**  
Type of value  \*DEC, \*CHAR, \*NAME

**LEN**  
Value length    Number

**DFT**  
Default value  Character value

**VALUES**  
Valid values    Character value

**MIN**  
Minimum values required  0-1

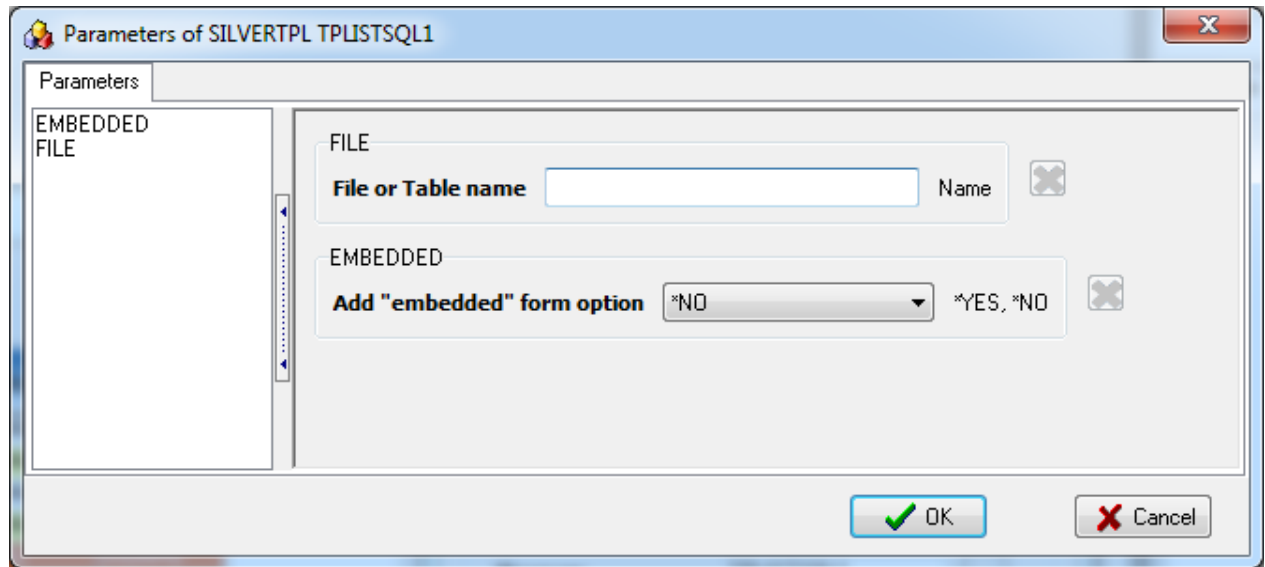
**PROMPT**  
Prompt  Character value

Model parameters declared this way can be used in the source in order to parameter the way source are copied.

Example :

```
// $TPL_VAR
// $SUBST VAR(FILE) TYPE(*NAME) LEN(10) MIN(1) PROMPT('File or Table name')
// $SUBST VAR(EMBEDDED) TYPE(*CHAR) LEN(4) DFT(*NO) VALUES(*YES *NO) PROMPT('Add
"embedded"
//form option')
// $END_TPL_VAR
```

Before the copy is created, a window is displayed with all the model parameters displayed so that you can change their values.



The copied source will be created depending of the values of these parameters.

For example, this section will be copied only if the model parameter "EMBEDDED" is set to \*YES.

```
//<!-- if EMBEDDED = '*YES'
// -- Embedded windows data
d isembedded      s          n
D PEmbedDS        DS          256
D Fparent          like(f1)
d Parentcompo      30a        varying
d ParentMainMnu    30a        varying
//<!-- endif
```

The next text will be replaced depending on the value of the model parameter FILE :

```
exec sql declare C1 cursor for
        select  T1.*, RRN(T1)
        from    !FILE as T1
        for fetch only;
```

Tags `//$TPL` and `//$SUBST` are not copied in the copied source.

## Raw copy

If you want to copy a model to another model, check the checkbox "Raw copy" in the copy dialog box.

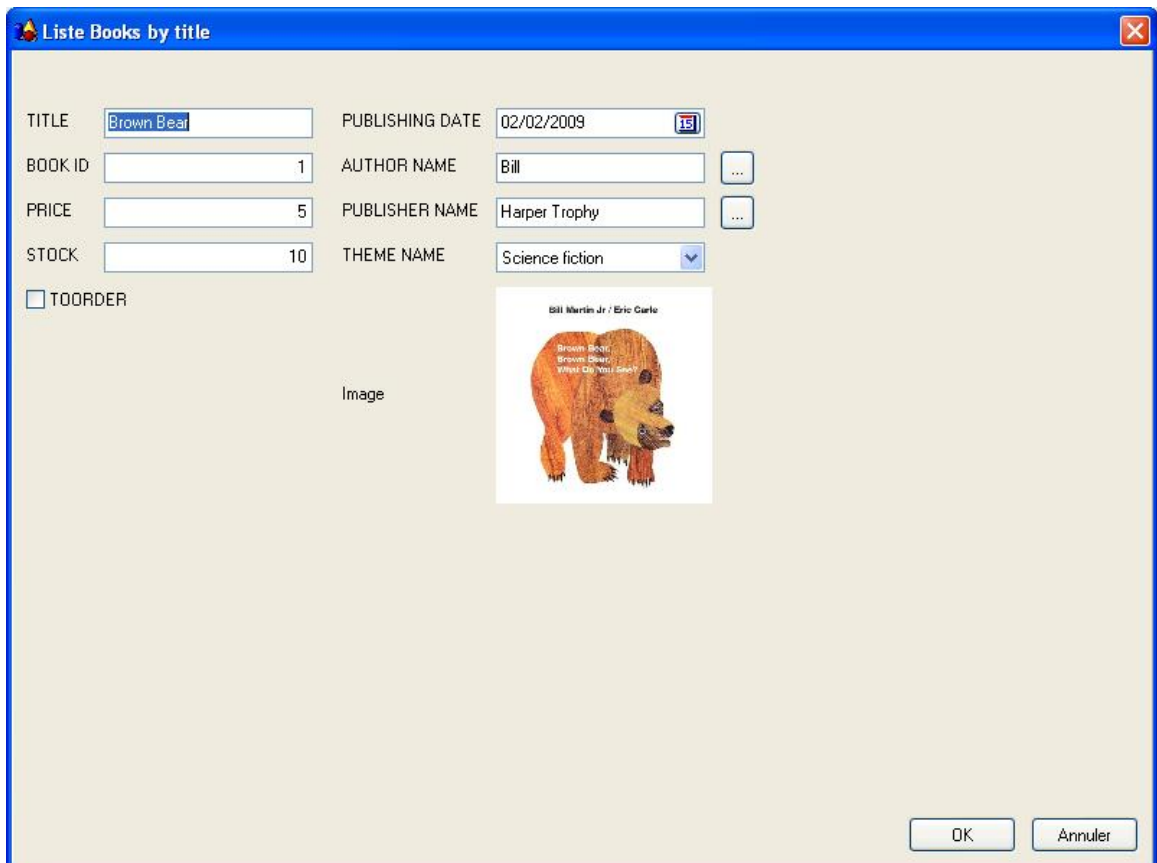
Doing so, parameters will not be prompted and the source will be copied as is.

## Chapitre 4. Wizard

### Introduction.

Grâce à ce tutoriel, vous allez, pas à pas, apprendre à créer votre premier programme SilverDev à l'aide du Wizard. L'assistant Wizard vous permettra de créer la base de votre application en quelques minutes, soit une fiche et une grille, sans avoir à écrire la moindre ligne de code.

#### Création d'un programme de gestion de livres :



**Liste Books by title**

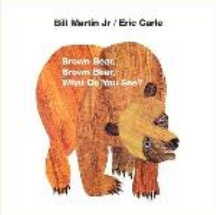
TITLE:  PUBLISHING DATE:

BOOK ID:  AUTHOR NAME:

PRICE:  PUBLISHER NAME:

STOCK:  THEME NAME:

☐ TOORDER

Image: 

OK Annuler



*Vous devez au préalable avoir installé SilverDev sur votre serveur System i ainsi que la partie cliente sur votre poste de travail. Vous devez également avoir démarré SilverDev (commande*



*STRSVD) sur le serveur. Si ces 2 étapes n'ont pas déjà été réalisées, veuillez vous référer au document « Démarrage rapide » avant de poursuivre.*

---

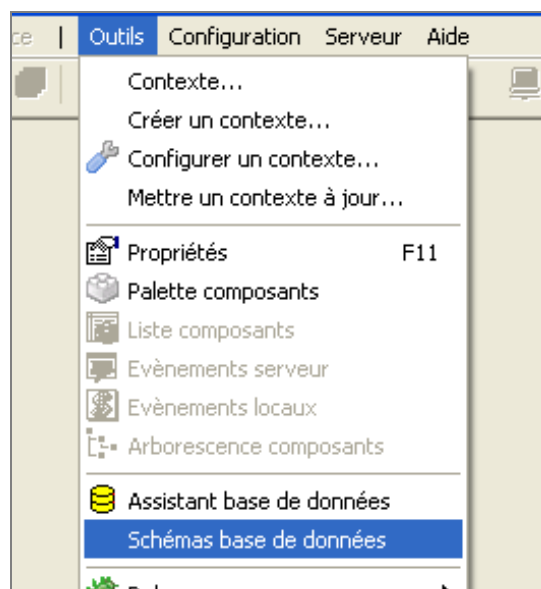
## Outil schéma



Lancez le programme Designer.exe et connectez-vous avec votre profil AS400

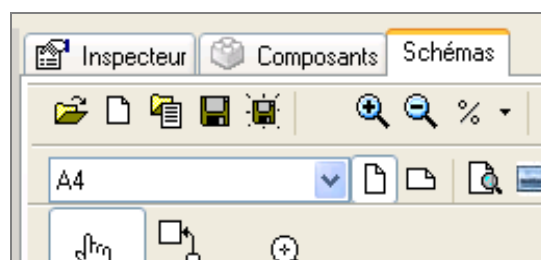
1

Utilisez le menu "Outils/Schémas base de données"



2

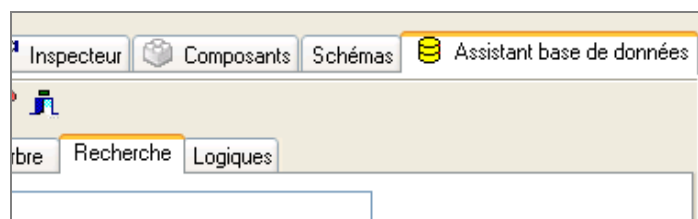
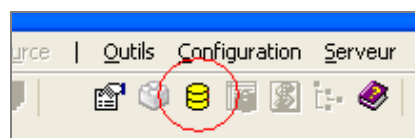
La fenêtre "Schémas" apparaît à l'écran. Cliquez sur le bouton "Nouveau schéma" pour ouvrir un schéma vierge.



## Assistant base de données

1

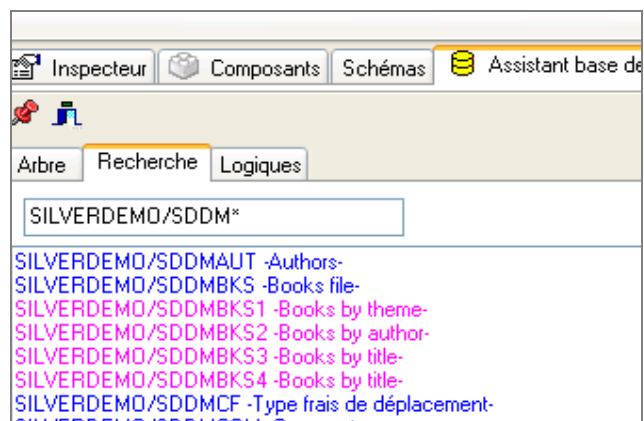
Utilisez l'icône "base de données" pour faire apparaître l'assistant base de données.



2

Dans la zone Recherche, saisissez le texte "**SILVERDEMO/SDDM\***" pour faire apparaître tous les fichiers commençant par SDDM dans la bibliothèque SILVERDEMO.

SILVERDEMO est une bibliothèque de démo installée avec le produit SilverDev.

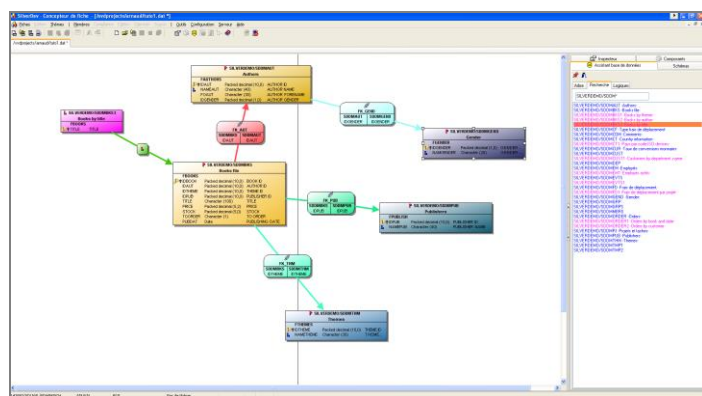


3

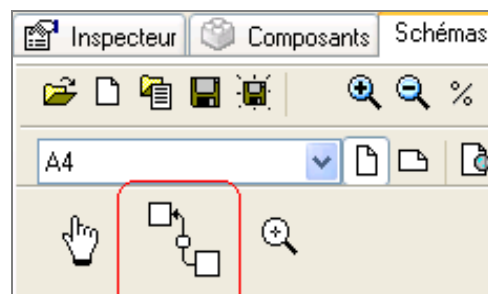
Faites glisser sur le schéma, à l'aide de la souris les fichiers suivants :

- SDDMAUT
- SDDMBKS
- SDDMBKS 3
- SDDMGEND
- SDDMPUB
- SDDMTHM

Des relations sont automatiquement créées entre les tables. Ces informations ont été déduites grâce à l'existence de contraintes (SQL ou natives) entre les tables.

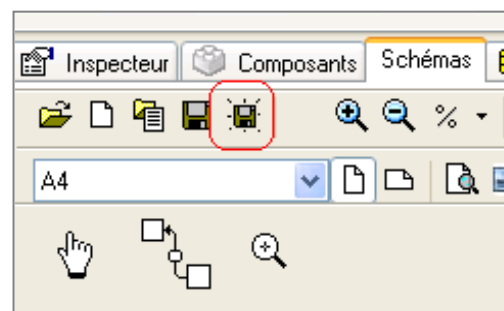


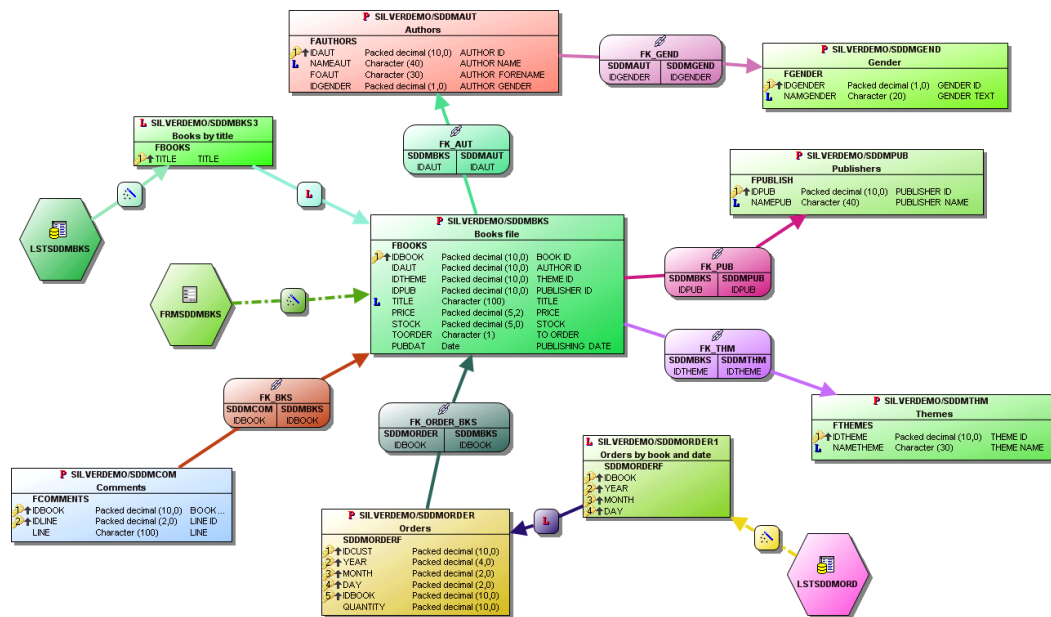
Lorsque vous travaillerez sur vos propres fichiers, et qu'il n'existera pas de contraintes entre les tables, vous pourrez relier deux tables manuellement dans le schéma grâce au bouton ci-contre.



## Sauvegarde du schéma.

Pour sauvegarder le schéma, utilisez le bouton ci-contre dans la fenêtre "schémas"





## Paramétrage des tables

Double-cliquez sur la table SDDMAUT. Une boîte de dialogue apparaît.

Remplissez les champs de la partie "Paramètres de clef étrangère" comme ci-dessous :

Procédez de même pour les tables SDDMPUB, SDDMTHM, SDDMGENDER

Pour les fichiers SDDMTHM et SDDMGENDER, nous choisissons l'option "liste déroulante" car il n'y a pas beaucoup d'enregistrements dans ces fichiers.

Pour les fichiers SDDMAUT et SDDMPUB, il y a trop d'enregistrements pour les afficher dans une liste déroulante. C'est pourquoi nous choisissons "Programme externe" pour ces fichiers.

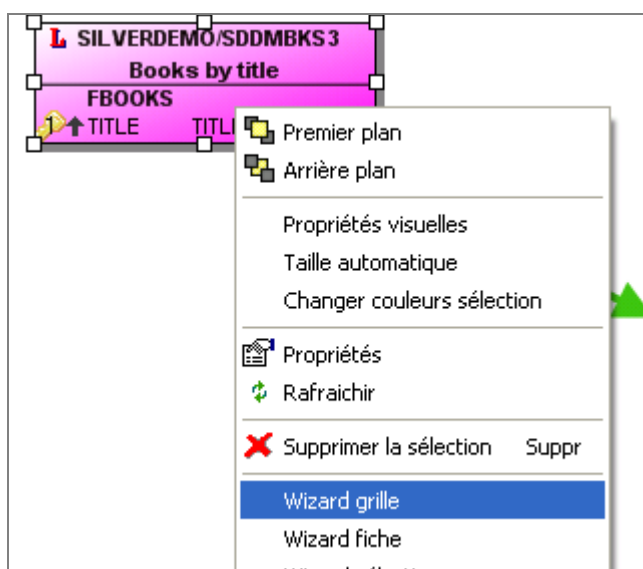
Nous verrons plus tard quelle influence a ce paramétrage.



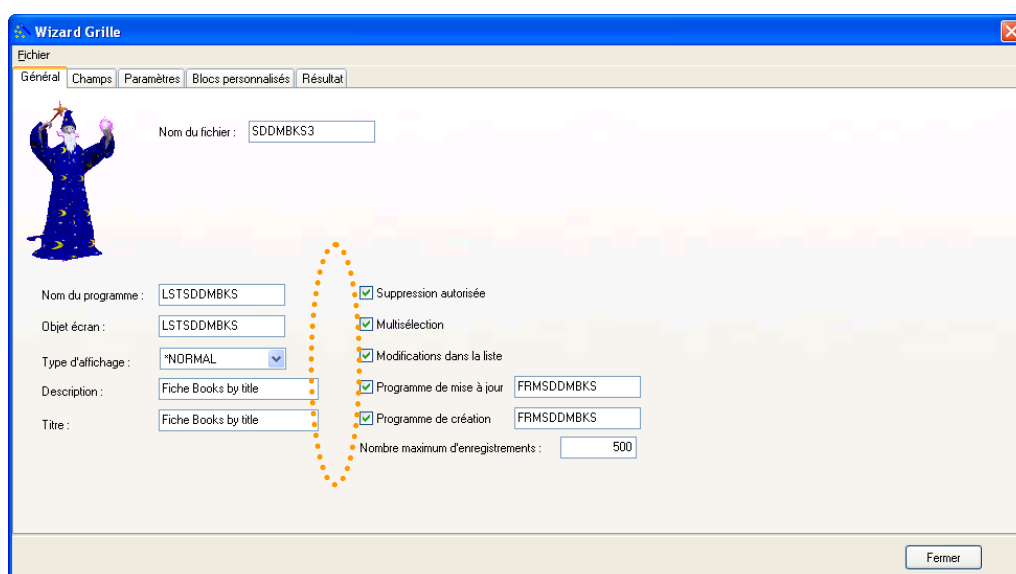
*Il est possible que ces champs soient pré-remplis. Le type de sélection sélectionné est basé sur le nombre d'enregistrements dans la table, et le champ à afficher sélectionné est le premier champ de type alpha.*

## Créer une grille avec le Wizard

Sélectionnez le fichier SDDMBKS3 dans le schéma, effectuez un clic droit, et choisissez le menu "Wizard grille"



Dans la fenêtre qui apparaît, nous choisissons de sélectionner toutes les options afin d'avoir le programme le plus complet possible.



