

# SILVERDEV

## GUIDE DE PROGRAMMATION

*P r o g r a m m a t i o n   R P G   I L E*

**Notice relative aux droits d'auteurs.**

Les informations contenues dans ce document pourront faire l'objet de modifications sans préavis et ne sauraient en aucune manière engager **EXPERIA**. La fourniture du progiciel est régie par un octroi de licence ou un accord de confidentialité. Le progiciel ne peut être utilisé, copié ou reproduit sur quelque support que ce soit que conformément aux termes de cette licence ou de cet accord de confidentialité. L'acheteur ne peut effectuer des copies que dans le but de sauvegarde ou d'archivage.

Aucune partie du manuel et du progiciel ne peut être reproduite ou transmise par quelque moyen que ce soit, électronique ou mécanique, y compris par photocopie, enregistrement ou tout autre procédé de stockage, de traitement et de récupération d'informations, pour d'autres buts que l'usage personnel de l'acheteur sans permission expresse et écrite de la société **EXPERIA**.

IBM, AS/400, iSeries, System i, i5, Power I sont des marques déposées de International Business Machines Corporation. Windows est une marque déposée de Microsoft.

Tous les autres produits sont des marques déposées de leur société respective.

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

## Table des matières

<b>Chapitre 1. Menu Serveur .....</b>	<b>5</b>	Fiches ouvertes .....	38
Connexion .....	5	Enregistrer .....	39
Déconnexion .....	6	Enregistrer sous .....	39
Configuration .....	6	Tout enregistrer .....	39
<b>Chapitre 2. Contexte .....</b>	<b>8</b>	Compiler .....	39
Introduction .....	8	Fermer .....	40
Création d'un contexte .....	8	Tout fermer .....	40
Configuration d'un contexte .....	8	Préférences .....	41
Sélection d'un contexte .....	10	Déconnexions .....	41
Mise à jour d'un contexte .....	13	<b>Chapitre 5. Editeur de sources .....</b>	<b>43</b>
<b>Chapitre 3. Création d'une application ....</b>	<b>14</b>	Menu membres .....	43
<b>Chapitre 4. Conception d'une fiche.....</b>	<b>17</b>	Fenêtre associée.....	43
Palette de composants .....	17	Compilation .....	44
Manipulation des composants.....	18	Commande.....	44
Déplacer/Dimensionner un composant.....	18	Contrôle d'intégrité .....	44
Déplacer/Dimensionner un composant au pixel près .....	19	Résultats de compilation, Spools et travaux soumis.....	44
Revenir au composant conteneur.....	19	Complétion de code .....	45
Sélectionner Plusieurs composants avec la souris.....	19	Fonctions .....	45
Inspecteur de propriétés .....	20	Arborescence fenêtres.....	46
Propriétés directement modifiables .....	20	Liste des fonctions et moments .....	47
Boîtes de dialogue.....	20	Signets .....	48
Objets .....	21	Raccourcis .....	49
Ensembles .....	21	Recherche d'une expression dans un source ....	49
Références .....	22	<b>Chapitre 6. Générateur .....</b>	<b>52</b>
Collections .....	22	Source RPG IV .....	52
Evenements .....	23	*/BLOCK .....	52
Composant non visible. ....	24	Gestionnaires d'événements : */EVENT .....	54
Menu Edition .....	24	Protection de QRPGLSRC .....	56
Couper .....	24	PSVDCMP .....	56
Copier .....	24	Source généré.....	57
Coller .....	25	Éléments de base d'un programme .....	57
Défaire .....	25	sdStart .....	58
Ordre de tabulation.....	26	<b>Chapitre 7. Mode RT ou LR .....</b>	<b>60</b>
Arrière plan .....	28	<b>Chapitre 8. Manipulation des composants</b>	
Premier plan .....	28	par programme .....	<b>62</b>
Alignement.....	28	sdSetString, sdGet .....	62
Tailles .....	28	SdSetNum, sdGetNum .....	62
Vue Texte .....	29	sdSetBool, sdgetBool .....	63
Verrouillage des composants .....	30	<b>Chapitre 9. Debug côté serveur .....</b>	<b>64</b>
Thèmes .....	30	Debug côté serveur avec le designer .....	64
Activer Thème.....	31	Lancement du debug .....	64
Appliquer un thème à la sélection .....	31	Choix du programme .....	66
Enregistrer sélection dans un thème .....	32	Choix du module .....	67
Verrouillage des sources .....	32	Menu contextuel .....	68
Aide .....	32	Zone de commandes de debug.....	70
Accès depuis le designer .....	32	Clicks et Doubles clicks .....	71
Sommaire .....	32	Log de debug .....	71
Index.....	33	Liste des fonctions.....	71
Recherche .....	34	Ajouter un module .....	72
Page d'aide.....	35	Choix de la vue.....	73
Aide depuis le réseau .....	36	Fonctionnalités dans une vue.....	73
Menu Fiches .....	37	Terminer le debug .....	73
Nouvelle fiche .....	37	Attributs de debug .....	74
Ouvrir fiche .....	37	Relancer un programme silverdev .....	74
Fiches récentes.....	37	Point d'entrée .....	75
Nouveau à partir d'un modèle .....	38		