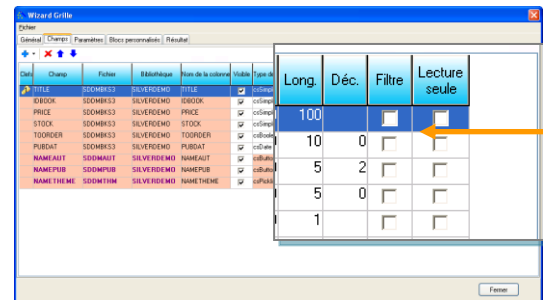
En cliquant sur l'onglet "Champs" on voit apparaître la liste des champs du fichier. Le Wizard a détecté automatiquement le type de colonne adapté à chaque champ. Les champs *NAMEAUT*, *NAMEPUB*, et *NAMETHÈME* proviennent d'autres tables que celles sur laquelle nous travaillons. Elles apparaissent d'une couleur différente.

Le paramétrage créé précédemment a permis de déterminer quel champ sera affiché ainsi que le type de colonne.

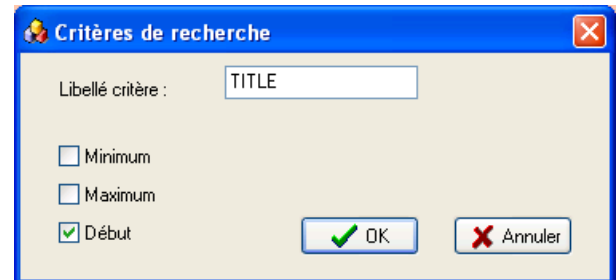


*Il est possible d'ajouter, de modifier ou de supprimer des champs.*

- 1 Cliquez sur la case à cocher dans la colonne **Filtre** sur la ligne du champ **TITLE**.



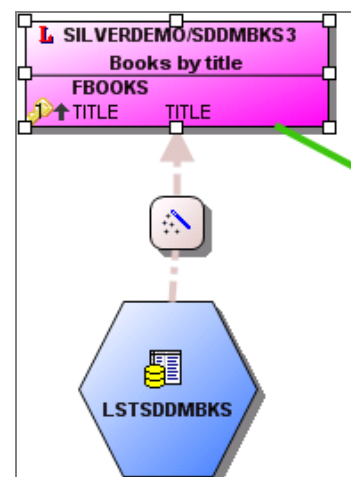
- 2 Cochez la case **début**. Cela permettra d'avoir un critère de recherche sur le début du champ **TITLE**.



- 3 Utilisez le menu "Fichier/Appliquer au schéma". Cela permet de transférer les informations du Wizard vers le schéma.



- 4 Une figure représentant le Wizard doit apparaître dans le schéma comme ci-contre. Il sera alors possible de revenir modifier les propriétés du Wizard en double cliquant sur cette figure.

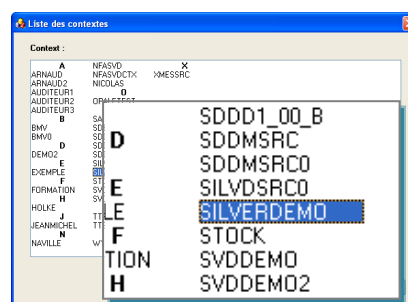


## Génération du source

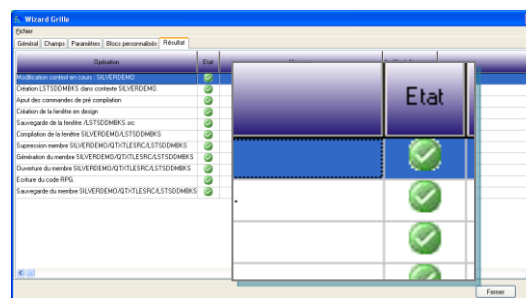
- 1 Nous allons maintenant générer le source du programme en utilisant le menu "Fichier/Générer"



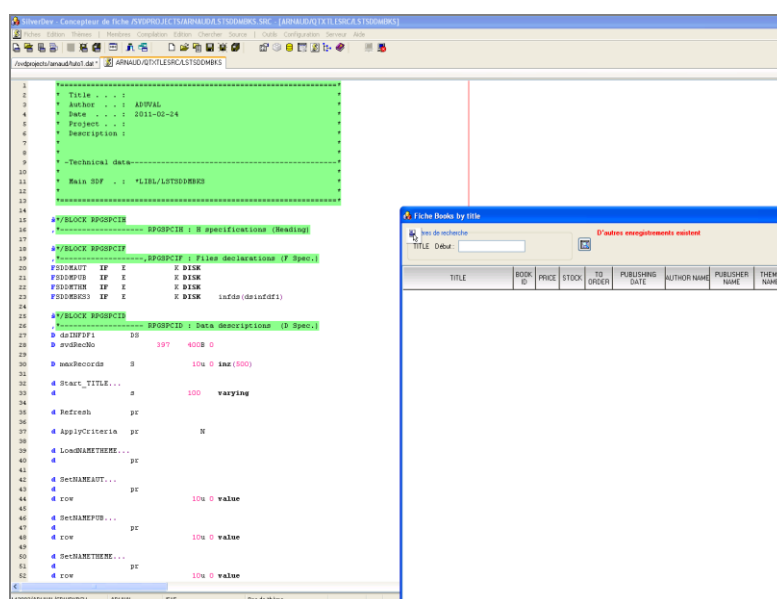
- 2 Une fenêtre nous demande dans quel contexte le programme doit être généré. Choisissez le contexte SILVERDEMO.



- 3 L'onglet résultat s'ouvre. La liste des opérations effectuées par le Wizard s'affiche. Si toutes les opérations se sont déroulées correctement, le symbole « OK » s'affiche derrière chaque opération.



- 4 L'objet écran est alors créé, ainsi que le source du programme qui apparaît en arrière plan. Fermez la fenêtre du Wizard pour accéder au source généré.





Vous pouvez utiliser la touche F10 pour passer du source RPG à la fenêtre créée.

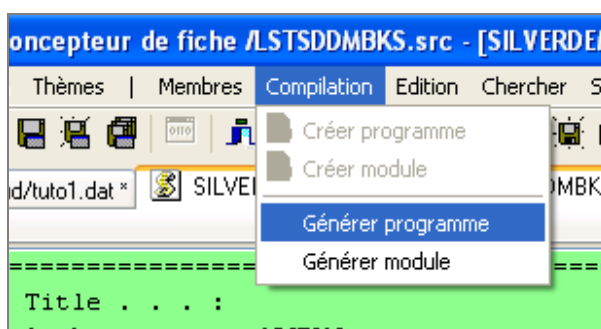
A ce stade, il est possible de modifier à la main l'écran et le source du programme pour lui apporter vos propres spécificités.

Reportez-vous alors au document « Développement Classique » disponible en téléchargement sur <http://www.silverdev.com/test-silverdev.php> ou dans la documentation incluse dans l'archive .zip de SilverDev.

## Compilation du programme

1

Pour compiler le programme, utilisez le menu "Compilation/Générer programme".



2

Le résultat de compilation s'affiche dans une nouvelle fenêtre.

Puisque nous n'avons pas modifié le source RPG, il n'y a aucune raison pour que le programme ne compile pas.

| Résultats de compilation                                       |       |         |        |   |                            |
|--|-------|---------|--------|---|----------------------------|
| Identifiant  | Ligne | Colonne | Niveau | Message   | Fichier source             |
| NF5409   | 246   | 1       | 0      | The prototyped call returns a value which is lost when CALLP is used. | SILVERDEMO/QRPGLESRC(LSTSD |
| NF7031   | 350   | 7       | 0      | The name or indicator EVT is not referenced.                          | SILVERDEMO/QRPGLESRC(LSTSD |
| NF7031   | 348   | 7       | 0      | The name or indicator PARAMETERS is not referenced.                   | SILVERDEMO/QRPGLESRC(LSTSD |
| NF7031   | 411   | 7       | 0      | The name or indicator WIN is not referenced.                          | SILVERDEMO/QRPGLESRC(LSTSD |
| NF7031   | 349   | 7       | 0      | The name or indicator WIN is not referenced.                          | SILVERDEMO/QRPGLESRC(LSTSD |
| NF7031   | 322   | 7       | 0      | The name or indicator WIN is not referenced.                          | SILVERDEMO/QRPGLESRC(LSTSD |
| NF7031   | 323   | 7       | 0      | The name or indicator EVT is not referenced.                          | SILVERDEMO/QRPGLESRC(LSTSD |
| NF7031   | 321   | 7       | 0      | The name or indicator PARAMETERS is not referenced.                   | SILVERDEMO/QRPGLESRC(LSTSD |
| B222/ADUVAL/SDWRKBCH ADUVAL JS15 Pas de thème 24:49 RPGLE Fich |       |         |        |   |                            |

## Mise à disposition du programme



MyDesk.exe

Maintenant que nous avons créé notre premier programme, nous allons utiliser le programme MyDesk pour ajouter un lien vers ce programme et lancer son exécution.

Assurez vous que vous êtes administrateur SilverDev en ajoutant votre profil dans le fichier SILVERDEV/PSVDADM.

Lancez le programme MyDesk.exe, connectez vous à l'aide du menu "Serveur/Connexion"

**Nouveau**

**Titre :**  
tuto1

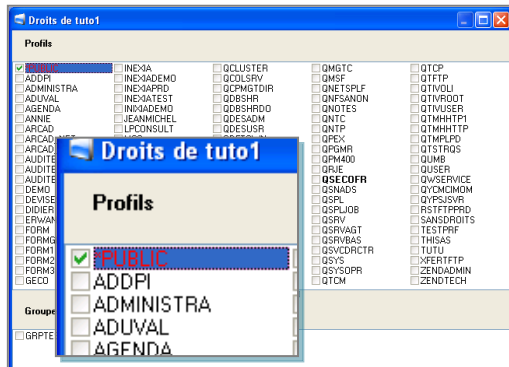
**Nom du fichier app :**  
tuto1

**Commande :**  
call silverdemo/ctuto1

**Description :**

☐ Execution unique

Positionnez le curseur dans le dossier de MyDesk dans lequel vous souhaitez créer l'icône.  
Faites un click droit et utilisez le menu "Nouvelle application"  
Remplissez les champs comme ci-contre :



Une fenêtre apparaît ensuite pour établir les droits d'accès à cette application.

Nous choisissons d'autoriser tout le monde à cette application en cochant « \*PUBLIC ».

Le programme appelé par cette icône est SILVERDEMO/CTUTO1.

Ce programme est un CLP avec le code suivant :

```

SEU==>
***** Beginning of data *****
0001.00 PGM
0002.00          ADDLIB   LIB(SILVERDEMO)
0003.00          MONMSG   MSGID(CPF0000)
0004.00          CALL     LSTSDDMBKS
0005.00 ENDPGM
***** End of data *****

```

Ce programme ne fait qu'ajouter SILVERDEMO en liste de bibliothèque et appeler le programme que nous venons de créer avec le Wizard. Vous pouvez ajouter ce CLP dans le fichier source SILVERDEMO/QCLSRC. Vous le compilerez ensuite dans la bibliothèque SILVERDEMO.

## Exécution du programme

Vous pouvez maintenant double cliquer sur l'icône créée dans myDesk.

Le programme se lance et l'écran suivant s'affiche :

**Fiche Books by title**

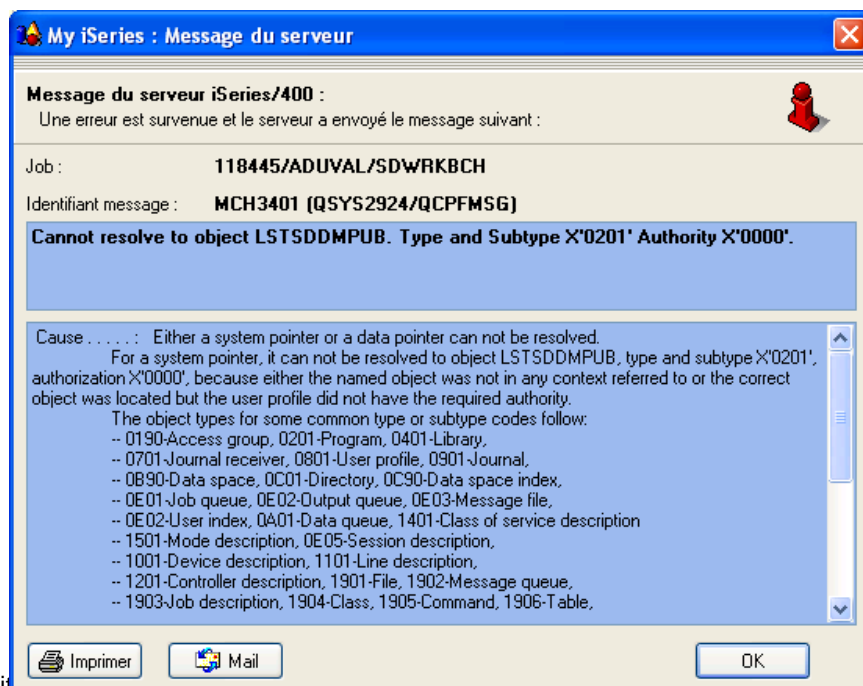
Critères de recherche  
 TITLE Debut :

|  | TITLE                            | BOOK ID | PRICE | STOCK | TO ORDER                 | PUBLISHING DATE | AUTHOR NAME   | PUBLISHER NAME             | THEME NAME       |
|--|----------------------------------|---------|-------|-------|--------------------------|-----------------|---------------|----------------------------|------------------|
|  | Alternative energy for dummies   | 23      | 16    | 3     | <input type="checkbox"/> | 19/05/2009      | DeGunter      | John Wiley & Sons          | Physics          |
|  | Animal breeding                  | 61      | 143   | 15    | <input type="checkbox"/> | 21/04/2010      | Turner        | Earthscan Ltd              | Animals          |
|  | Anne Frank the diary of...       | 56      | 4     | 6     | <input type="checkbox"/> | 01/06/1993      | Frank         | Bantam Books               | History          |
|  | Applied Nutrition for Young Pigs | 62      | 74    | 18    | <input type="checkbox"/> | 06/11/2006      | Mavromichalis | CABI Publishing            | Animals          |
|  | Asterix in britain               | 29      | 7     | 11    | <input type="checkbox"/> | 01/06/2010      | Goscinny      | Orion                      | Comics           |
|  | Black's law dictionary           | 35      | 55    | 0     | <input type="checkbox"/> | 17/08/2009      | Garner        | West Group                 | Law              |
|  | Brown Bear                       | 1       | 5     | 10    | <input type="checkbox"/> | 02/02/2009      | Bill          | Henry Holt and Co.         | Children's books |
|  | Bullet                           | 27      | 19    | 7     | <input type="checkbox"/> | 01/06/2010      | Hamilton      | Berkley Publishing group   | Romance          |
|  | Contemporary strategy analysis   | 55      | 39    | 8     | <input type="checkbox"/> | 27/11/2009      | Grant         | John Wiley & Sons          | Business         |
|  | Dental anthropology              | 46      | 38    | 2     | <input type="checkbox"/> | 28/09/1996      | Hilson        | Cambridge University Press | Medicine         |
|  | Disability on equal terms        | 37      | 25    | 10    | <input type="checkbox"/> | 17/08/2009      | French        | SAGE Publications          | Law              |
|  | Duma key                         | 8       | 6     | 7     | <input type="checkbox"/> | 01/10/2008      | King          | Pocket Books               | Science fiction  |
|  | Dune                             | 58      | 5     | 12    | <input type="checkbox"/> | 01/02/1996      | Herbert       | Ace books                  | Science fiction  |
|  | Fluid mechanics measurements     | 24      | 168   | 1     | <input type="checkbox"/> | 01/03/1996      | Goldstein     | Taylor & Francis           | Physics          |
|  | Football for dummies             | 41      | 14    | 8     | <input type="checkbox"/> | 22/06/2007      | Long          | John Wiley & Sons          | Sports           |
|  | French cooking                   | 12      | 14    | 11    | <input type="checkbox"/> | 21/10/2009      | Child         | Penguin Books              | Cooking          |
|  | From Head to Toe                 | 2       | 4.39  | 11    | <input type="checkbox"/> | 14/03/2007      | Carle         | Harper Trophy              | Children's books |
|  | Fundamentals of physics          | 18      | 52    | 2     | <input type="checkbox"/> | 10/08/2007      | Walker        | John Wiley & Sons          | Physics          |
|  | I know how to cook               | 15      | 37    | 12    | <input type="checkbox"/> | 24/09/2009      | Mathiot       | Phadon Press               | Cooking          |
|  | Indian food made easy            | 16      | 16    | 10    | <input type="checkbox"/> | 24/09/2009      | Anand         | Quadrille Publishing       | Cooking          |
|  | Italian wines 2009               | 14      | 25    | 14    | <input type="checkbox"/> | 15/09/1999      | Rosso         | Gambero Rosso Inc          | Cooking          |
|  | Japanese Kinono designs          | 59      | 3     | 7     | <input type="checkbox"/> | 01/01/2009      | Sun           | Dover Publications Inc     | Home             |
|  | JavaScript                       | 49      | 36    | 7     | <input type="checkbox"/> | 25/08/2006      | Flanagan      | O'Reilly Media             | Programming      |
|  | Just after sunset                | 10      | 6     | 7     | <input type="checkbox"/> | 04/08/2009      | King          | Simon & Schuster           | Science fiction  |
|  | Learning php                     | 50      | 29    | 4     | <input type="checkbox"/> | 24/07/2009      | Nixon         | O'Reilly Media             | Programming      |
|  | Man vs Wild                      | 52      | 19    | 3     | <input type="checkbox"/> | 01/05/2008      | Gylls         | Hypenion Books             | Nature           |
|  | Medical statistics at a glance   | 45      | 24    | 8     | <input type="checkbox"/> | 10/07/2009      | Petrie        | Wiley-Blackwell            | Medicine         |
|  | Mythic river                     | 7       | 3     | 12    | <input type="checkbox"/> | 01/04/2002      | Lehane        | Harper Torch               | Science fiction  |
|  | Naruto                           | 30      | 7     | 4     | <input type="checkbox"/> | 01/06/2010      | Hishimoto     | Viz Media                  | Comics           |
|  | Open                             | 39      | 22    | 9     | <input type="checkbox"/> | 10/05/2009      | Agassi        | HarperCollins Publishers   | Sports           |
|  | Phide and prejudice              | 25      | 2     | 4     | <input type="checkbox"/> | 01/08/1995      | Austen        | Dover thrit                | Children's books |

Valider

Il est déjà possible d'effectuer un filtre avec la zone en entête, de trier sur une ou plusieurs colonnes en cliquant sur les entêtes, de supprimer des enregistrements, ou de modifier directement dans la grille.

En revanche, si vous effectuez un double click sur une ligne ou un click sur le bouton dans la colonne Publisher, un message d'erreur s'affiche :

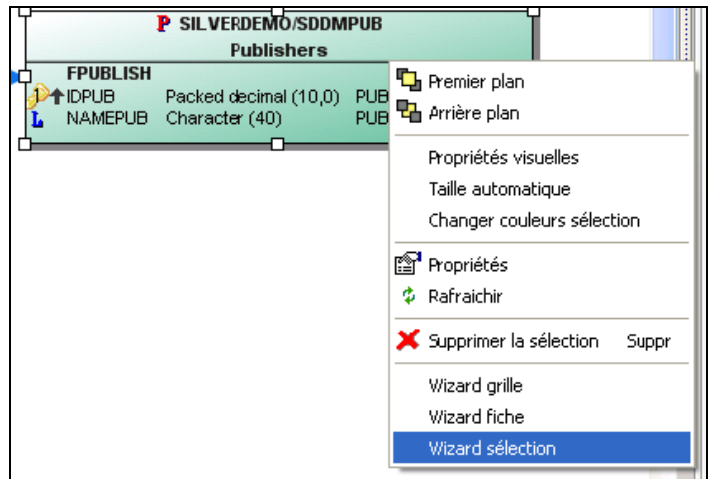


En effet, dans le Wizard, nous avons demandé d'appeler les programmes FRMSDDMBKS, LSTSDMPUB et LSTSDDMAUT, mais nous n'avons pas créé ces programmes.

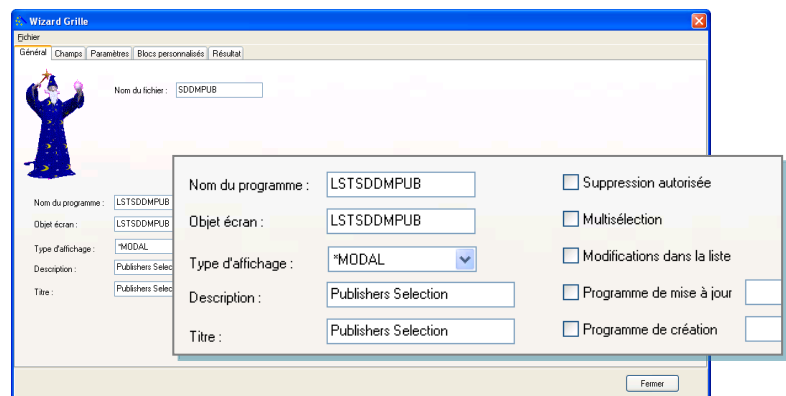
Pour créer ces programmes nous allons à nouveau utiliser le Wizard.

## Wizard Sélection

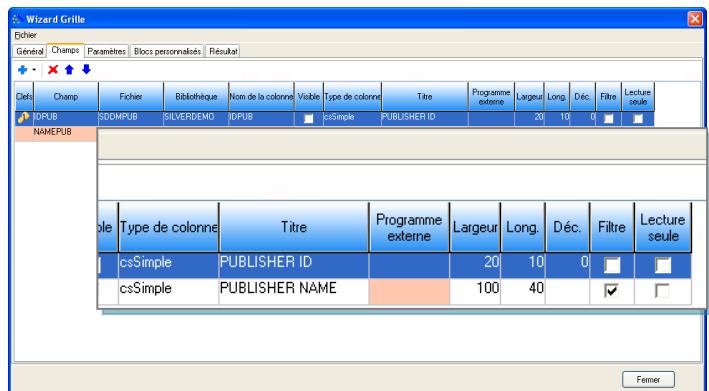
1 Sélectionnez la table SDDMPUB dans le schéma , effectuez un clic droit, et utilisez le menu "Wizard sélection"



2 Nous n'allons rien modifier dans l'onglet Général.



3 Dans l'onglet "Champs", nous ajoutons un filtre sur le champ *NAMEPUB* comme nous l'avions fait dans le programme principal.



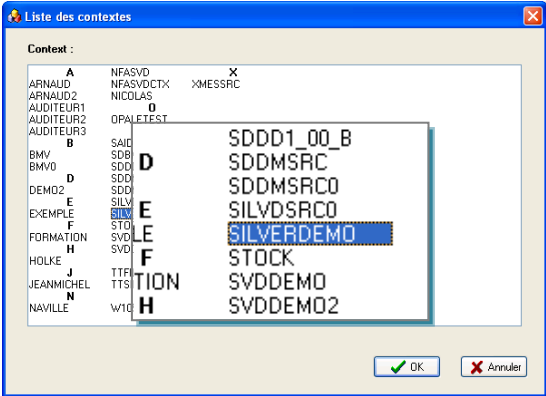
4 Appliquez le Wizard au schéma à l'aide du menu "Fichier/Appliquer au schéma"



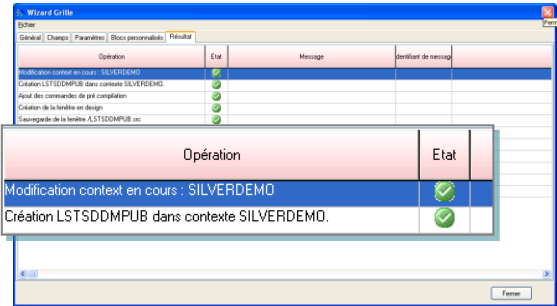
5 Générez le programme à l'aide du menu "Fichier/Générer"



6 Choisissez à nouveau le contexte SILVERDEMO.



7 Le résultat de l'opération de génération est affiché dans l'onglet "Résultats"

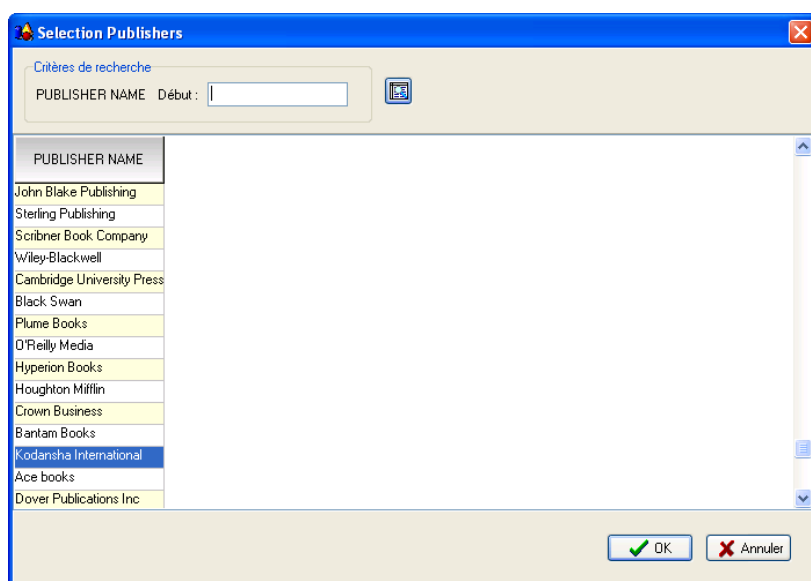


8 Relancez le programme LSTSDDMBKS. Maintenant, lorsqu'on clique sur le bouton dans la colonne Publisher...

| AME | PUBLISHER NAME          | THEME     |
|-----|-------------------------|-----------|
|     | Murdoch Books           | Cooking   |
|     | Scribner Book Company   | Sports    |
|     | Puffin Books            | Children' |
|     | Workman Publishing      | Parentin  |
|     | Workman Publishing      | Parentin  |
|     | Hodder & Stoughton Ltd  | Science   |
|     | Kodansha International  | Travel    |
|     | Putnam Publishing Group | Children' |
|     | Low pressure publishing | Sports    |

9

...le programme LSTSDMPUB est appelé et l'écran suivant est affiché :

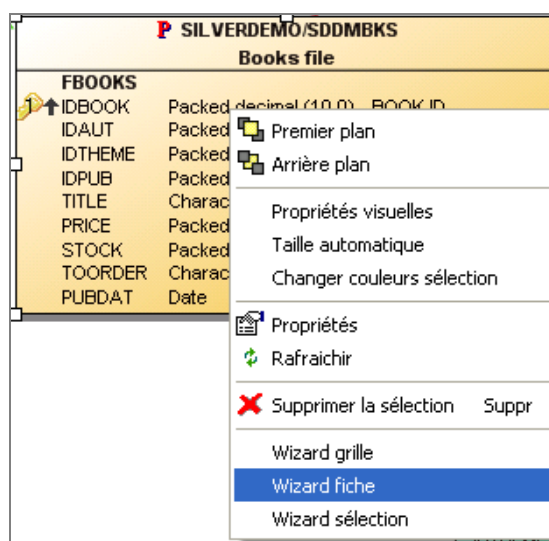


## Créer une fiche avec le Wizard

1

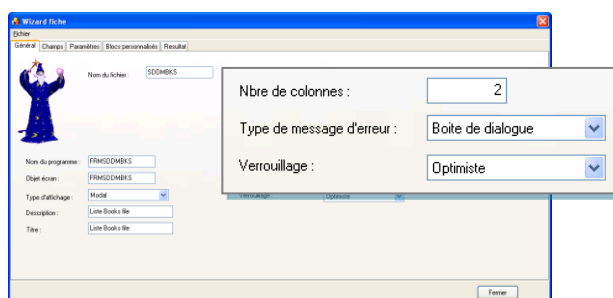
Nous allons maintenant créer le programme FRMSDDMBKS qui est appelé depuis un double clic dans le programme principal.

Sélectionnez la table SDDMBKS dans le schéma, effectuez un clic droit et choisissez l'option "Wizard fiche".

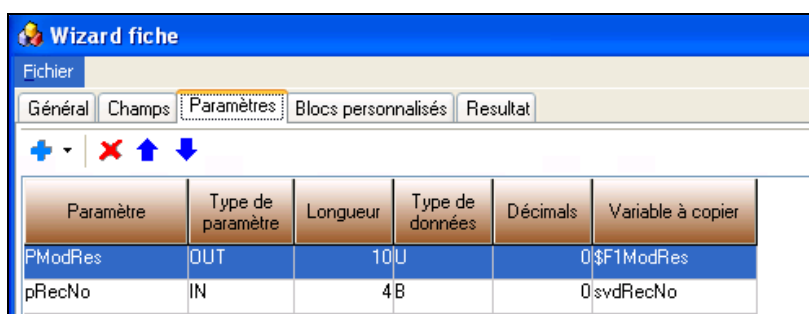


2

Dans l'onglet général, nous pouvons changer le type de message d'erreur en « boîte de dialogue » pour avoir quelque chose de plus élaboré.



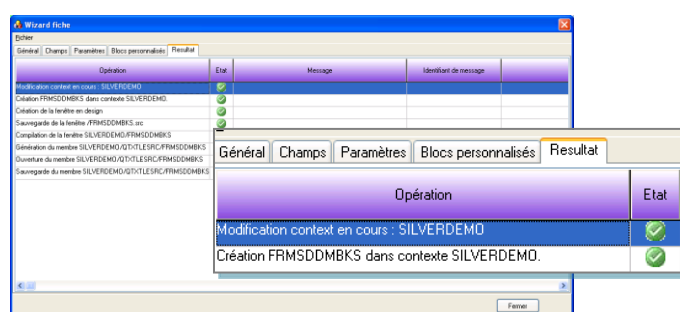
3 A noter dans l'onglet paramètres que le Wizard a cette fois-ci préparé deux paramètres pour le programme.



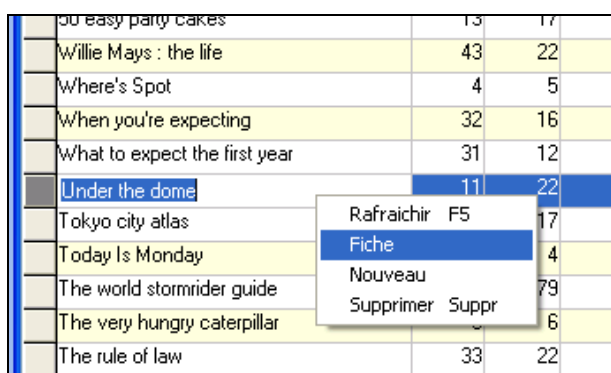
Utilisez les menus "Fichier/Appliquer au schéma" et "Fichier/Générer"

Le résultat est affiché :

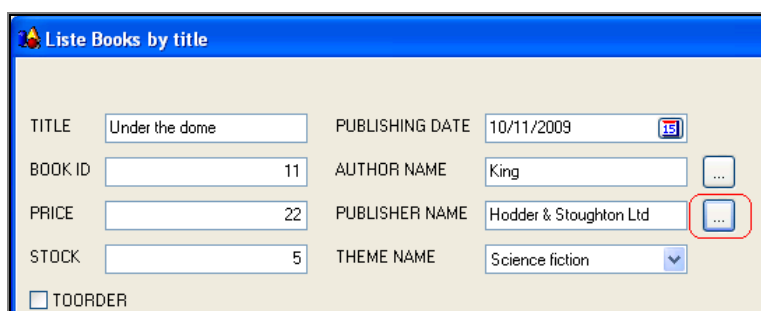
Puis utilisez le menu pour  
"Compiler/générer  
programme"



Redémarrez le programme *LSTSDDMBKS*. Il est maintenant possible d'appeler la fiche par un double click sur la grille ou par un click droit menu "Fiche"

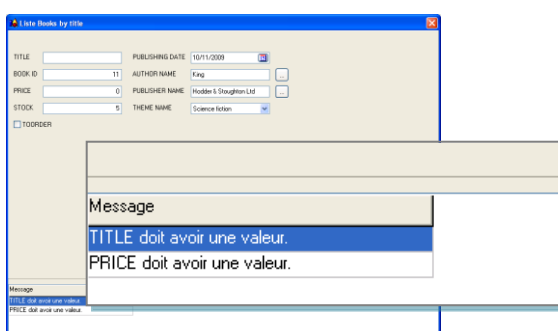


Le bouton devant la zone « PUBLISHER NAME » appelle le programme *LSTSDMPUB* que nous avons déjà appelé depuis le programme principal.



Dans le Wizard, nous avons précisé que le titre et le prix étaient obligatoires.

Si nous essayons de valider sans renseigner ces deux champs, une grille avec la liste des erreurs apparaît. Double cliquez sur une ligne pour placer le « focus » sur le champ incriminé.

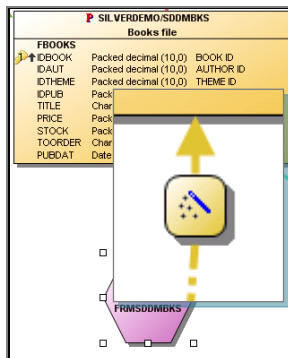




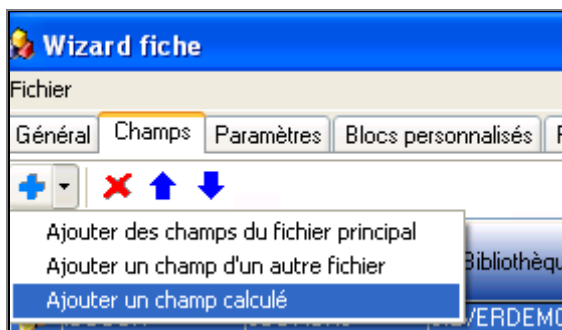


## Pour aller plus loin : afficher une image

- 1 Nous allons revenir sur le Wizard *FRMSDDMBKS*. Double cliquez sur le Wizard dans le schéma :



- 2 Dans la fenêtre du Wizard, sélectionnez l'onglet "Champs" et sélectionnez le menu "Ajouter un champ calculé"

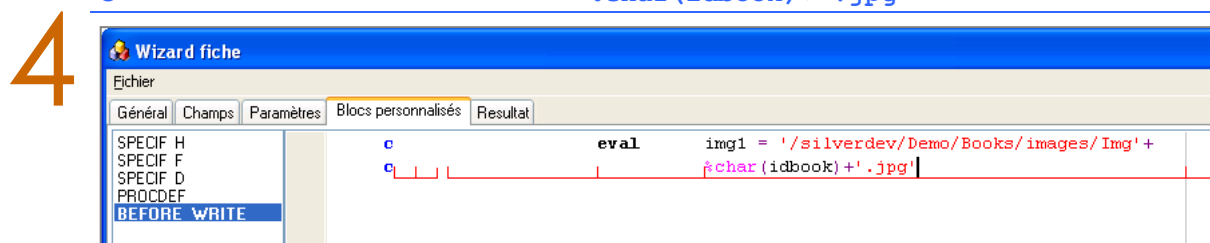


- 3 Renseignez les zones comme ci-contre :

|                     |                                      |                  |  |
|---------------------|--------------------------------------|------------------|--|
| Nom du champ :      | <input type="text" value="Img1"/>    | Type de donnée : | <input type="text" value="Alpha varying"/> |
| Description :       | <input type="text" value="Image :"/> | Longueur :       | <input type="text" value="250"/>           |
| Type de composant : | <input type="text" value="CImage"/>  | Décimales :      | <input type="text" value="0"/>             |
| Type d'image :      | <input type="text" value="jpeg"/>    |                  |  |

Dans l'onglet "Blocs personnalisés", saisissez le code suivant :

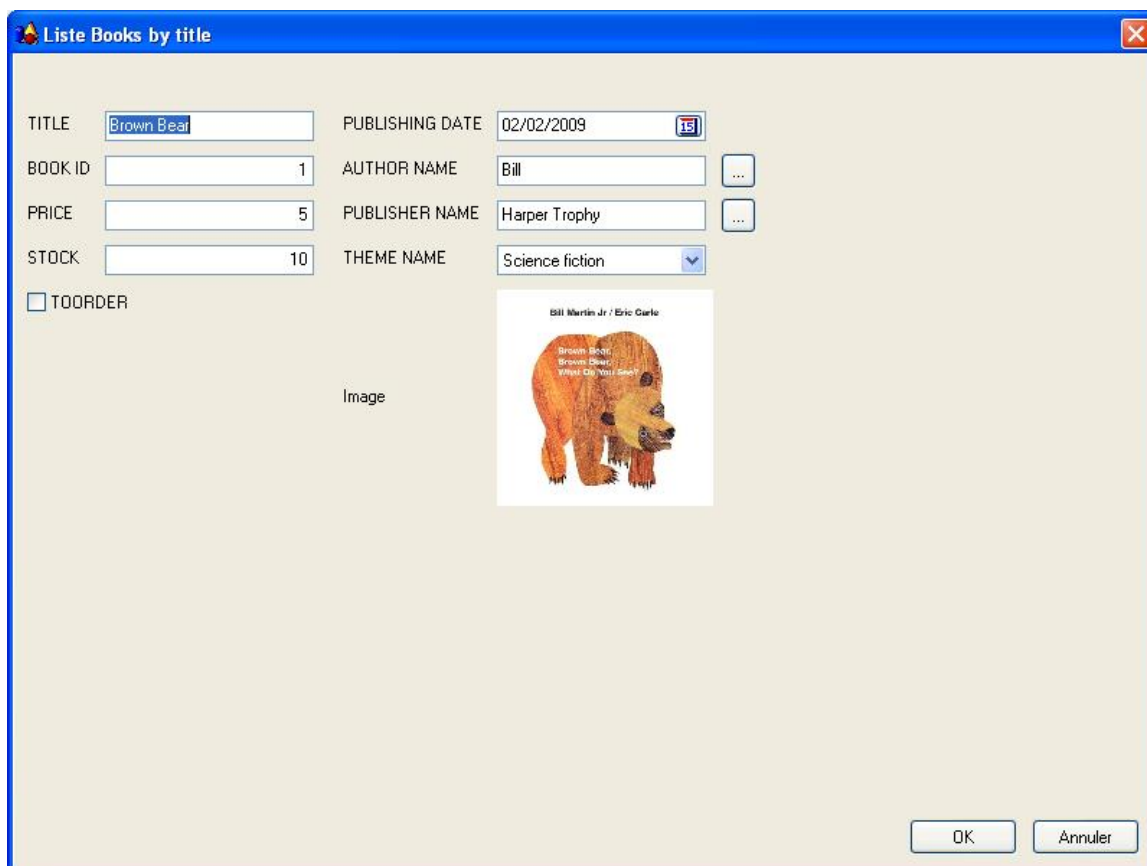
```
C          eval      img1 = '/silverdev/Demo/Books/images/Img'+
C          %char(idbook) + '.jpg'
```



- 5 Utilisez les menus "Fichier/appliquer au schéma" , puis "Fichier/Générer" puis "Compilation/Générer le programme" comme nous l'avons déjà vu pour les programmes précédents.

6

Relancez l'application principale, la fiche contient à présent une image :



The screenshot shows a Windows-style application window titled "Liste Books by title". The window contains a form with the following fields and values:

| Field           | Value           |
|-----------------|-----------------|
| TITLE           | Brown Bear      |
| PUBLISHING DATE | 02/02/2009      |
| BOOK ID         | 1               |
| AUTHOR NAME     | Bill            |
| PRICE           | 5               |
| PUBLISHER NAME  | Harper Trophy   |
| STOCK           | 10              |
| THEME NAME      | Science fiction |

Below the form, there is a checkbox labeled "TOORDER" which is currently unchecked. To the right of the checkbox, there is a placeholder text "Image" and a small image of a book cover. The book cover features a brown bear and the text "Bill Martin Jr / Eric Carle", "Brown Bear", and "What Do You See?". At the bottom right of the window, there are two buttons: "OK" and "Annuler".