Debug en 5250 .....	76	Execution seule .....	116
Point d'entrée .....	78	<b>Chapitre 17. Menu Outils.....</b>	<b>118</b>
Instructions de debug .....	79	Commande .....	118
ATTR .....	79	Job log .....	120
Break .....	79	Liste des travaux .....	120
Sbreak .....	79	Liste des spools .....	120
Clear .....	80	Liste de bibliothèques .....	121
EVAL .....	80	Propriétés .....	122
QUAL .....	81	Assistant .....	122
Step Statement .....	81	Schémas base de données .....	125
Watch .....	81	Liste de composants .....	127
<b>Chapitre 10. Debug côté client .....</b>	<b>82</b>	Liste événements serveur .....	127
Onglet Script .....	83	Liste des événements locaux .....	127
Onglet Trace : .....	84	Arborescence .....	128
Microsoft Script Debugger .....	85	Décompilation .....	130
<b>Chapitre 11. Erreurs.....</b>	<b>89</b>	Explorateur .....	131
Erreurs côté serveur .....	89	Versionning .....	131
Erreurs côté client .....	89	Comparer des sources .....	131
Debugger just in time : .....	91	Comparer des contenus .....	131
<b>Chapitre 12. JOBD programmes en MSGW</b>	<b>92</b>	<b>Chapitre 18. Erreurs fréquentes .....</b>	<b>133</b>
<b>Chapitre 13. Sources écrans .....</b>	<b>93</b>	<b>Chapitre 19. Solutions remplacées.....</b>	<b>135</b>
<b>Chapitre 14. Impressions.....</b>	<b>95</b>	Composant CStringGrid .....	135
Impression simple .....	95	SdSelectColor, sdSelectFile, sdPrintDlg.....	135
Aperçu avant impression .....	95	SdSetSet, sdAddSet, sdDelSet .....	135
Orientation .....	96	sdAddNode .....	135
Imprimer une liste.....	96	<b>Chapitre 20. Technique avancées.....</b>	<b>136</b>
CRepLabel.....	97	Fonctions de la catégorie List.....	136
CRepMemo : .....	97	Optimisation .....	136
CRepChart : .....	97	SFL : Recopie des données en locales.....	136
Formatage d'une occurrence .....	98	Evenements locaux .....	136
Ruptures .....	98	Applications Multifenêtres .....	136
Rupture sans détails.....	102	Applications MDI .....	136
Dessiner un tableau .....	102	Application multifenêtres avec des onglets .....	136
Reports de valeurs: .....	103	MultiOccurrences d'une fenêtre .....	136
Sauts de page.....	103	Multioccurrences , programme de service sdsrvlst.....	136
Formatage au milieu d'une chaîne .....	103	Exemples et astuces.....	136
Impression en pdf.....	104	Web services.....	136
Choix d'imprimante.....	105	Applications multilingues .....	136
Imposer une imprimante .....	106	Fonctions catégorie images .....	136
FromPage ToPage .....	106	Fonction boîtes de dialogues .....	136
Conversions mm/Pixels .....	107	Fonction catégorie fichiers pc.....	136
<b>Chapitre 15. Evenements.....</b>	<b>108</b>	Fonctions catégorie collections .....	136
Gestionnaire d'évènement .....	108	Fonctions catégorie TStrings.....	136
Paramètres d'évènement .....	108	Fonction catégorie ensembles .....	136
Affectation dynamique d'évènements .....	111	Fonctions diverses.....	136
<b>Chapitre 16. Fiches modales .....</b>	<b>114</b>	Liste des fenêtres et des événements .....	137
Introduction .....	114	Couleurs .....	137
ModalResult .....	114	Ancrer des composants .....	137
Exemple .....	114	Opérations de drag and drop. ....	137
		Adaptation taille écran .....	137
		ADO .....	137
		Exit points .....	137
		Interaction avec des programmes externes. ....	137
		Programme de service sdsrvjson .....	137
		WebBrowser.....	137

# Chapitre 1. Menu Serveur

## Connexion

Le menu "Serveur/Connexion" permet d'établir une connexion avec le serveur de Silverdev. Il est possible de travailler sur une fiche sans être connecté, mais pour sauvegarder ou ouvrir une fiche, il faut impérativement être connecté.

Si vous essayez d'effectuer une opération qui nécessite une connexion et que vous n'êtes pas connecté, le designer vous propose la boîte de dialogue pour vous connecter.

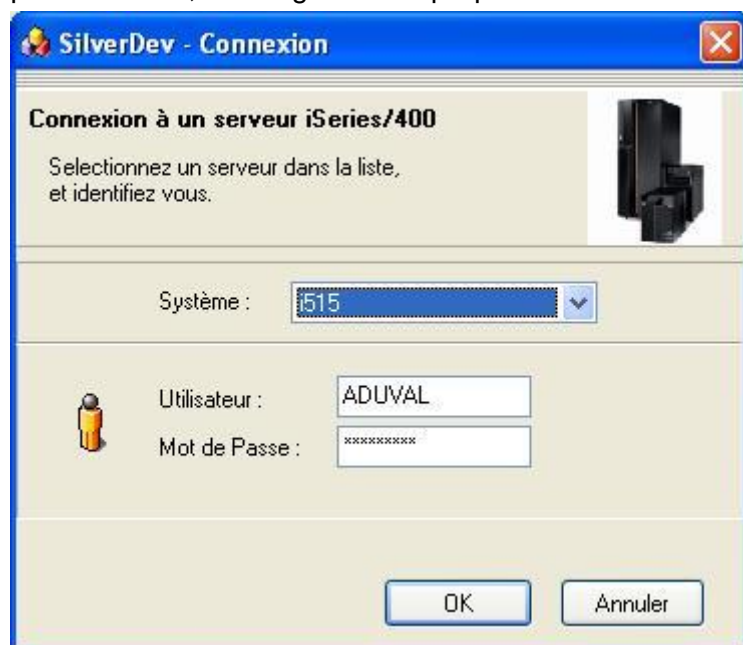


Figure 1

La liste des serveurs affichés dans la liste déroulante provient d'un fichier ini stocké sur le pc du client. Cette liste peut être modifiée en utilisant le menu "Serveur/Configuration".

Lorsque vous êtes connecté, un job vous est attribué sur le serveur. Le numéro de ce job est affiché en bas à gauche du designer.

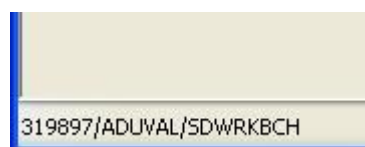


Figure 2

---

## Déconnexion

Le menu "Serveur/Déconnexion" permet de se déconnecter. Si vous fermez le designer, celui-ci effectuera une déconnexion automatiquement.