---

## Modifier le programme généré

Une fois que ce programme est généré par le Wizard, vous pouvez modifier à la main l'écran ou le source RPG.

Pour cela, utilisez le menu "Outils/Context...", sélectionnez un contexte. La liste des programmes du contexte sélectionné est alors affichée.

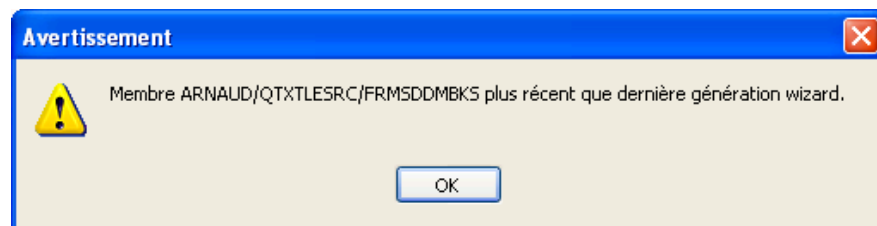
Double cliquez sur un élément dans la liste pour ouvrir le source RPG et la fenêtre en design.

Vous pouvez alors modifier le source généré comme n'importe quel programme SilverDev.

---

## Protection contre l'écrasement.

Une fois qu'un programme est modifié à la main, il n'est plus possible de générer par dessus avec le Wizard. Vous obtiendrez le message suivant :

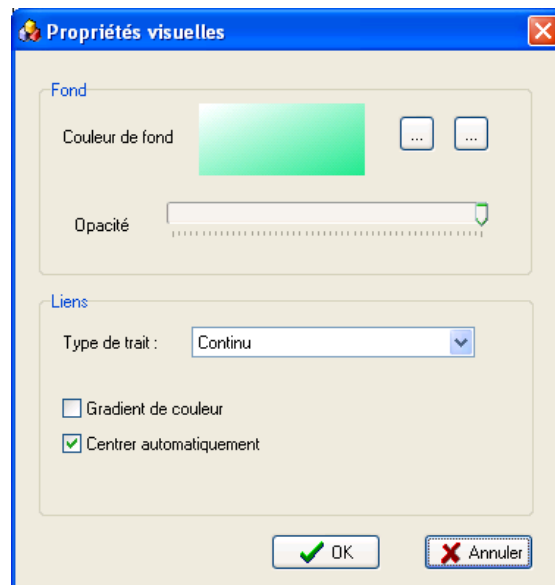
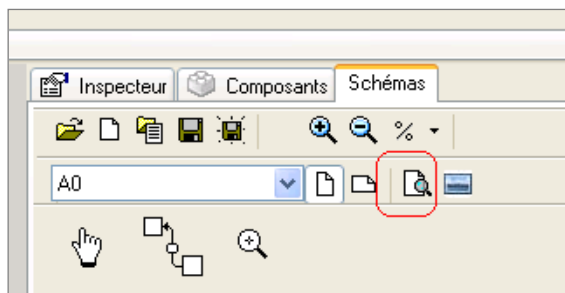


Le Wizard peut toujours être utilisé pour générer dans un autre contexte, sous un autre nom ou en supprimant les objets que vous avez modifié à la main.

---

## Outils

- i) Impression du schéma
- ii) Modification de l'aspect des éléments



Accédez au menu « Propriétés visuelles » par un clic droit.

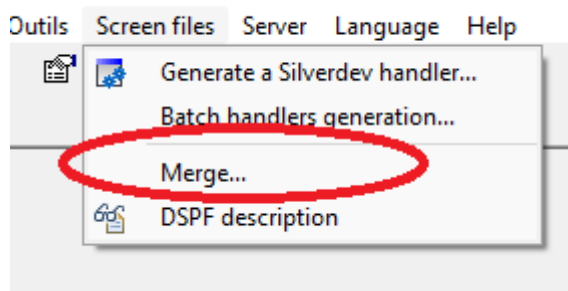
---

## Chapitre 5. Screen conversion

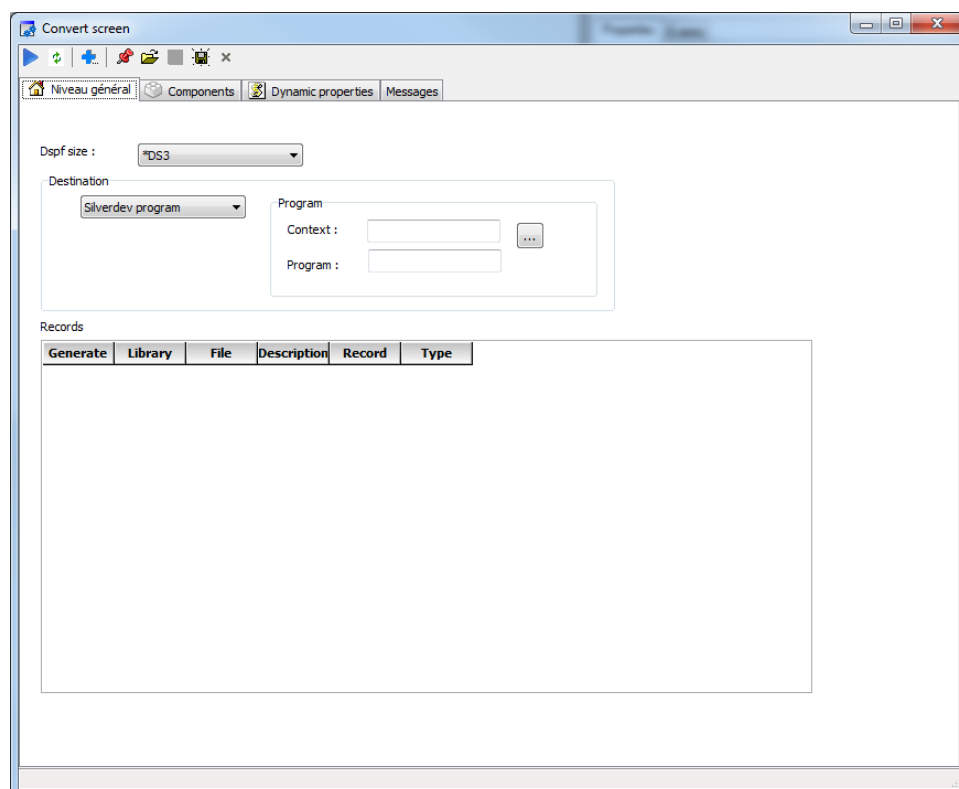
---

### Launch the conversion tool

Use the menu 'Screen files/Merge..'



The workspace window opens.

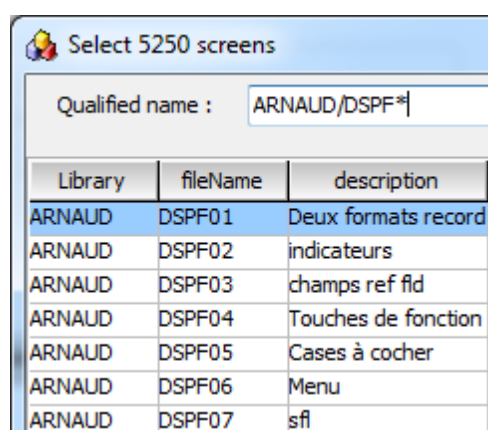
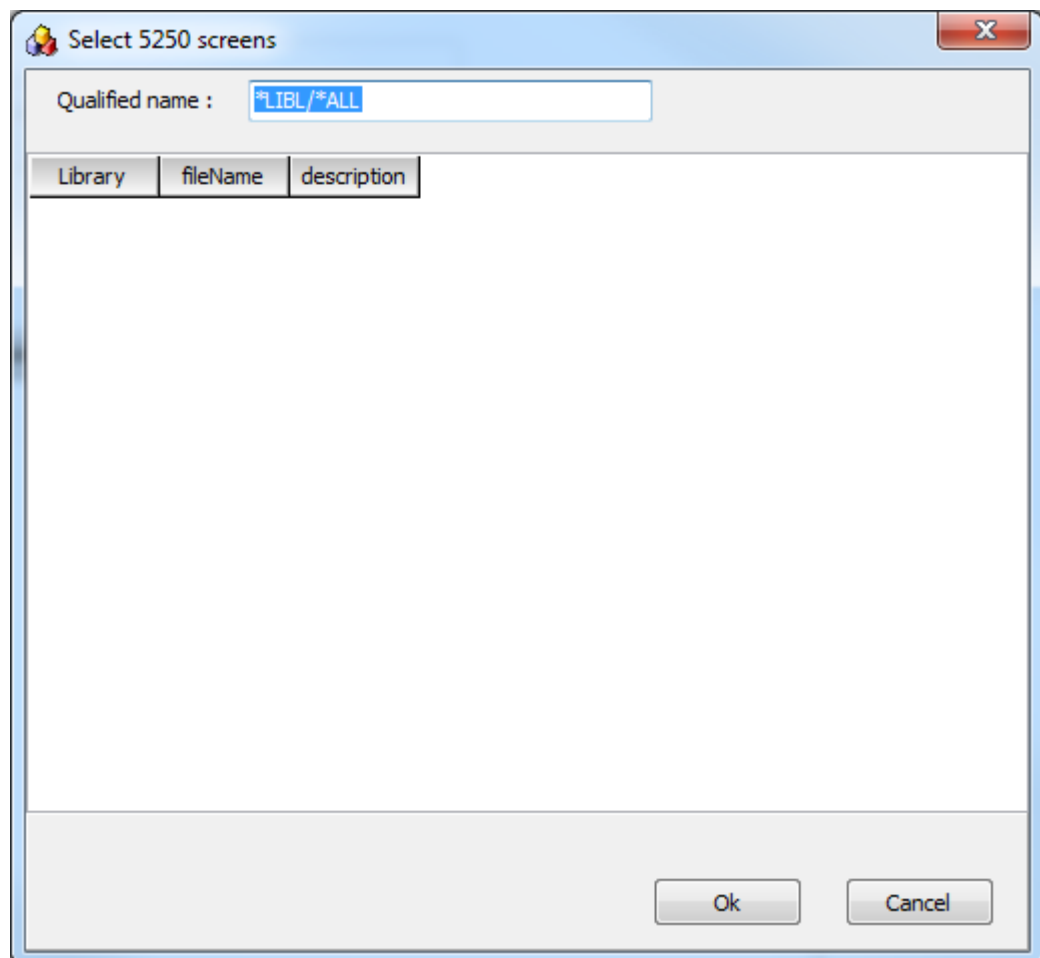


---

### Add 5250 screens

To add 5250 screens, use button :  .

The following window opens :



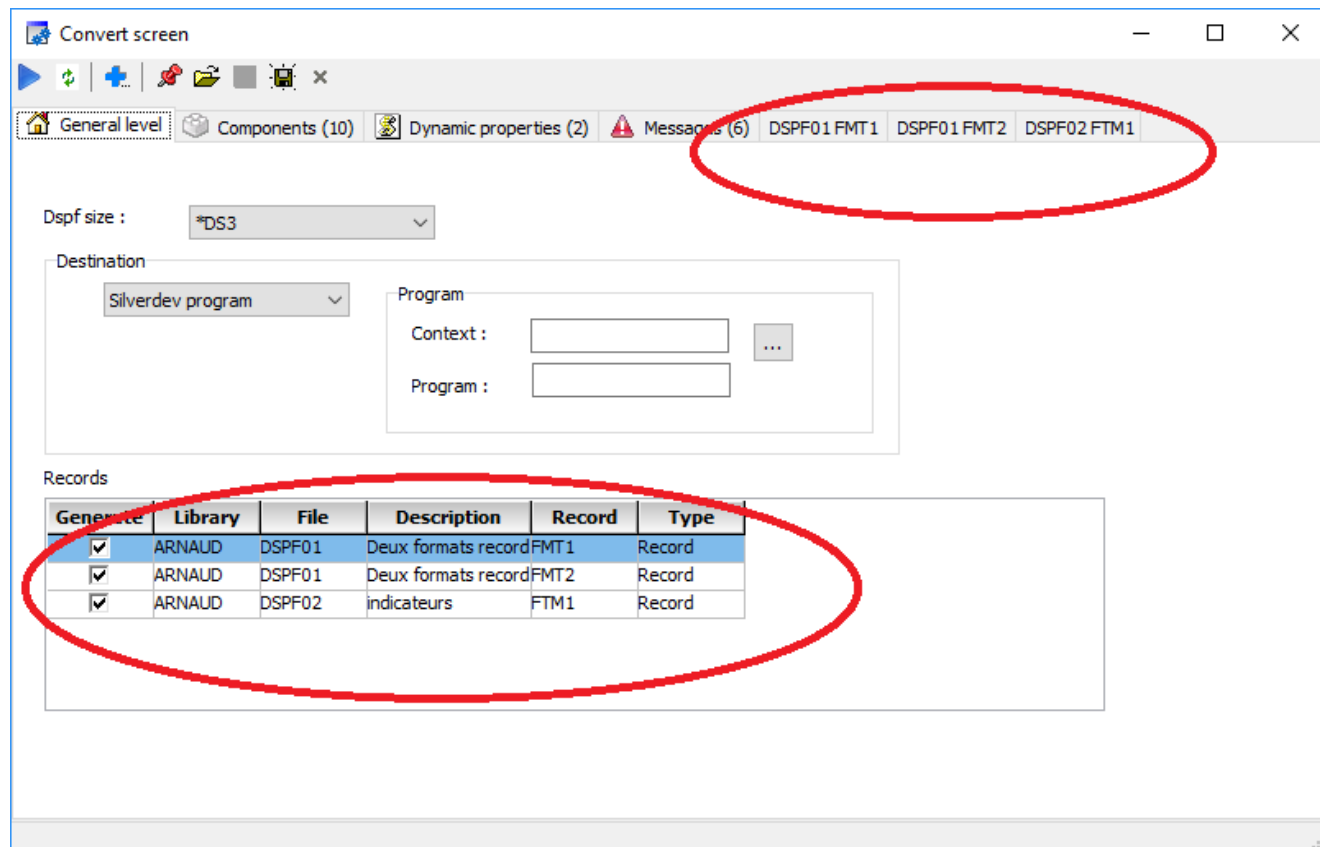
You can select screens to display by filling in the qualified names of these screens. (Press enter to validate)


You can select one or many screens. One of the advantages of SilverDev is that you can put several 5250 screens on one unique silverdev screen.

## Record list

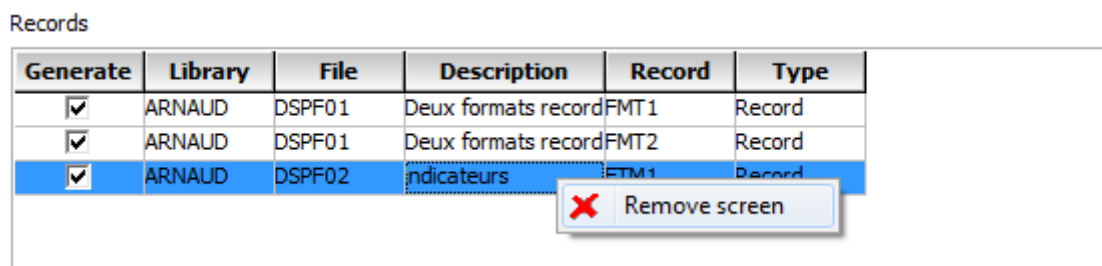
First tabsheet displays your list of records in the selected screens. For each record, a tabsheet is created.

You can disable a record by unchecking it. The corresponding tabsheet is then deleted.



Reminder: You can add 5250 screens with button 

Note : You can also remove a 5250 screen with right click :



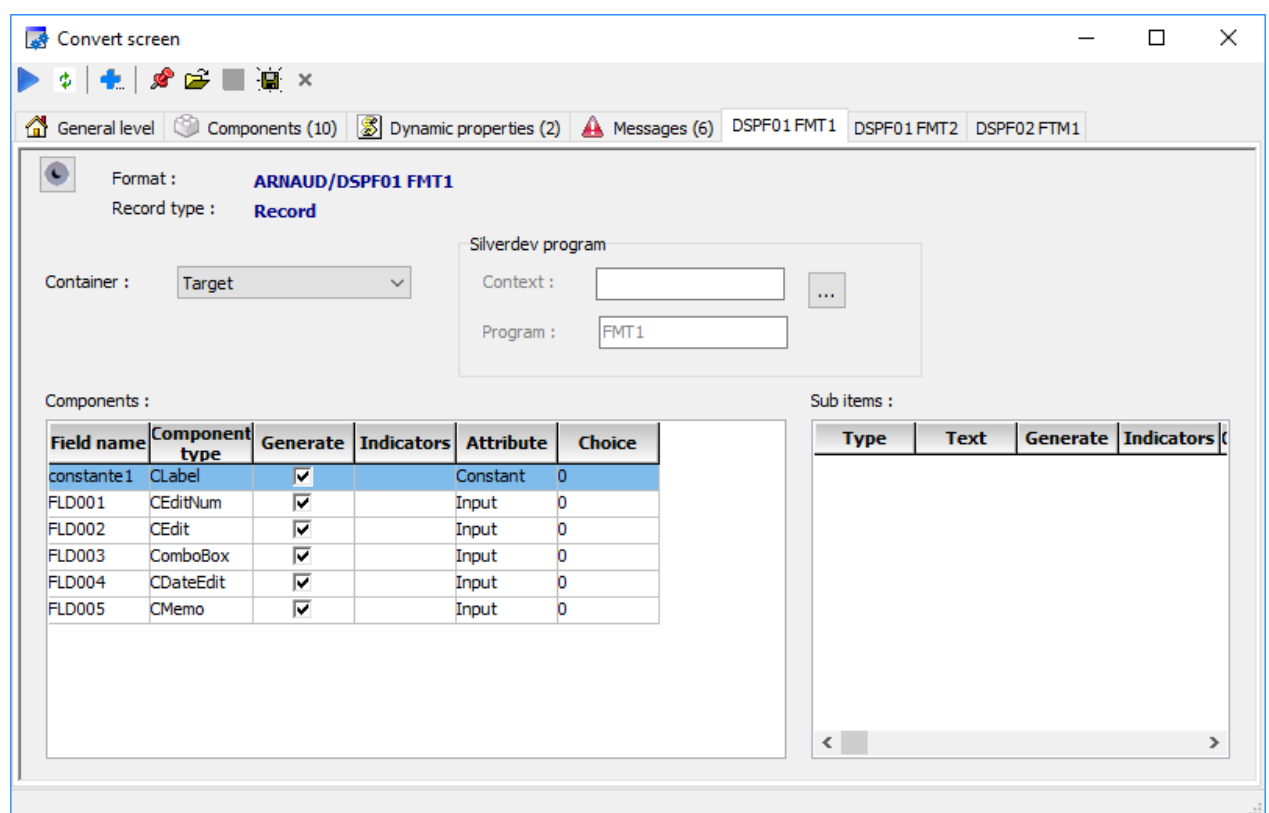


## Record Tabsheets

### Components Grid

For each record, a tabsheet is created. Its appearance depends on record type (record, sfl, menu...)

For a record of record type, the tabsheet looks like the following picture :



'Generate' column allows to determine if a component will be generated for the field. When the box is unchecked; the line is grayed, so is the corresponding line in the 'components' tabsheet.

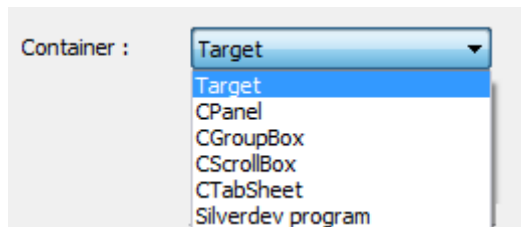
'Component type' column allows to select the type of component that will be used for this field.

The pre-selected value is determined depending of the attributes of the field.

Grid on the right is only useful for 5250 screen fields with keywords such as CHOICE, MNUBARHC, PSHBTNHC.

## Container

The 'container' list box allows to determine how components of this record will be inserted.



|                                      |  |
|--------------------------------------|--|
| <b>Target</b>                        | Directly on the target in design.  |
| <b>CPanel, CGroupBox, CScrollBar</b> | A component of CPanel, CGroupBox or CScrollBar type will be inserted on the target in design and the created components will be inserted in this component.        |
| <b>CTabSheet</b>                     | If the selected target is a CPageControl, a CTabSheet will be added, otherwise, a CPageControl will be created and a CTabSheet will be added to this CPageControl. |
| <b>SilverDev program</b>             | In this case, 'context' and 'program' fields must be filled in. Another program will be created. This Value is pre selected by default for records of type window. |

Note: When one or many records are set with the container type 'SilverDev program', the tree view in the 'components' tabsheet has several root nodes.