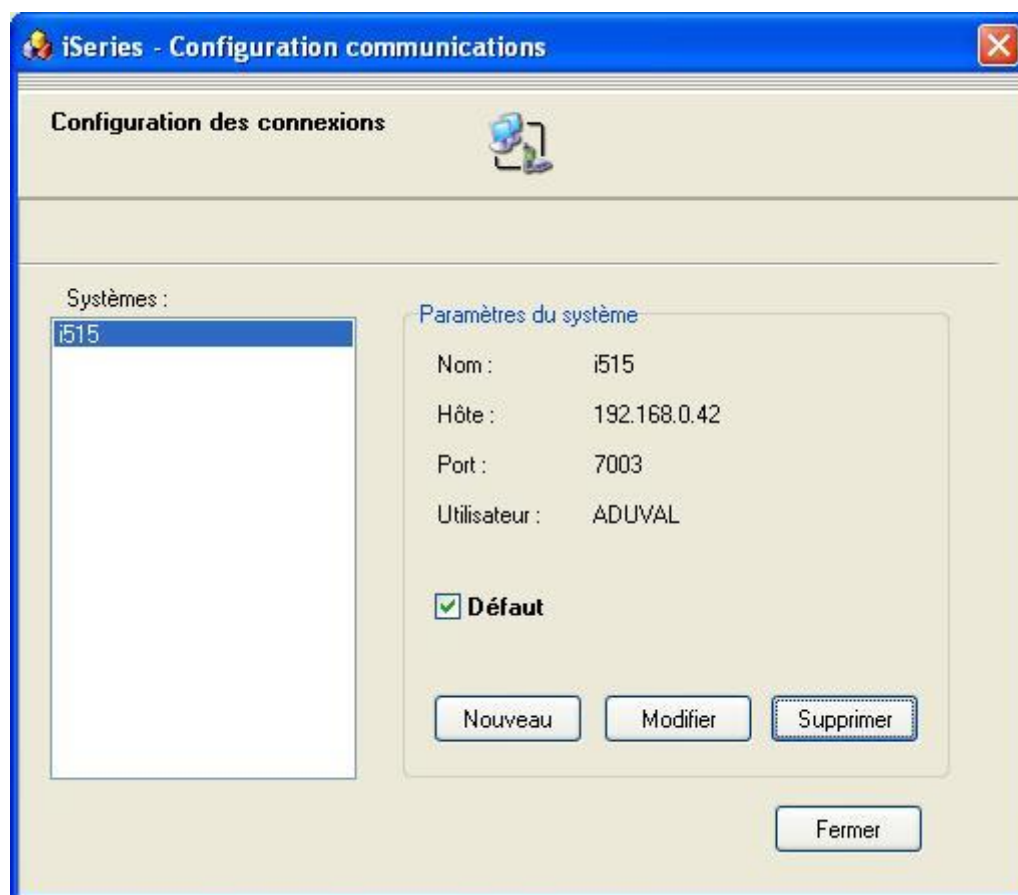
---

## Configuration

Afin de se connecter plus rapidement, vous pouvez mémoriser les données permettant d'accéder au serveur SilverDev. Il est possible de configurer plusieurs serveurs.

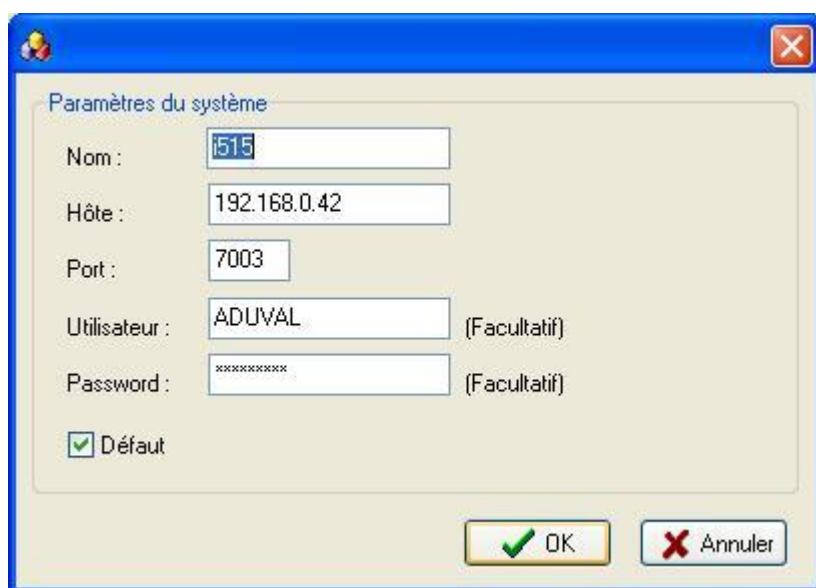
Cette partie configuration est commune à la configuration de MyDesk.exe.

Utilisez le menu serveur/Configuration



**Figure 3**

Pour ajouter un nom de système, utilisez le bouton Nouveau et pour modifier les données liées à un système, utilisez le bouton modifier.

**Figure 4**

Dans la case "Nom", saisissez un nom de votre choix. C'est le nom qui sera proposé dans la boîte de dialogue de connexion

Ensuite saisissez l'adresse IP de l'iSeries/400, puis le numéro de port avec lequel le serveur SILVERDEV a été démarré.

Saisissez le profil sous lequel vous serez connecté.

Il n'est pas obligatoire de saisir le mot de passe à cet endroit.

S'il est saisi, le mot de passe sera stocké (de manière codée) dans le fichier SilverDev.ini.

Cette configuration vous permet de saisir les informations de plusieurs serveurs silverdev mais aussi d'un même serveur avec différents profils.

---

# Chapitre 2. Contexte

---

## Introduction

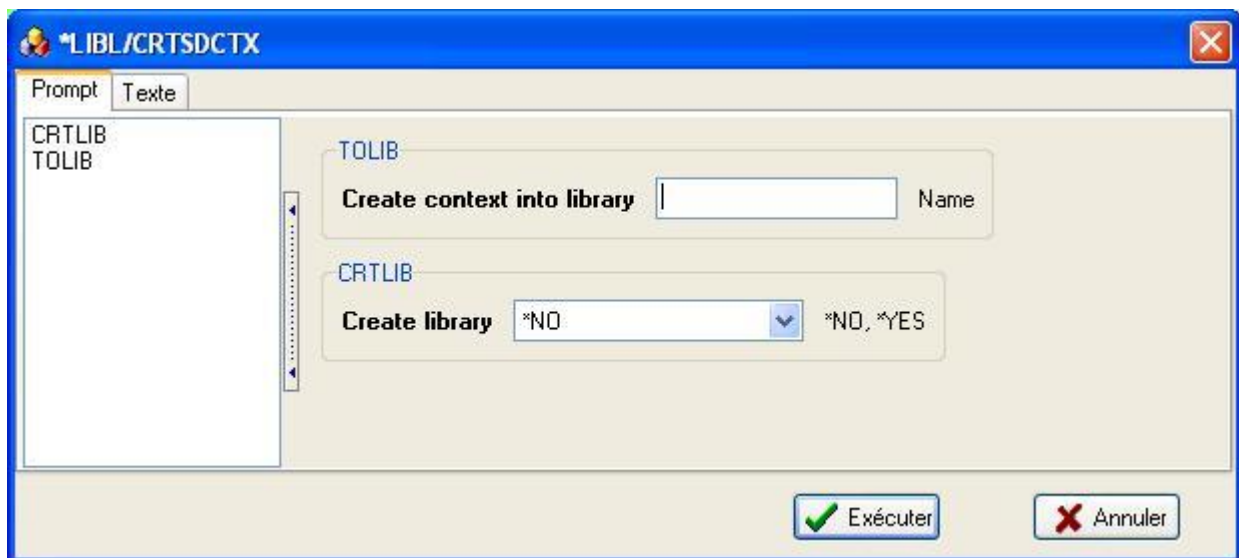
Un contexte est un environnement de travail.

---

## Création d'un contexte

Pour créer un contexte, utilisez la commande SILVERDEV/CRTSDCTX .

Vous pouvez utiliser le menu "Outils/Créer un contexte..." qui utilise cette commande.



Cette commande crée trois fichiers dans la bibliothèque passée en paramètres.

Ces fichiers sont :

PSVDSPA, PSVDSPG et PSVDSPL

Le fichier PSVDSPA stocke les informations générales du contexte.

Le fichier PSVDSPG stocke la liste des programmes du contexte.

Le fichier PSVDSPL stocke la liste de bibliothèques du contexte.