In the example below, two records are set for the container CTabSheet, and a record is set with the container type 'silverdev program'

Convert screen

General level Components (14) Dynamic properties (2) Messages (5) DSPF01 FMT1 DSPF01 FMT2 DSPF02 FTM1

| Record | Source           | Component name             | Component type     | Number of dynamic properties |
|--------|------------------|----------------------------|--------------------|------------------------------|
| FMT1   | Record           | pageCtrlDSPF01CPageControl |                    | 0                            |
| FMT1   | Record           | tsFMT1                     | CTabSheet          | 0                            |
| FMT1   | Field constante1 | constante1                 | CLabel             | 0                            |
| FMT1   | Field FLD001     | FLD001                     | CEditNum           | 0                            |
| FMT1   | Field FLD002     | FLD002                     | CEdit              | 0                            |
| FMT1   | Field FLD003     | FLD003                     | ComboBox           | 0                            |
| FMT1   | Field FLD004     | FLD004                     | CDateEdit          | 0                            |
| FMT1   | Field FLD005     | FLD005                     | CMemo              | 0                            |
| FMT2   | Record           | tsFMT2                     | CTabSheet          | 0                            |
| FMT2   | Field FLD001     | FLD001_1                   | CEdit              | 0                            |
| FMT2   | Field FLD002     | FLD002_1                   | CEdit              | 0                            |
| FTM1   | Record           | *FORM                      | Silverdev program0 | 0                            |
| FTM1   | Field FLD001     | FLD001                     | CEditNum           | 1                            |
| FTM1   | Field FLD002     | FLD002                     | CEditNum           | 1                            |

Visual representation of the screen components and their dynamic properties:

- \*FORM
  - pageCtrlDSPF01
    - tsFMT1
      - constante1
      - FLD001
      - FLD002
      - FLD003
      - FLD004
      - FLD005
    - tsFMT2
      - FLD001\_1
      - FLD002\_1
  - \*FORM FTM1
    - FLD001
    - FLD002

### 'Program to system' fields

When hidden field or 'program to system' fields are declared in the record, an additional grid is displayed, allowing to determine whether variables are added in the rpg instructions generated.

Program fields

| Name    | Attribute         | Generate                            | Indicators |   |
|---------|-------------------|-------------------------------------|------------|---|
| MARKTXT | Program to system | <input checked="" type="checkbox"/> |            | 1 |
| CTLFLD1 | Hidden            | <input checked="" type="checkbox"/> |            | 2 |
| CTLFLD2 | Hidden            | <input checked="" type="checkbox"/> |            | 2 |
| CTLFLD3 | Hidden            | <input checked="" type="checkbox"/> |            | 2 |

Dependency

Field F1, choice 03, text &MARKTXT

Grid on the right displays the dynamic properties depending on this field.

## Components tabsheet

'Components' tabsheet summaries the components that will be created.

Two views exist. One is a grid view, the other one is a tree view.

In the grid, all components are displayed even the disabled ones (they are grayed)

In the tree view, only the enabled components are displayed.

When an item is selected in the grid, the corresponding item in the tree is automatically selected.

The screenshot shows the 'Convert screen' application window. The 'Components' tabsheet is active, displaying a table of components on the left and a tree view on the right.

| Record | Source           | Component name | Component type    | Number of dynamic properties |
|--------|------------------|----------------|-------------------|------------------------------|
| FMT1   | Record           | pageCtrlDSPF01 | CPageControl      | 0                            |
| FMT1   | Record           | tsFMT1         | CTabSheet         | 0                            |
| FMT1   | Field constante1 | constante1     | CLabel            | 0                            |
| FMT1   | Field FLD001     | FLD001         | CEditNum          | 0                            |
| FMT1   | Field FLD002     | FLD002         | CEdit             | 0                            |
| FMT1   | Field FLD003     | FLD003         | ComboBox          | 0                            |
| FMT1   | Field FLD004     | FLD004         | CDateEdit         | 0                            |
| FMT1   | Field FLD005     | FLD005         | CMemo             | 0                            |
| FMT2   | Record           | tsFMT2         | CTabSheet         | 0                            |
| FMT2   | Field FLD001_1   | FLD001_1       | CEdit             | 0                            |
| FMT2   | Field FLD002_1   | FLD002_1       | CEdit             | 0                            |
| FTM1   | Record           | *FORM          | Silverdev program | 0                            |
| FTM1   | Field FLD001     | FLD001         | CEditNum          | 1                            |
| FTM1   | Field FLD002     | FLD002         | CEditNum          | 1                            |

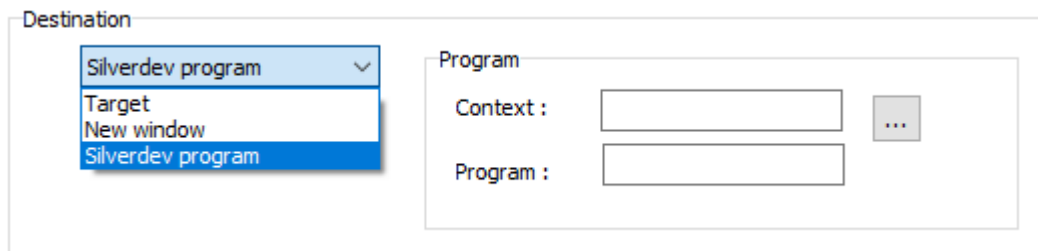
The tree view on the right shows the hierarchy of components. The selected component, FLD002, is highlighted in the tree view.

---

## Destination

In 'General' tab destination area allows to choose among three possible destinations.

- Silverdev program. In this case, you must chose the 'context' and the name of the program.
- New window. In this case, a new window will be created.
- Target. In this case, components will be added to current designed window.  
If no window is in design , an error message will be displayed( See chapter on messages)



The screenshot shows a 'Destination' dialog box. On the left, a dropdown menu is open, displaying four options: 'Silverdev program', 'Target', 'New window', and 'Silverdev program'. To the right, there is a 'Program' section containing two text input fields labeled 'Context' and 'Program', and a button with three dots '...' next to the 'Context' field.

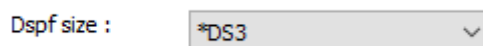
---

## Dspf Size

Size area allows to choose used information to determine created components positions.

Created SilverDev components will be positioned in ratio. For example, if a field is at row 5 and \*DS3 is selected (\*DS3 = 20 rows) , then corresponding component will be placed at 1/5 of target height.

If \*DS4 is selected, and some formats don't have any information about this size, a warning message will be added and \*DS3 information will be used.



The screenshot shows a label 'Dspf size :' followed by a dropdown menu. The dropdown menu is open, showing the selected option '\*DS3'.

## Dynamic properties

In dsp screens, some fields are indicator dependents, or variable dependents. For this fields, rpg code is generated in order to get the same behavior.

These instructions are displayed in tabsheet 'dynamic properties'

A grid displays all dynamic properties, and below, the corresponding code is displayed.

A column displays list of fields that the dynamic property is dependent of, another column displays the list of indicators that the dynamic property is dependent of.

When a dynamic property is selected in the grid, the corresponding code is signaled by a green arrow.

| Record | Dynamic properties       | Dependency field | Indicators |
|--------|--------------------------|------------------|------------|
| FTM1   | Component FLD001 visible |                  | N05 08     |
| FTM1   | Component FLD002 visible |                  | 05         |

```

D clearResponses...
D                                     pr
  //Procedures implementations

P refreshDspf02Ftm1...
P                                     B
  /free
    sdSetBool(F1:'FLD001':'visible':(not *IN05 and *IN08));
    sdSetBool(F1:'FLD002':'visible':*IN05);
  /end-free
P refreshDspf02Ftm1...
P                                     E

P validForm...
P                                     B
  /free
  /end-free
P validForm
  
```

Note: When component generation is deactivated, corresponding dynamic properties are also deactivated and code is commented.

When a dynamic property is dependent of a field and that this field is deactivated, the dynamic property is deactivated.

The screenshot shows the 'Convert screen' window in SilverDev. The window has a toolbar with icons for running, saving, and other actions. Below the toolbar is a tabbed interface with tabs for 'General level', 'Components (11)', 'Dynamic properties (1)', 'Messages (5)', and several DSPF01 FMT tabs. The 'Dynamic properties (1)' tab is active, displaying a table with the following data:

| Record | Dynamic properties       | Dependency field | Indicators |
|--------|--------------------------|------------------|------------|
| FTM1   | Component FLD001 visible |                  | N05 08     |
| FTM1   | Component FLD002 visible |                  | 05         |

Below the table is a code editor showing the implementation of the dynamic properties. The code is as follows:

```

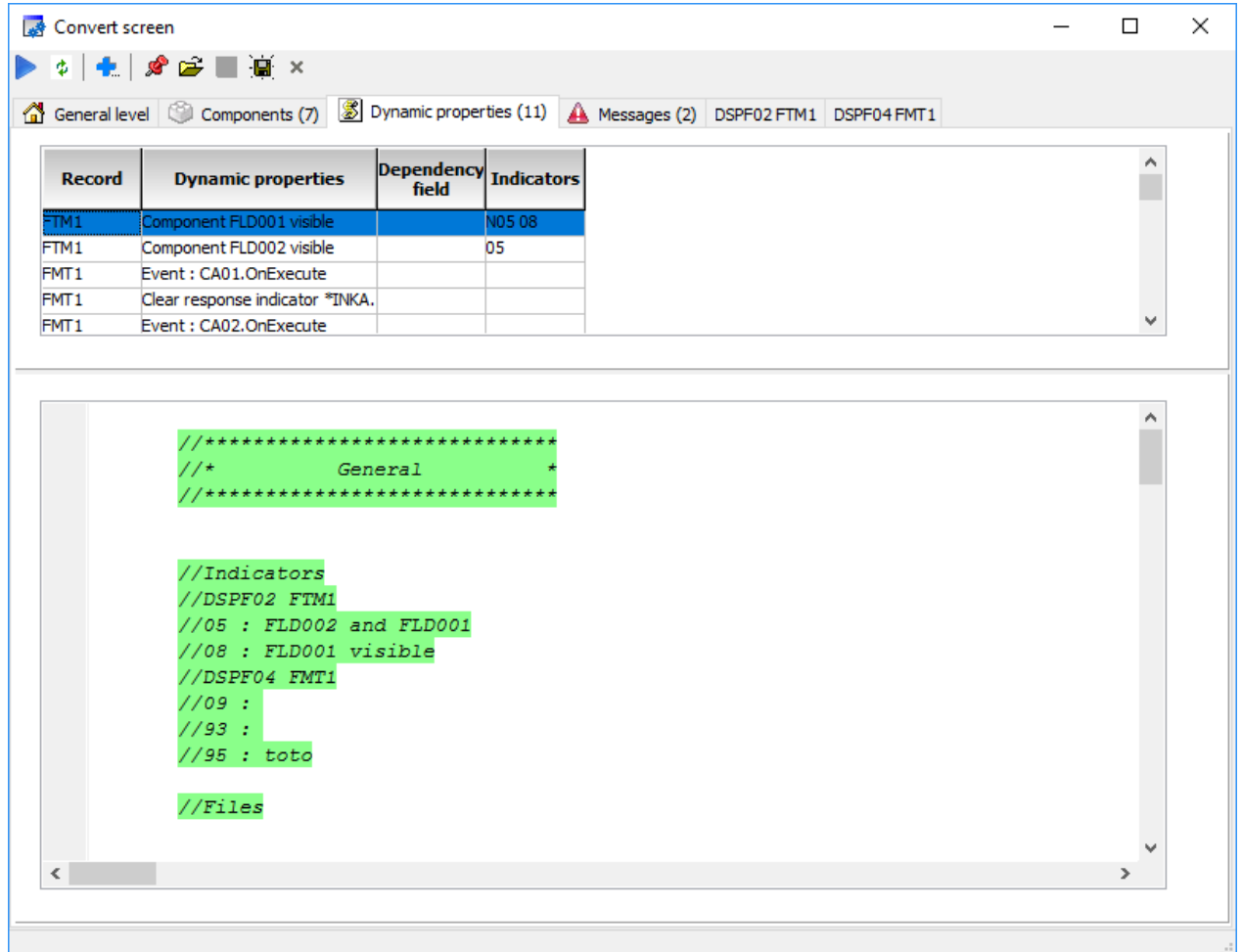
D validForm...
D                                     pr

D clearResponses...
D                                     pr
//Procedures implementations

P refreshDspf02Ftm1...
P                                     B
/free
//sdSetBool(F1:'FLD001':'visible':(not *IN05 and *IN08));
sdSetBool(F1:'FLD002':'visible':*IN05);
/end-free
P refreshDspf02Ftm1...
P                                     E
  
```

Note : For each format, a function is create, because in a 5250 rpg program, sometimes, an indicator is used for different purposes for each format.

Note : indicators text is added in comments



The screenshot shows the 'Convert screen' application window. The window has a menu bar with 'General level', 'Components (7)', 'Dynamic properties (11)', and 'Messages (2)'. Below the menu bar is a table with four columns: 'Record', 'Dynamic properties', 'Dependency field', and 'Indicators'. The table contains five rows of data. Below the table is a large text area containing comments for the dynamic properties and indicators.

| Record | Dynamic properties              | Dependency field | Indicators |
|--------|---------------------------------|------------------|------------|
| FTM1   | Component FLD001 visible        |                  | N05 08     |
| FTM1   | Component FLD002 visible        |                  | 05         |
| FMT1   | Event : CA01.OnExecute          |                  |            |
| FMT1   | Clear response indicator *INKA. |                  |            |
| FMT1   | Event : CA02.OnExecute          |                  |            |

```
//*****  
//*          General          *  
//*****  
  
//Indicators  
//DSPF02 FTM1  
//05 : FLD002 and FLD001  
//08 : FLD001 visible  
//DSPF04 FMT1  
//09 :  
//93 :  
//95 : toto  
  
//Files
```



## Links fields/Components/Dynamic properties

In 'Components' tabsheet, you can display the origin field. Select the component, right click and select 'origin':

| Record | Source                | Component name | Component type | Number of dynamic properties |
|--------|-----------------------|----------------|----------------|------------------------------|
| FTM1   | Field FLD001          | FLD001         | CEditNum       | 1                            |
| FTM1   | Field FLD002          | FLD002         | CEditNum       | 1                            |
|        |                       | actionList     | CActionList    | 0                            |
| FMT1   | Command key CA01      | CA01           | TAction        | 2                            |
| FMT1   | Command key C         |                |                | 2                            |
| FMT1   | Command key C         |                |                | 2                            |
| FMT1   | Command key C         |                |                | 3                            |
| FMT1   | Command key CLEAR     | CLEAR          | TAction        | 2                            |
| FMT1   | Command key VLDCMDKEY | VLDCMDKEY      | TAction        | 2                            |

Origin  
Propriétés dynamiques

In 'Components' tabsheet, you can detect dynamic properties corresponding to this component.

Select the component, right click and click on 'dynamic properties'.

Note: A component can have no dynamic property. The number of dynamic properties for a component is displayed in column 'number of dynamic properties'

In 'Dyanmic properties' tabsheet, you can retrieve the corresponding component or the origin field with a right click.

|      |                                 |  |     |
|------|---------------------------------|--|-----|
| FMT1 | Event : CA01.OnExecute          |  |     |
| FMT1 | Clear response indicator *INKA  |  |     |
| FMT1 | Event : CA02.OnExecute          |  |     |
| FMT1 | Clear response indicator *IN92. |  |     |
| FMT1 | Event : CA03.OnExecute          |  |     |
| FMT1 | Clear response indicator *IN93. |  |     |
| FMT1 | Command key CA04                |  | N09 |

Component  
Origin

## 'Messages' tabsheet

When problems occur, messages are displayed.

There are two types of messages, errors and warnings. Errors are in red and it is impossible to launch conversion while there are errors.

Warnings are displayed in black and don't prevent conversion to be done.

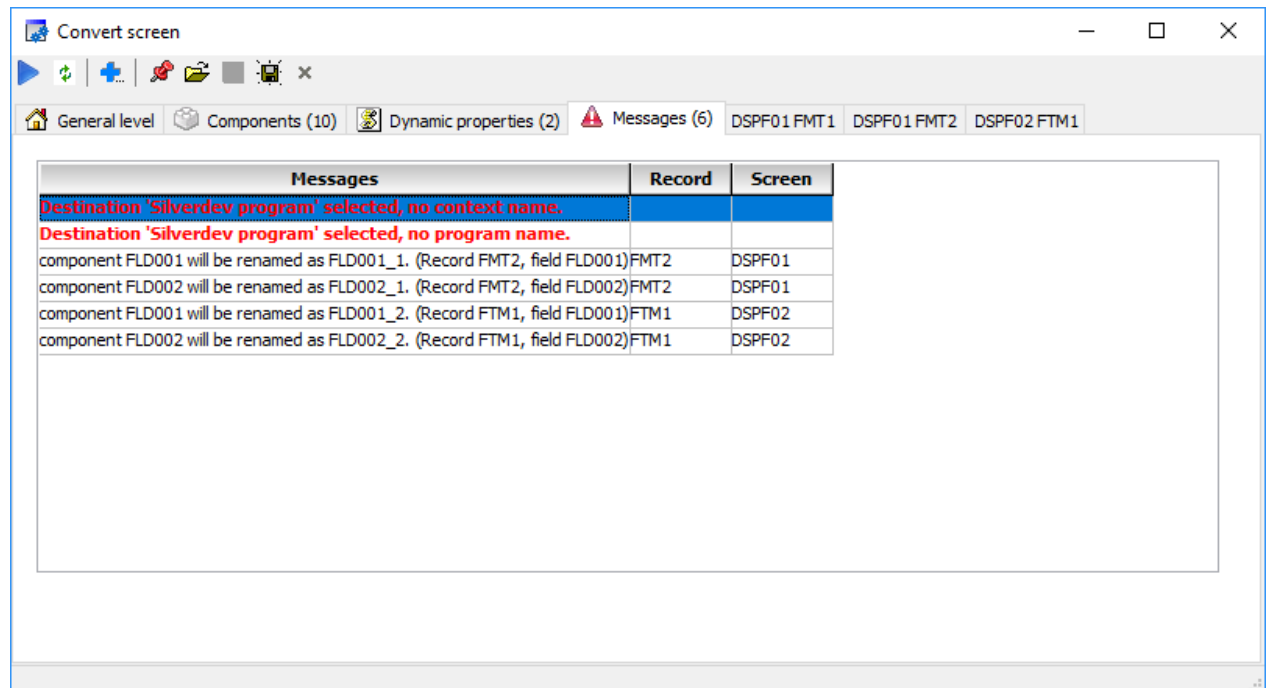
Among possible errors :


'Target' selected for destination, and no window currently in design.

'Silverdev program' selected for destination, but 'context' is not set , or program is not set, or program already exists, or a source already exists for the rpg or for the screen.

Among possible warnings:

Two formats have a field with same name. The two components that will be create can't have the same name. One of them will be renamed.



Note : if there is an error because the program already exists, and you delete this program, you can click on  to remove the error.

## Genération

Generation is done with button 

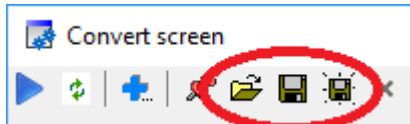
Generation can be launched only if there are no error messages.

If selected destination is 'silverdev program', or if a format is configured with container 'silverdev program', rpg sources and screen sources are created, they are open in the designer.

## Save/Open

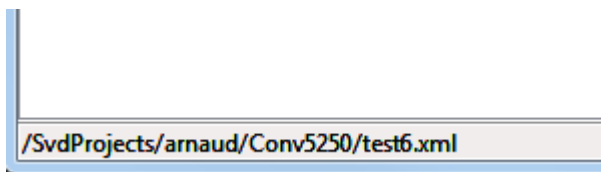
You can save your work, and reopen it.  
Data are saved in xml.

Use following buttons



Note: Second button is activated only if a document is open.

Document name is then displayed at the bottom of the main window.



## Concrete cases

### Several screens or several formats

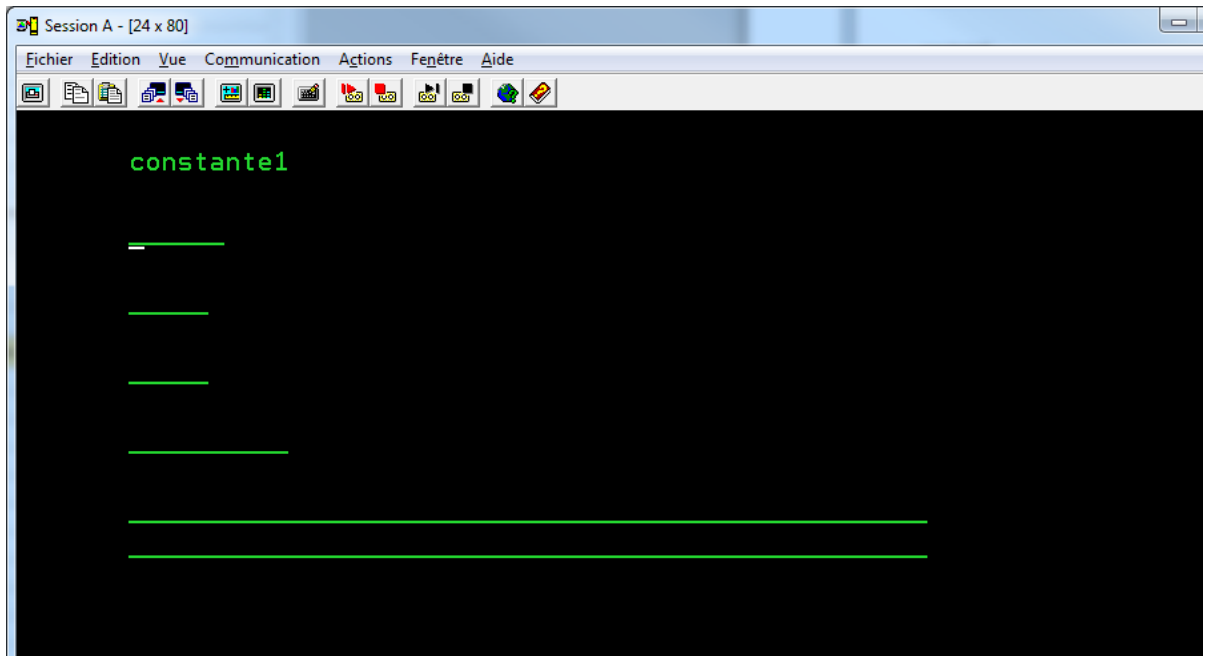
When several formats are selected, all formats can be added to same silverdev screen.  
You can chose to put each format in a tabsheet.

#### Screen dds

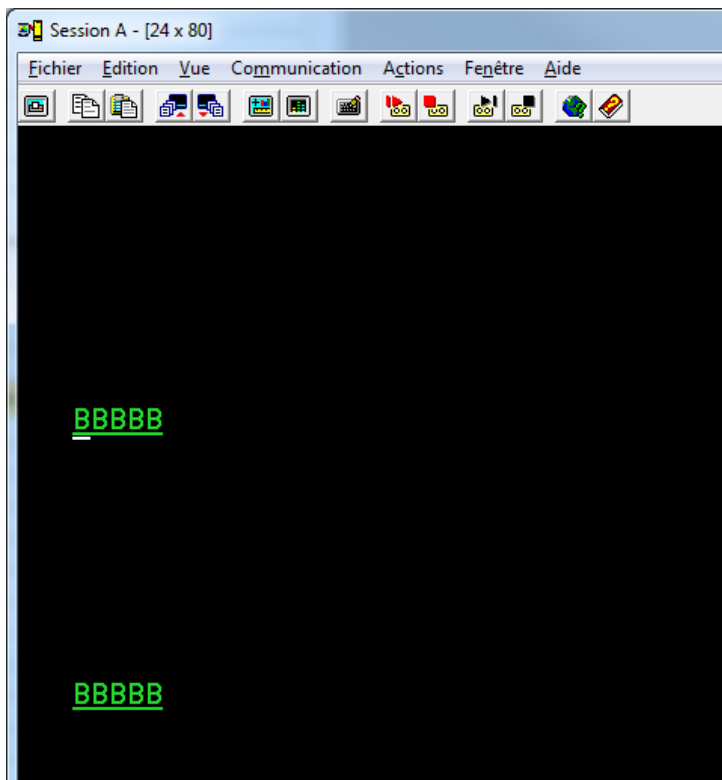
|   |        |        |    |    |                                 |
|---|--------|--------|----|----|---------------------------------|
| A |        |        |    |    | DSPSIZ (24 80 *DS3 27 132 *DS4) |
| A | R FMT1 |        |    |    |                                 |
| A |        |        | 2  | 8  | 'constante1'                    |
| A | FLD001 | 5S 0I  | 4  | 8  | TEXT('toto')                    |
| A | FLD002 | 5A I   | 6  | 8  |                                 |
| A | FLD003 | 5A I   | 8  | 8  | VALUES('a' 'b' 'c')             |
| A | FLD004 | L I    | 10 | 8  |                                 |
| A | FLD005 | 100A I | 12 | 8  | CNTFLD (50)                     |
| A | R FMT2 |        |    |    |                                 |
| A | FLD001 | 5A B   | 8  | 4  |                                 |
| A | FLD002 | 5A B   | 15 | 4  |                                 |
| A | *DS4   |        | 15 | 12 |                                 |
| A |        |        |    |    |                                 |

**5250 screen**

Format FMT 1:



Format FMT 2 :



Here, 'tabsheet' is selected as container for each format.

### Components tabsheet

The screenshot shows the 'Convert screen' window in SilverDev. The window has a toolbar with icons for running, adding, deleting, and saving. Below the toolbar is a tabbed interface with the following tabs: 'General level', 'Components (11)', 'Dynamic properties (0)', 'Messages (4)', 'DSPF01 FMT1', and 'DSPF01 FMT2'. The 'Components (11)' tab is active, displaying a table with the following data:

| Record | Source           | Component name              | Component type | Number of dynamic properties |
|--------|------------------|-----------------------------|----------------|------------------------------|
| FMT1   | Record           | pageCtrlDSPF01CPageControl0 |                |                              |
| FMT1   | Record           | tsFMT1                      | CTabSheet      | 0                            |
| FMT1   | Field constante1 | constante1                  | CLabel         | 0                            |
| FMT1   | Field FLD001     | FLD001                      | CEditNum       | 0                            |
| FMT1   | Field FLD002     | FLD002                      | CEdit          | 0                            |
| FMT1   | Field FLD003     | FLD003                      | ComboBox       | 0                            |
| FMT1   | Field FLD004     | FLD004                      | CDateEdit      | 0                            |
| FMT1   | Field FLD005     | FLD005                      | CMemo          | 0                            |
| FMT2   | Record           | tsFMT2                      | CTabSheet      | 0                            |
| FMT2   | Field FLD001     | FLD001_1                    | CEdit          | 0                            |
| FMT2   | Field FLD002     | FLD002_1                    | CEdit          | 0                            |

To the right of the table is a tree view of the form structure. The root node is '\*FORM'. It has a child node 'pageCtrlDSPF01', which in turn has a child node 'tsFMT1'. The 'tsFMT1' node has several children: 'constante1' (a label), 'FLD001' (a numeric edit), 'FLD002' (a text edit), 'FLD003' (a combobox), 'FLD004' (a date edit), and 'FLD005' (a memo). Below 'tsFMT1' is another node 'tsFMT2', which has two children: 'FLD001\_1' and 'FLD002\_1' (both text edits).

**Final result :**

tsFMT1 tsFMT2

constante1

0

//

tsFMT1 tsFMT2

## Indicators

When developping with new silverdev programs it is not advised to work with indicators. Here we want to remake as fast as possible an existing 5250 program.

Conversion takes account as much as possible on indicators, so you can copy paste rpg code without having to get into details of meaning of indicators.

### DDS 5250

```

A                                DSPSIZ(24 80 *DS3)
A                                INDTEXT(05 'FLD002 and FLD001')
A                                R FTM1
A                                INDTEXT(08 'FLD001 visible')
A N05 08      FLD001      5S 0I  4  8
A  05        FLD002      5S 0I  4  8

```

### Dynamic properties

The screenshot shows the 'Convert screen' window in SilverDev. The 'Dynamic properties (2)' tab is active, displaying a table with the following data:

| Record | Dynamic properties       | Dependency field | Indicators |
|--------|--------------------------|------------------|------------|
| FTM1   | Component FLD001 visible |                  | N05 08     |
| FTM1   | Component FLD002 visible |                  | 05         |

Below the table, the generated code is shown in a text editor. The code includes comments and procedure implementations for setting the visibility of FLD001 and FLD002 based on the indicators N05 and 08.

```

D                                pr
D                                //Procedures implementations

P refreshDspf02Ftm1...
P                                B
P                                /free
P                                sdSetBool(F1:'FLD001':'visible':(not *IN05 and *IN08));
P                                sdSetBool(F1:'FLD002':'visible':*IN05);
P                                /end-free
P refreshDspf02Ftm1...
P                                E

P validForm...

```

## reference fields

When a field in a dspf screen references a database field, informations are read in the database so the component matches the dspf screen field.

A dynamic property is added corresponding to the declaration of the file.

The type of the file, (read , write , add) will depend of the selected fields and their attributes.(  
input, output, both)

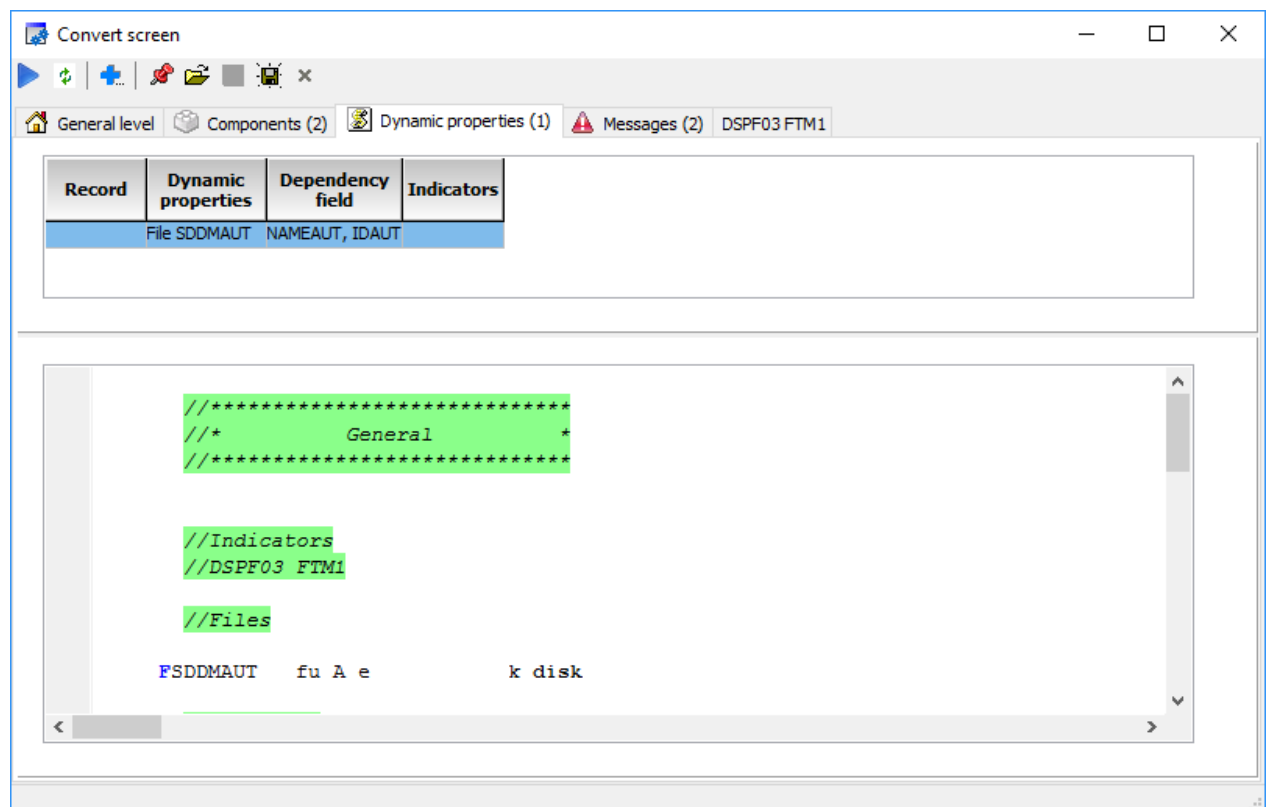
## Screen dds

```

A                                     DSPSIZ(24 80 *DS3)
A          R FTM1
A          NAMEAUT      R          B 16 29REFFLD (FAUTHORS/NAMEAUT
SILVERDEMO/-
A                                     SDDMAUT)
A          IDAUT      R          O 10 29REFFLD (FAUTHORS/IDAUT SDDMAUT)

```

## Dynamic properties





## function keys

For all function keys, CAXX/CFXX, a component of TAction type is created with the shortcut matching the function key FXX.

Several dynamic properties are created.

An OnExecute event is added on the TAction component.

In this event, the matching response indicator is set to \*on and all other response indicators are set to \*off.

## Screen dds

|       |                                |
|-------|--------------------------------|
| A     | VLDCMDKEY (90 'Any valid key') |
| A     | CA01                           |
| A     | CA02 (92)                      |
| A     | R FMT1                         |
| A     | CA03 (93)                      |
| A N09 | CA04 (95 'toto')               |
| A     | CLEAR (95)                     |

## Components tabSheet

The screenshot shows the 'Convert screen' window with the following components table:

| Record | Source                | Component name | Component type | Number of dynamic properties |
|--------|-----------------------|----------------|----------------|------------------------------|
|        |                       | actionList     | CActionList    | 0                            |
| FMT1   | Command key CA01      | CA01           | TAction        | 2                            |
| FMT1   | Command key CA02      | CA02           | TAction        | 2                            |
| FMT1   | Command key CA03      | CA03           | TAction        | 2                            |
| FMT1   | Command key CA04      | CA04           | TAction        | 3                            |
| FMT1   | Command key CLEAR     | CLEAR          | TAction        | 2                            |
| FMT1   | Command key VLDCMDKEY | VLDCMDKEY      | TAction        | 2                            |

On the right, the tree view shows the screen structure:

- \*FORM
  - actionList
    - CA01
    - CA02
    - CA03
    - CA04

## Checkboxes

### Screen dds

|       |          |                     |  |                                    |
|-------|----------|---------------------|--|------------------------------------|
| A     |          |                     |  | DSPSIZ(24 80 *DS3)                 |
| A     | R RECORD |                     |  |                                    |
| A     | F1       | 2Y 0B 3 35MLTCHCFLD |  |                                    |
| A N01 |          |                     |  | CHOICE(1 '>Undo')                  |
| A     |          |                     |  | CHOICE(2 &MARKTXT)                 |
| A     |          |                     |  | CHOICE(3 '>Copy')                  |
| A     |          |                     |  | CHCCTL(1 &CTLONE MSG1111 QUSER/A)  |
| A     |          |                     |  | CHCCTL(2 &CTLTWO &MSG1 &LIB/&MSGF) |
| A     |          |                     |  | CHCCTL(3 &CTLTHREE)                |
| A     | CTLONE   | 1Y 0H               |  |                                    |
| A     | CTLTWO   | 1Y 0H               |  |                                    |
| A     | CTLTHREE | 1Y 0H               |  |                                    |
| A     | MSGF     | 10A P               |  |                                    |
| A     | LIB      | 10A P               |  |                                    |
| A     | MSG1     | 7A P                |  |                                    |
| A     | MARKTXT  | 10A P               |  |                                    |

### 5250 screen



For the F1 field, three checkbox components are created.  
In this case, a 5250 field generates several components.

## Format tabSheet

Format : ARNAUD/DSPF05 RECORD  
Record type : Record

Container : Target

Silverdev program  
Context :  
Program : RECORD

Program fields

| Name     | Attribute         | Generate                            | Indicators |   |
|----------|-------------------|-------------------------------------|------------|---|
| CTONE    | Hidden            | <input checked="" type="checkbox"/> |            | 2 |
| CTLTWO   | Hidden            | <input checked="" type="checkbox"/> |            | 2 |
| CTLTHREE | Hidden            | <input checked="" type="checkbox"/> |            | 2 |
| MSGF     | Program to system | <input checked="" type="checkbox"/> |            | 0 |

Components :

| Field name | Component type | Generate                            | Indicators | Attribute | Choice |
|------------|----------------|-------------------------------------|------------|-----------|--------|
| F1         | Check boxes    | <input checked="" type="checkbox"/> |            | Both      | 3      |

Sub items :

| Type      | Text     | Generate                            | Indicators | Choice number |
|-----------|----------|-------------------------------------|------------|---------------|
| CCheckBox | Undo     | <input checked="" type="checkbox"/> | N01        | 01            |
| CCheckBox | &MARKTXT | <input checked="" type="checkbox"/> |            | 02            |
| CCheckBox | Copy     | <input checked="" type="checkbox"/> |            | 03            |

Dependency

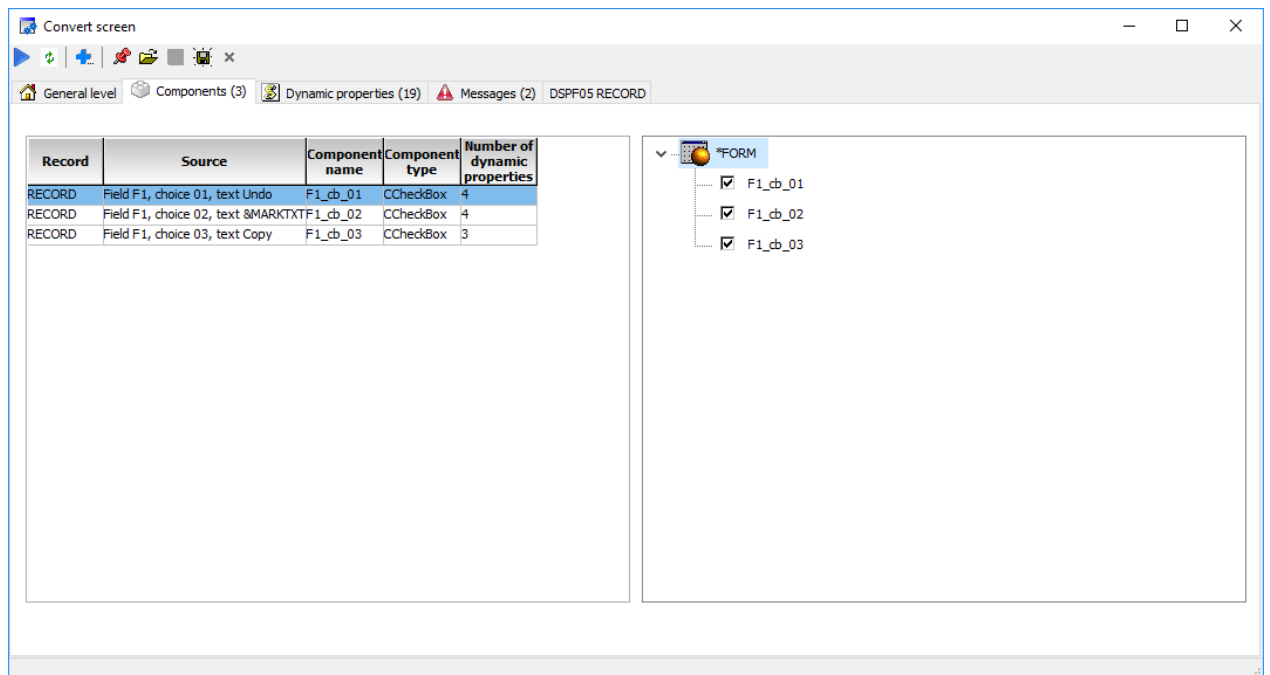
Field F1, choice 01, text Undo, control field in.  
Field F1, choice 01, text Undo, control field out.

Note : When the text of a choice depends on a variable, the text starts with character '& '.