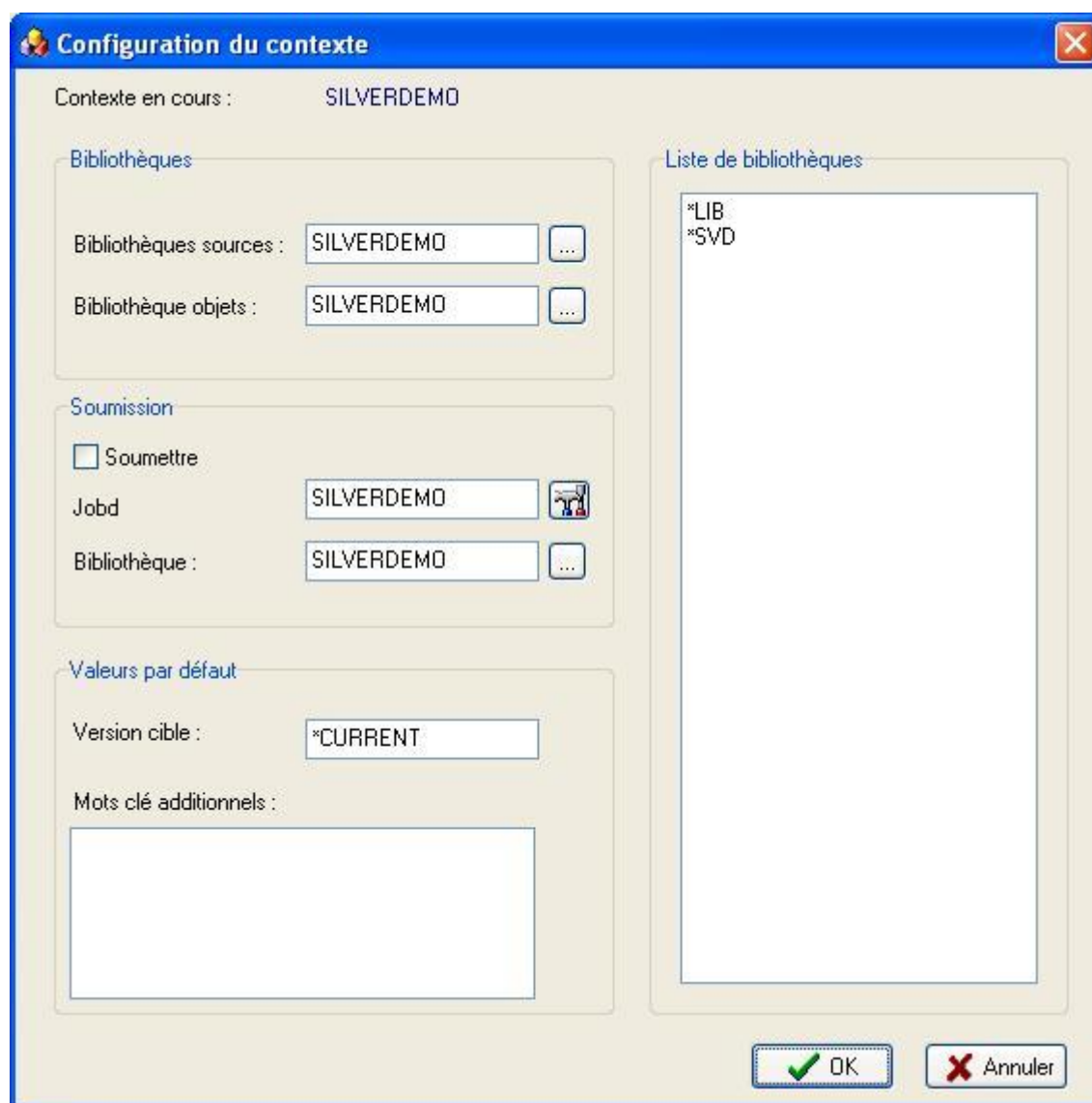
---

## Configuration d'un contexte

Pour configurer un contexte, utilisez le menu "Outils/Configurer un contexte..."

Après avoir choisi un contexte dans la liste, vous obtenez la fenêtre suivante :





**Bibliothèque sources** : Indique la bibliothèque où seront stockés les sources. Lors de la création d'un programme depuis l'option 1 du menu silver, deux sources sont générés. Un dans QXTLESRC pour les moments et un dans QRPGLSRC pour le programme final.

**Bibliothèque objets :** Indique la bibliothèque ou les objets seront compilés.

Soumission JOBD : Indique le nom de la file d'attente de travaux où sont placés les travaux de génération et de compilation.

Valeurs par défaut : Cette section permet de saisir de manière générale les paramètres de compilation. Il est cependant possible de redéfinir pour chaque programme Silverdev, grâce à l'option 8 les paramètres de la compilation.

Liste de bibliothèques : Cette section permet de saisir une liste de bibliothèques qui sera attribuée au job lorsque ce contexte est sélectionné.

Vous pouvez utiliser les valeurs particulières \*LIB et \*SVD.

\*LIB correspond à la bibliothèque où se trouvent les fichiers PSVDSPA,PSVDPG et PSVDSPL.

\*SVD correspond à la bibliothèque où est installée silverdev.

## Sélection d'un contexte

Utilisez le menu " Outils/Contexte... " pour faire apparaître la liste des contextes.

Cette liste est créée en recherchant les fichiers PSVDSPA.



Cliquez sur OK pour sélectionner un contexte.

La liste des programmes du contexte Silverdemo est affichée dans une grille.