Sub items :

| Type      | Text     | Generate                            | Indicators | Choice number |
|-----------|----------|-------------------------------------|------------|---------------|
| CCheckBox | Undo     | <input checked="" type="checkbox"/> | N01        | 01            |
| CCheckBox | &MARKTXT | <input checked="" type="checkbox"/> |            | 02            |
| CCheckBox | Copy     | <input checked="" type="checkbox"/> |            | 03            |

## Components tabsheet



Several RPG instructins are generated in order to have the same behavior than in the 5250 program.

```

P refreshDspf05Record...
P          B
/free
sdSetBool(F1:'F1_cb_01':'enabled':not *IN01);
  if sdGetBool(F1:F1_cb_01:checked);
    F1 = F1 + 1;
  endif;
sdSetBool(F1:'F1_cb_01':'checked':CTLONE = 1);
sdSetBool(F1:'F1_cb_01':'enabled':CTLONE <= 1);
if CTLONE = 2;
  sdSetFocus(F1:F1_cb_01)
endif;
sdSetString(F1:'F1_cb_02':'caption':MARKTXT);
  if sdGetBool(F1:F1_cb_02:checked);
    F1 = F1 + 1;
  endif;
sdSetBool(F1:'F1_cb_02':'checked':CTLTWO = 1);
sdSetBool(F1:'F1_cb_02':'enabled':CTLTWO <= 1);
if CTLTWO = 2;
  sdSetFocus(F1:F1_cb_02)
endif;
  if sdGetBool(F1:F1_cb_03:checked);
    F1 = F1 + 1;
  endif;
sdSetBool(F1:'F1_cb_03':'checked':CTLTHREE = 1);
sdSetBool(F1:'F1_cb_03':'enabled':CTLTHREE <= 1);
if CTLTHREE = 2;
  sdSetFocus(F1:F1_cb_03)
endif;
/end-free
P refreshDspf05Record...
P          E
P validForm...
P          B
/free
  if sdGetBool(F1:F1_cb_01:'checked');
    CTLONE = 1;
  else;
    CTLONE = 0;
  endif
  if sdGetBool(F1:F1_cb_02:'checked');
    CTLTWO = 1;
  else;
    CTLTWO = 0;
  endif
  if sdGetBool(F1:F1_cb_03:'checked');
    CTLTHREE = 1;
  else;
    CTLTHREE = 0;
  endif
/end-free
P validForm...
P          E

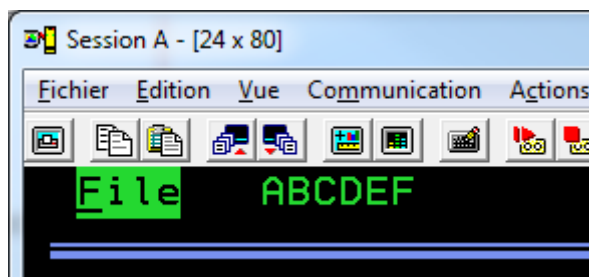
```

## Menus

### Screen DDS

|    |       |         |     |    |   |                           |
|----|-------|---------|-----|----|---|---------------------------|
| A  | R     | MNURCD  |     |    |   | MNUBAR                    |
| A  |       |         |     |    |   | MNUBARDSP                 |
| A  |       | MNUFLD  | 2Y  | 0B | 1 | 2                         |
| A  | 05    |         |     |    |   |                           |
| AO | 06N07 |         |     |    |   | MNUBARHC(1 PULL1 '>File ' |
| -  |       |         |     |    |   |                           |
| A  |       |         |     |    |   | &RTNFLD)                  |
| A  | 08    |         |     |    |   | MNUBARHC(2 PULL2 &MARKTXT |
| -  |       |         |     |    |   |                           |
| A  |       |         |     |    |   | &RTNFLD)                  |
| A  |       | MARKTXT | 10A | P  |   |                           |
| A  |       | RTNFLD  | 2Y  | 0H |   |                           |
| A  |       | R PULL1 |     |    |   | PULLDOWN                  |
| A  |       | R PULL2 |     |    |   | PULLDOWN                  |

### 5250 screen



### Components tabsheet

Convert screen

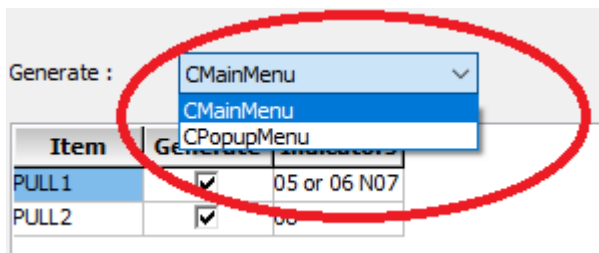
General level Components (3) Dynamic properties (7) Messages (2) DSPF06 MNURCD

| Record | Source                                 | Component name | Component type | Number of dynamic properties |
|--------|--|----------------|----------------|------------------------------|
| MNURCD | Record                                 | MainMenuMNURCD | CMainMenu      | 0                            |
| MNURCD | Field MNUFLD, choice 01, text File     | miMNUFLD_01    | TMenuItem      | 2                            |
| MNURCD | Field MNUFLD, choice 02, text &MARKTXT | miMNUFLD_02    | TMenuItem      | 3                            |

FORM

- MainMenuMNURCD
  - miMNUFLD\_01
  - miMNUFLD\_02

By default, a CMainMenu component is suggested. You can chose a component of type CPopupMenu.



Some instructions are generated to get the original behavior.

```

D MARKTXT...
D          s          10
D RTNFLD...
D          s          2  0
...
P refreshDspf06Mnurcd...
P          B
/free
sdSetBool(F1:'miMNUFLD_01':'visible':(*IN05 or (*IN06 and not *IN07)));
sdSetBool(F1:'miMNUFLD_02':'visible':*IN08);
sdSetString(F1:'miMNUFLD_02':'caption':MARKTXT);
/end-free
P refreshDspf06Mnurcd...
P          E
...
*/EVENT miMNUFLD_01
/free
RTNFLD=01;
/end-free

*/EVENT miMNUFLD_02
/free
RTNFLD=02;
/end-free

```

## Subfiles

### Screen dds

|      |           |         |                        |
|------|-----------|---------|------------------------|
| A    | R SFL1    |         | SFL                    |
| A    | FLD001    | 5 O 10  | 3                      |
| A    | FLD002    | 5 20 12 | 3                      |
| A    | FLD003    | 5 I 14  | 3VALUES('a' 'b')       |
| A    | FLD004    | 5 20 16 | 3TEXT('Header fld004') |
| A    | FLD005    | 5 2H    |                        |
| A    | R SFLCTL1 |         | SFLCTL(SFL1)           |
| A 03 |           |         | CF04                   |
| A    |           |         | SFLDSP                 |
| A    |           |         | SFLSIZ(&FLD002)        |
| A    |           |         | SFLPAG(0001)           |
| A    | FLD002    | 5S 0P   |                        |

A CSFL component is suggested to convert the sfl type format.

For each field of sfl format, a column is suggested in the CSFL component.

### Format tabsheet

Format : **ARNAUD/DSPF07 SFL1**  
Record type : **SFL**

Container : Target

Silverdev program  
Context :   
Program : SFL1

Program fields

| Name   | Attribute | Generate                            | Indicators |   |
|--------|-----------|-------------------------------------|------------|---|
| FLD001 | Output    | <input checked="" type="checkbox"/> |            | 1 |

Dependency  
List control field

Generate : CSFL

Columns :

| Field name | Column type | Generate                            | Indicators | Attribute |
|------------|-------------|-------------------------------------|------------|-----------|
| FLD002     | csSimple    | <input checked="" type="checkbox"/> |            | Output    |
| FLD003     | csPicklist  | <input checked="" type="checkbox"/> |            | Input     |
| FLD004     | csSimple    | <input checked="" type="checkbox"/> |            | Output    |
| FLD005     | csSimple    | <input checked="" type="checkbox"/> |            | Hidden    |

### Components tabsheet

| Record  | Source                    | Component name | Component type | Number of dynamic properties |
|---------|---------------------------|----------------|----------------|------------------------------|
| SFL1    | Record                    | SFL1           | CSFL           | 0                            |
| SFL1    | Record SFL1, field FLD002 | FLD002         | TSRColumn      | 0                            |
| SFL1    | Record SFL1, field FLD003 | FLD003         | TSRColumn      | 0                            |
| SFL1    | Record SFL1, field FLD004 | FLD004         | TSRColumn      | 0                            |
| SFL1    | Record SFL1, field FLD005 | FLD005         | TSRColumn      | 0                            |
|         |                           | actionList     | ActionList     | 0                            |
| SFLCTL1 | Command key CF04          | CF04           | TAction        | 3                            |

Component structure tree:

- \*FORM
  - SFL1
    - actionList
      - CF04

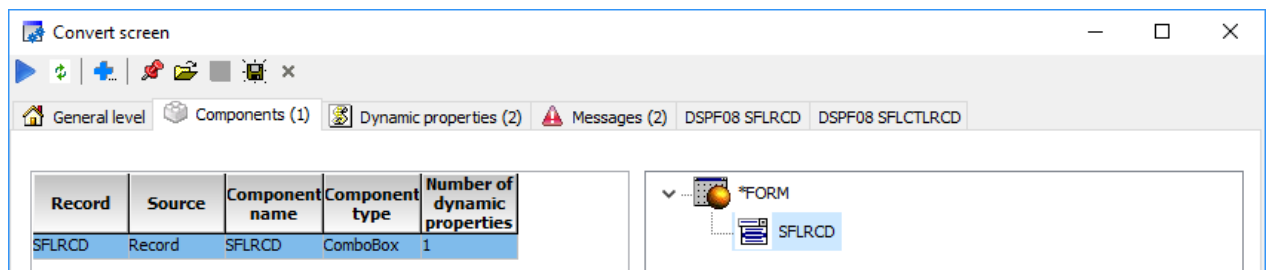


## Subfiles simple choice

### Screen dds

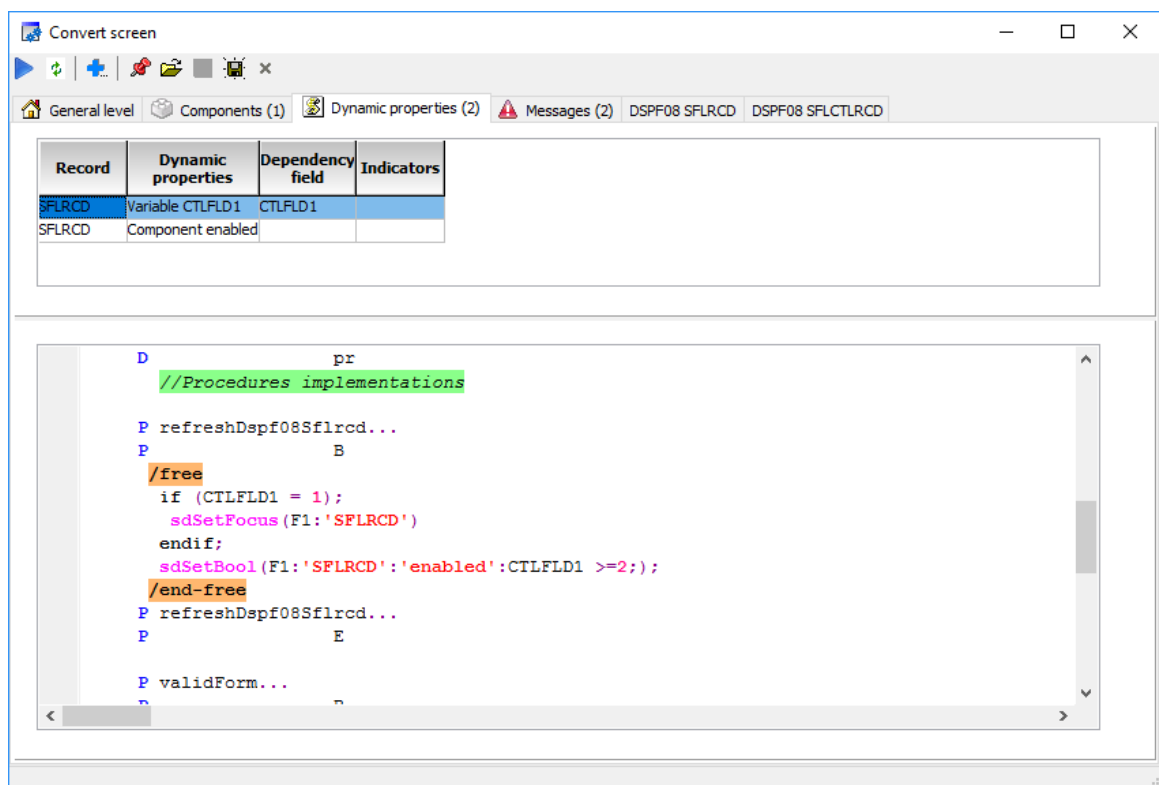
|   |   |           |    |    |   |                            |
|---|---|-----------|----|----|---|----------------------------|
| A | R | SFLRCD    |    |    |   | SFL                        |
| A |   | CTLFLD1   | 1Y | 0H |   | SFLCHCCTL                  |
| A |   | F31       | 5A | O  | 1 | 1                          |
| A | R | SFLCTLRCD |    |    |   | SFLCTL(SFLRCD)             |
| A |   |           |    |    |   | SFLSNGCHC                  |
| A |   |           |    |    |   | SFLPAG(5) SFLSIZ(6)        |
| A |   |           |    |    |   | SFLDSP                     |
| A |   |           |    |    |   | WINDOW(1 2 5 11 *NOMSGLIN) |
| A |   |           |    |    |   | SFLDSPCTL                  |

## Components tabsheet



| Record | Source | Component name | Component type | Number of dynamic properties |
|--------|--------|----------------|----------------|------------------------------|
| SFLRCD | Record | SFLRCD         | ComboBox       | 1                            |

## Dynamic properties tabsheet



| Record | Dynamic properties | Dependency field | Indicators |
|--------|--------------------|------------------|------------|
| SFLRCD | Variable CTLFLD1   | CTLFLD1          |            |
| SFLRCD | Component enabled  |                  |            |

```

D      pr
//Procedures implementations
P refreshDspf08Sflrcd...
P      B
/free
if (CTLFLD1 = 1);
sdSetFocus(F1:'SFLRCD')
endif;
sdSetBool(F1:'SFLRCD':'enabled':CTLFLD1 >=2);
/end-free
P refreshDspf08Sflrcd...
P      E

P validForm...

```

## Subfile multiple choice

### Screen dds

|   |   |            |    |    |   |                       |
|---|---|------------|----|----|---|-----------------------|
| A | R | SFLRCDM    |    |    |   | SFL                   |
| A |   | CTLFLD2    | 1Y | 0H |   | SFLCHCCTL             |
| A |   | F31        | 5A | O  | 1 | 1                     |
| A | R | SFLCTLRCDM |    |    |   | SFLCTL (SFLRCDM)      |
| A |   |            |    |    |   | SFLMLTCHC             |
| A |   |            |    |    |   | SFLPAG (5) SFLSIZ (6) |
| A |   |            |    |    |   | SFLDSP                |
| A |   |            |    |    |   | SFLDSPCTL             |

In case of a multiple choice sfl, it's the Listbox type that is selected by default.

Generate : ListBox

Columns :

| Field name | Column type | Generate                            | Indicators | Attribute |
|------------|-------------|-------------------------------------|------------|-----------|
| F31        | csSimple    | <input checked="" type="checkbox"/> |            | Output    |

### Components tabsheet

Convert screen

General level Components (1) Dynamic properties (2) Messages (2) DSPF09 SFLRCDM DSPF09 SFLCTLRCDM

| Record  | Source | Component name | Component type | Number of dynamic properties |
|---------|--------|----------------|----------------|------------------------------|
| SFLRCDM | Record | SFLRCDM        | Listbox        | 1                            |

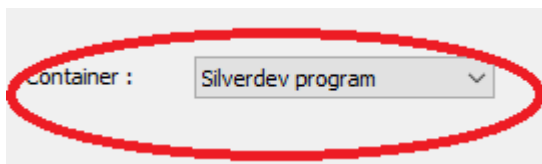
\*FORM

SFLRCDM

**window****Screen dds**

|   |     |           |     |    |   |    |       |  |                         |
|---|-----|-----------|-----|----|---|----|-------|--|-------------------------|
| A |     |           |     |    |   |    |       |  | DSPSIZ(24 80 *DS3)      |
| A |     | R RECORD1 |     |    |   |    |       |  |                         |
| A |     | FLD001    | 3S  | 0B | 3 | 27 |       |  |                         |
| A |     | FLD002    | 5A  | B  | 5 | 27 |       |  |                         |
| A |     |           |     |    | 3 | 13 | 'abc' |  |                         |
| A |     |           |     |    | 5 | 13 | 'def' |  |                         |
| A |     | R RECORD2 |     |    |   |    |       |  |                         |
| A |     |           |     |    |   |    |       |  | WINDOW(&LINE &POS 9 30) |
| A | N08 |           |     |    |   |    |       |  | WDWTITLE(*TEXT &TTL2)   |
| A |     | LINE      | 2S  | 0P |   |    |       |  |                         |
| A |     | POS       | 3S  | 0P |   |    |       |  |                         |
| A |     | FIELD3    | 5A  | B  | 4 | 9  |       |  |                         |
| A |     | FIELD4    | 20A | B  | 6 | 9  |       |  |                         |
| A |     | TTL2      | 10A | P  |   |    |       |  |                         |

In case of a format of window type, the pre-selected value for container area is 'Silverdev program'

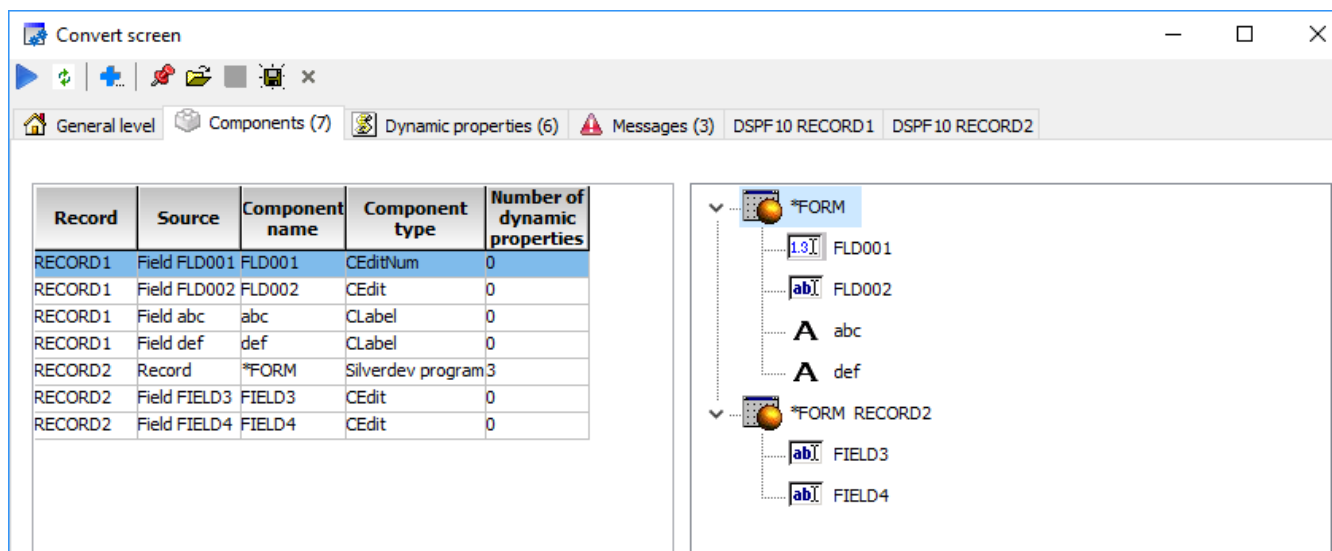


This means that components will be added in another window as we can see in the tree view.

The generated program will be with a window of type modal.

This behavior is the closest to a window type format.

**Components tabsheet**



The screenshot shows the 'Convert screen' tool interface. On the left, a table lists dynamic properties for various records. On the right, a tree view shows the hierarchical structure of the screen components.

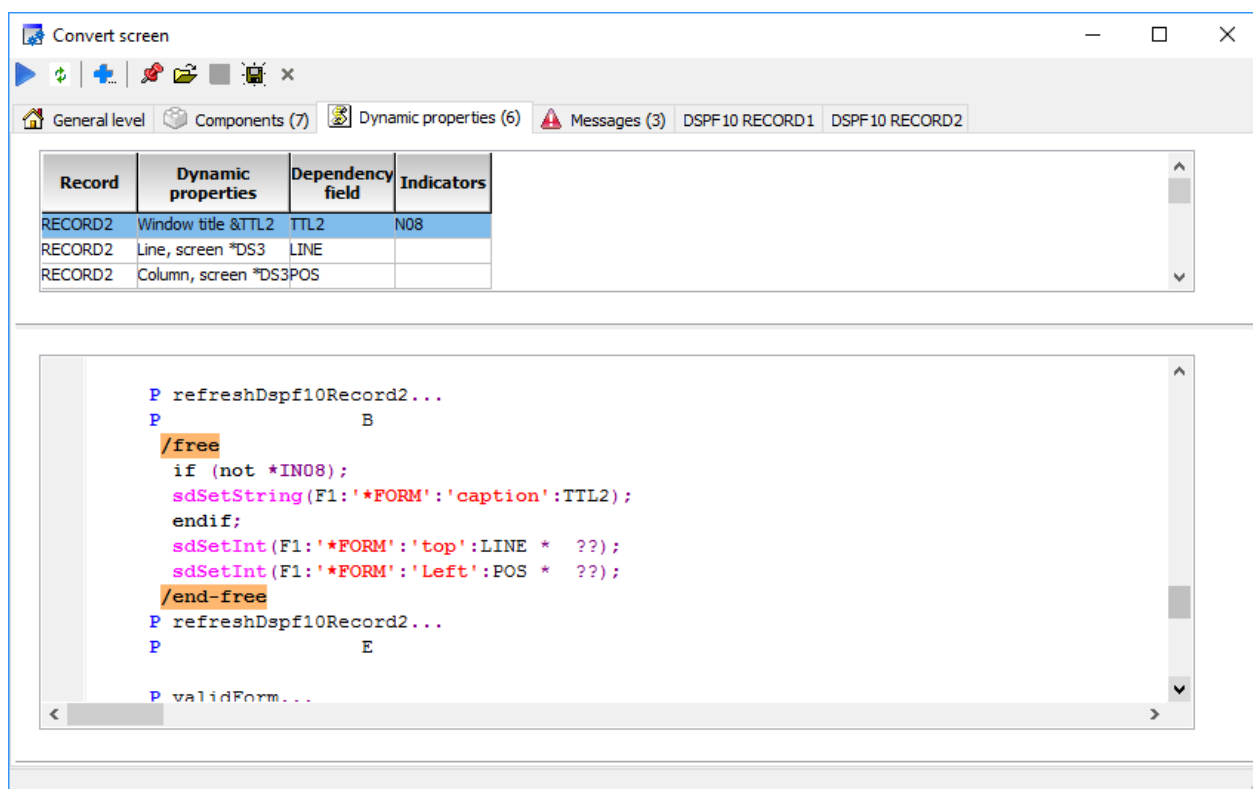
| Record  | Source       | Component name | Component type    | Number of dynamic properties |
|---------|--------------|----------------|-------------------|------------------------------|
| RECORD1 | Field FLD001 | FLD001         | CEditNum          | 0                            |
| RECORD1 | Field FLD002 | FLD002         | CEdit             | 0                            |
| RECORD1 | Field abc    | abc            | CLabel            | 0                            |
| RECORD1 | Field def    | def            | CLabel            | 0                            |
| RECORD2 | Record       | *FORM          | Silverdev program | 3                            |
| RECORD2 | Field FIELD3 | FIELD3         | CEdit             | 0                            |
| RECORD2 | Field FIELD4 | FIELD4         | CEdit             | 0                            |

The tree view on the right shows the following structure:

- \*FORM
  - FLD001
  - FLD002
  - abc
  - def
- \*FORM RECORD2
  - FIELD3
  - FIELD4

Some instructions are added to simulate the behavior of dynamic properties. Question marks visible in the screen capture below will be replaced by a constant when generating, when ratio between 5250 screen size and silverdev screen size will be known.

### Dynamic properties tabsheet.



The screenshot shows the 'Convert screen' tool interface. On the left, a table lists dynamic properties for various records. On the right, a code editor shows the generated code for the dynamic properties.

| Record  | Dynamic properties     | Dependency field | Indicators |
|---------|------------------------|------------------|------------|
| RECORD2 | Window title &TTL2     | TTL2             | N08        |
| RECORD2 | Line, screen *DS3      | LINE             |            |
| RECORD2 | Column, screen *DS3POS |                  |            |

```

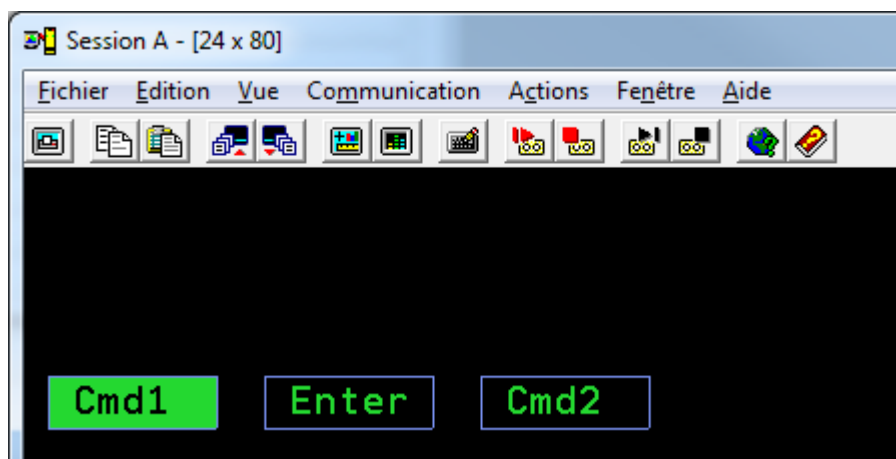
P refreshDspf10Record2...
P
  B
  /free
  if (not *IN08);
  sdSetString(F1:*FORM': 'caption':TTL2);
  endif;
  sdSetInt(F1:*FORM': 'top':LINE * ??);
  sdSetInt(F1:*FORM': 'Left':POS * ??);
  /end-free
P refreshDspf10Record2...
P
  E
P validForm...
  
```

## push buttons

### Screen dds

|       |          |         |  |  |  |
|-------|----------|---------|--|--|--|
| A     |          |         |  |  | DSPSIZ (24 80 *DS3)                    |
| A     | R RECORD |         |  |  |  |
| A     | F1       | 2Y 0B 5 |  |  | 2PSHBTNFLD ( (*NUMCOL 3) (*GUTTER 2) ) |
| A     |          |         |  |  | PSHBTNCHC (1 'Cmd1' )                  |
| A     |          |         |  |  | PSHBTNCHC (2 'Enter' )                 |
| A N06 |          |         |  |  | PSHBTNCHC (3 &MARKTXT)                 |
| A     |          |         |  |  | CHCCTL (1 &CTLBTN1 )                   |
| A     |          |         |  |  | CHCCTL (2 &CTLBTN2)                    |
| A     |          |         |  |  | CHCCTL (3 &CTLBTN3 )                   |
| A     | CTLBTN1  | 1Y 0H   |  |  |  |
| A     | CTLBTN2  | 1Y 0H   |  |  |  |
| A     | CTLBTN3  | 1Y 0H   |  |  |  |
| A     | MARKTXT  | 10A P   |  |  |  |

### 5250 screen



## Format tabsheet

Convert screen

General level Components (3) Dynamic properties (13) Messages (2) DSPF11 RECORD

Format : **ARNAUD/DSPF11 RECORD**  
Record type : **Record**

Container : Target

Silverdev program  
Context :   
Program : RECORD

Program fields

| Name    | Attribute         | Generate                            | Indicators |   |
|---------|-------------------|-------------------------------------|------------|---|
| CTLBTN1 | Hidden            | <input checked="" type="checkbox"/> |            | 2 |
| CTLBTN2 | Hidden            | <input checked="" type="checkbox"/> |            | 2 |
| CTLBTN3 | Hidden            | <input checked="" type="checkbox"/> |            | 2 |
| MARKTXT | Program to system | <input checked="" type="checkbox"/> |            | 1 |

Components :

| Field name | Component type | Generate                            | Indicators | Attribute | Choice |
|------------|----------------|-------------------------------------|------------|-----------|--------|
| F1         | Buttons        | <input checked="" type="checkbox"/> |            | Both      | 3      |

Sub items :

| Type | Text | Generate | Indicators | Cho |
|------|------|----------|------------|-----|
|------|------|----------|------------|-----|

Dependency

Field F1, choice 01, text Cmd1, control field in.  
Field F1, choice 01, text Cmd1, control field out.

## Components tabsheet

Convert screen

General level Components (3) Dynamic properties (13) Messages (2) DSPF11 RECORD

| Record | Source                               | Component name | Component type | Numl dyn prop |
|--------|--------------------------------------|----------------|----------------|---------------|
| RECORD | Field F1, choice 01, text Cmd1       | F1_btn_01      | CButton        | 2             |
| RECORD | Field F1, choice 02, text Enter      | F1_btn_02      | CButton        | 2             |
| RECORD | Field F1, choice 03, text &MARKTXTF1 | F1_btn_03      | CButton        | 4             |

FORM

- F1\_btn\_01
- F1\_btn\_02
- F1\_btn\_03

```

D CTLBTN1...
D                               S                1  0
D CTLBTN2...
D                               S                1  0
D CTLBTN3...
D                               S                1  0
D MARKTXT...
D                               S                10
D F1...
D                               S                2  0
...
P refreshDspfl1Record...
P                               B
  /free
  sdSetBool(F1:'F1_btn_01':'enabled':CTLBTN1 <= 1);
  if CTLBTN1 = 2;
    sdSetFocus(F1:F1_btn_01)
  endif;
  sdSetBool(F1:'F1_btn_02':'enabled':CTLBTN2 <= 1);
  if CTLBTN2 = 2;
    sdSetFocus(F1:F1_btn_02)
  endif;
  sdSetBool(F1:'F1_btn_03':'enabled':not *IN06);
  sdSetString(F1:'F1_btn_03':'caption':MARKTXT);
  sdSetBool(F1:'F1_btn_03':'enabled':CTLBTN3 <= 1);
  if CTLBTN3 = 2;
    sdSetFocus(F1:F1_btn_03)
  endif;
  /end-free
P refreshDspfl1Record...
P                               E
...
*/EVENT F1_btn_01
  F1 = 01;

*/EVENT F1_btn_02
  F1 = 02;

*/EVENT F1_btn_03
  F1 = 03;

```

## Radio buttons

### Screen dds

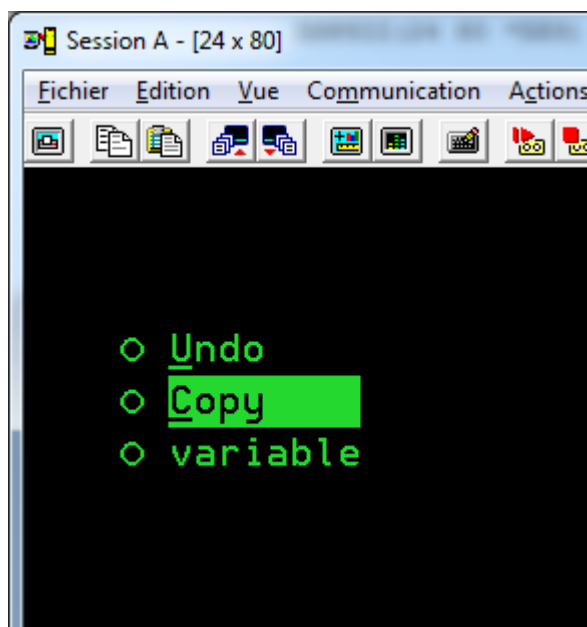
```

A                                     DSPSIZ(24 80 *DS3)
A      R OFTM1
A      F1                          2Y 0B  4 35SNGCHCFLD
A
A      CHOICE(1 '>Undo           ')
A      CHOICE(2 '>Copy           ')
A      CHOICE(3 &MARKTXT)
A      CHCCTL(1 &CTLFLD1 )
A      CHCCTL(2 &CTLFLD2 )
A      CHCCTL(3 &CTLFLD3 )

A      MARKTXT          10A  P
A      CTLFLD1          1Y 0H
A      CTLFLD2          1Y 0H
A      CTLFLD3          1Y 0H

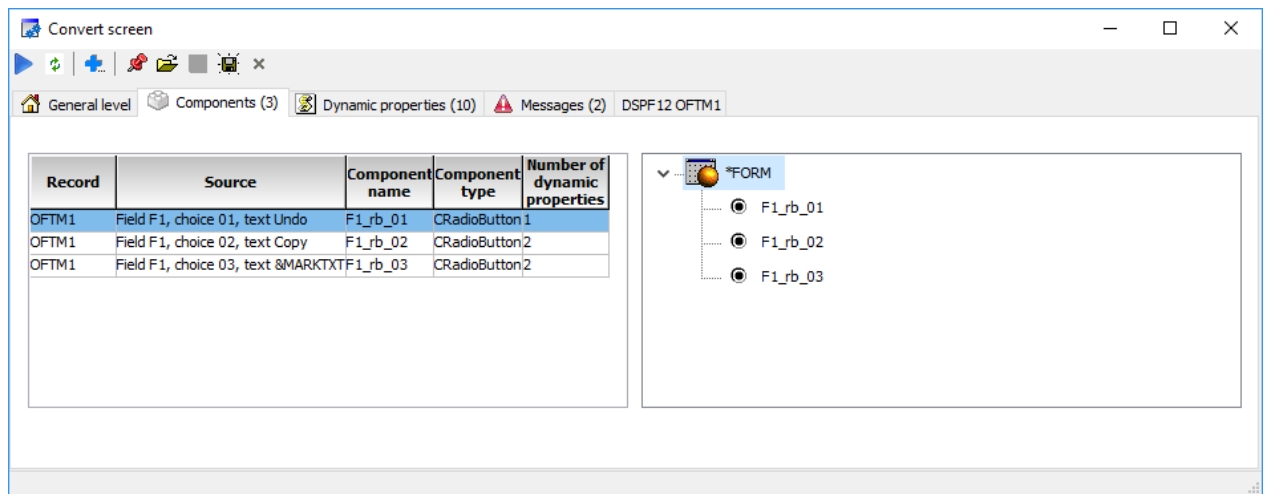
```

### 5250 screen

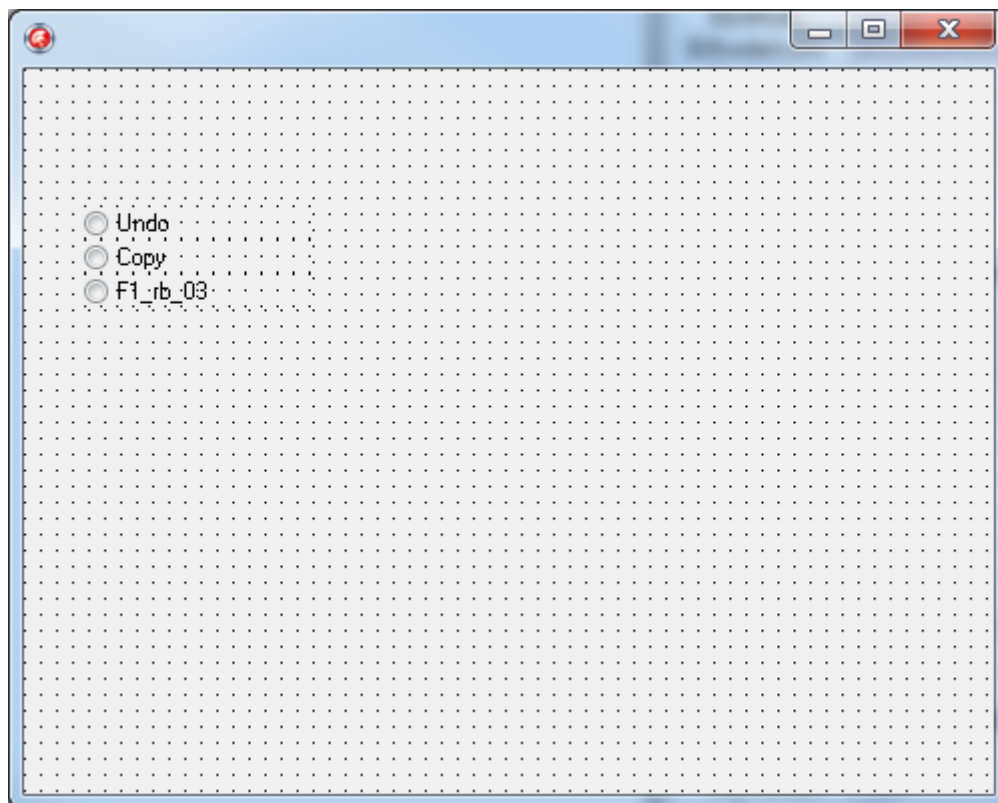




## Components tabsheet



## Final result



**msgid/msgcon****Screen dds**

|   |     |           |     |         |                                      |
|---|-----|-----------|-----|---------|--------------------------------------|
| A |     |           |     |         | DSPSIZ (24 80 *DS3 27 132 *DS4)      |
| A |     | R FMT1    |     |         |                                      |
| A | N25 |           |     | 2       | 1MSGCON(100 SVD0111                  |
| a |     |           |     |         | SILVERDEV/SILVDMMSGF)                |
| A |     |           |     |         |                                      |
| A | 01  | MSGFIELD2 | 10A | O 04 60 |                                      |
| A | 25  |           |     |         | MSGID (&MSGIDNUM &MSGFILENM)         |
|   |     |           |     |         |                                      |
| A | 26  |           |     |         | MSGID (SVD0112                       |
|   |     |           |     |         | SILVERDEV/&MSGFILENM)                |
|   |     |           |     |         |                                      |
| A | 27  |           |     |         | MSGID (SVD0112                       |
|   |     |           |     |         | &MSGFILELB/&MSGFILENM)               |
|   |     |           |     |         |                                      |
| A | 28  |           |     |         | MSGID (USR 0112 +                    |
| A |     |           |     |         | &MSGFILELB/&MSGFILENM)               |
|   |     |           |     |         |                                      |
| A | 29  |           |     |         | MSGID (USR &MSGNBR +                 |
| A |     |           |     |         | &MSGFILELB/&MSGFILENM)               |
|   |     |           |     |         |                                      |
| A | 30  |           |     |         | MSGID (*NONE)                        |
|   |     |           |     |         |                                      |
| A |     |           |     |         | MSGID (SVD0114 SILVERDEV/SILVDMMSGF) |
|   |     |           |     |         |                                      |
| A |     | MSGIDNUM  | 7A  | P       |                                      |
| A |     | MSGFILENM | 10A | H       |                                      |
| A |     | MSGFILELB | 10A | H       |                                      |
| A |     | MSGNBR    | 4A  | H       |                                      |

MsgId and Msgcon keywords result in rpg instructions.

## Dynamic properties tabsheet

Convert screen

General level Components (2) Dynamic properties (15) Messages (3) DSPF13 FMT1

| Record | Dynamic properties        | Dependency field     | Indicators |
|--------|---------------------------|----------------------|------------|
| FMT1   | Variable MSGIDNUM         | MSGIDNUM             |            |
| FMT1   | Variable MSGFILENM        | MSGFILENM            |            |
| FMT1   | Variable MSGFILELB        | MSGFILELB            |            |
| FMT1   | Variable MSGNBR           | MSGNBR               |            |
| FMT1   | Component _1 msgcon       |                      |            |
| FMT1   | Component MSGFIELD2 msgid |                      |            |
| FMT1   | Component MSGFIELD2 msgid | MSGIDNUM, MSGFILENM  | 25         |
| FMT1   | Component MSGFIELD2 msgid | MSGFILENM            | 26         |
| FMT1   | Component MSGFIELD2 msgid | MSGFILENM, MSGFILELB | 27         |

```

P refreshDspf13Fmt1...
P                                     B
/free
sdSetString(F1: '_1': 'caption': sdRtvMsg('SILVERDEV SILVMSGF ': 'SVD0111'));
select;
  when *IN25;
    sdSetString(F1: ':text': sdRtvMsg('*LIBL      ' + MSGFILENM:MSGIDNUM))
  when *IN26;
    sdSetString(F1: ':text': sdRtvMsg('SILVERDEV ' + MSGFILENM: 'SVD0112'))
  when *IN27;
    sdSetString(F1: ':text': sdRtvMsg(MSGFILELB + MSGFILENM: 'SVD0112'))
  when *IN28;
    sdSetString(F1: ':text': sdRtvMsg(MSGFILELB + MSGFILENM: 'USR0112'))
  when *IN29;
    sdSetString(F1: ':text': sdRtvMsg(MSGFILELB + MSGFILENM: 'USR' + MSGNBR))
  when *IN30;
    sdSetString(F1: ':text': '')
  other;
    sdSetString(F1: ':text': sdRtvMsg('SILVERDEV ' + 'SILVMSGF ': 'SVD0114'))
end;
sdSetBool(F1: 'MSGFIELD2': 'visible': *IN01);
/end-free
P refreshDspf13Fmt1...
P                                     E
P validForm

```