Contexte

SILVERDEMO

Nom

Programme	Description	Fenêtre	Bibliothèque fenêtre	Mode affichage	Type	Chemin source écran	Nom éc
SDADDGRP	Group creation	SDADDGRP	*LIBL	*MODAL	SVDRPG		
SDBKLS		SDBKLS	*LIBL	*NORMAL	SVDRPG	*DEFAULT	
SDCHGCFG		SDCHGCFG	*LIBL	*NOMODAL	SVDRPG		
SDCHGCFG0		SDCHGCFG0	*LIBL	*NOMODAL	SVDRPG		
SDDEMOCLI		SDDEMOCLI	*LIBL	*NORMAL	SVDRPG		
SDDEMOEMP		SDDEMOEMP	*LIBL	*NOSHOW	SVDRPG		
SDDEMOEXP		SDDEMOEXP	*LIBL	*NORMAL	SVDSQLRPG		
SDDEMO1		SDDEMO1	*LIBL	*NORMAL	SVDRPG		
SDDMABBY		*PGM	*LIBL	*NORMAL	SVDRPG	*DEFAULT	
SDDMACT	AS/400 activity	SDDMACT1	*LIBL	*NORMAL	SVDRPG		
SDDMAUT	Authors	SDDMAUT	*LIBL	*NOMODAL	SVDRPG		
SDDMBARS	Bars	SDDMBARS	*LIBL	*NORMAL	SVDRPG		
SDDMBKSA	Books	*PGM	*LIBL	*NORMAL	SVDSQLRPG	*DEFAULT	
SDDMBKS1	Book management main screen	*PGM	*LIBL	*NOMODAL	SVDSQLRPG		
SDDMBKS10	Enlarged image	SDDMBKS10	*LIBL	*MODAL	SVDRPG		
SDDMBKS11	Form printing	*PGM	*LIBL	*NOSHOW	SVDRPG		

Figure 5

Pour modifier les propriétés d'un programme, utilisez le bouton  et pour ajouter un programme à la liste, utilisez le bouton .

En modification, vous obtenez la fenêtre suivante :

**Programme Silverdev**

**Général**

Nom programme :

Type :

Description :

Créé par :  le :

Type de fin  
☒ RT  
☐ LR

**Fenêtre principale**

Bibliothèque :  ...

Fenêtre :  ...

source :

Mode d'affichage  
☒ Normal  
☐ Modal  
☐ Ne pas afficher

**Compilation**

Version cible :

Mots clés additionnels :

Valeur configuration :

☐ Ignorer les mots clés config

OK Annuler

Nom du programme : Nom du programme généré.

Type : Type de source

Description : Description rattachée à l'objet (\*PGM) généré.

Fenêtre principale : Nom de l'objet écran utilisé. (objet du type \*USRSPC)

Mode Affiche fenêtre : 3 valeurs possibles.

\*NOMODAL par défaut. La fenêtre est affichée normalement.

\*MODAL : La fenêtre est affichée en modal.

\*NOSHOW : La fenêtre n'est pas affichée, utile pour les maquettes d'impression.

Type de fin : LR ou RT.

La valeur par défaut est RT.

Le code principal d'un programme silverdev est :

C	callp	sdStart(%paddr('INIT'))
C	eval	*inrt = *on

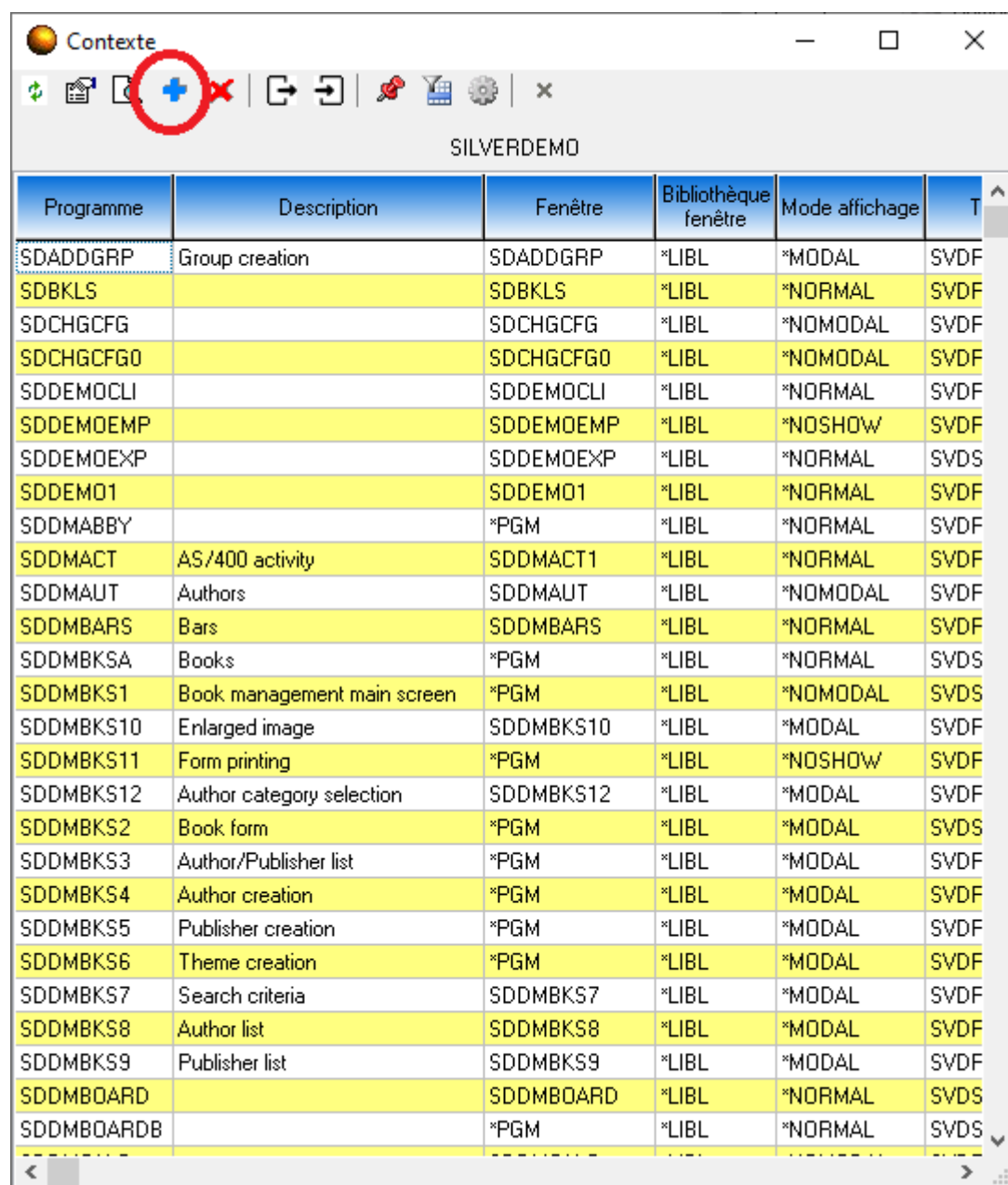
---

## Mise à jour d'un contexte

Voir document Administration, chapitre compatibilité entre les versions

## Chapitre 3. Création d'une application

A partir de la fenêtre contexte, cliquez sur le bouton suivant :



The screenshot shows the 'Programme Silverdev' dialog box with the 'Général' tab selected. The dialog has three tabs: 'Général', 'Compilation', and 'Commandes'. The 'Général' section contains the following fields and options:

- Nom programme :** A text box containing 'TEST'.
- Type :** A dropdown menu showing 'SVDRPG'.
- Description :** An empty text box.
- Créé par :** A text box containing 'ADUVAL'.
- le :** An empty text box.
- Type de fin :** Radio buttons for 'RT' (selected) and 'LR'.
- Fenêtre principale :** A section containing:
  - Bibliothèque :** A text box containing '\*LIBL' and a browse button (...).
  - Fenêtre :** A text box containing '\*PGM' and a browse button (...).
  - Source :** A text box containing '\*DEFAULT' and a browse button (...).
- Mode d'affichage :** Radio buttons for 'Normal' (selected), 'Modal', and 'Ne pas afficher'.

At the bottom right, there are two buttons: 'OK' (with a green checkmark icon) and 'Annuler' (with a red X icon).

Le programme créé s'ajoute à la liste, double cliquez dessus pour ouvrir le source rpg et la fenêtre associée.

Programme	Description	Fenêtre	Bibliothèque fenêtre	Mode affichage	Type	Chemi
SDDMXFER		SDDMXFER	*LIBL	*NORMAL	SVDRPG	
SDDMXFER2		SDDMXFER2	*LIBL	*NOMODAL	SVDRPG	
SDDM5250	5250 pgm call example	SDDM5250	*LIBL	*NOMODAL	SVDRPG	
SDDYNMNU	Dynamic menu creation	SDDYNMNU	*LIBL	*NOMODAL	SVDRPG	
SDFILEREF	Programmes utilisant un fichier	*PGM	*LIBL	*NORMAL	SVDSQLRPG	*DEFA
SDLIBLIST		*PGM	*LIBL	*NORMAL	SVDRPG	*DEFA
SDLSTMSGF	Display message file	*PGM	*LIBL	*NORMAL	SVDRPG	*DEFA
SDTSTCMD		SDTSTCMD	*LIBL	*NOMODAL	SVDRPG	
SDUPDGRP	Group modification	SDADDGRP	*LIBL	*MODAL	SVDRPG	
SDWRKADM	Administrator management	SDWRKADM	*LIBL	*NOMODAL	SVDRPG	
SDWRKGRP	Group management	SDWRKGRP	*LIBL	*NOMODAL	SVDRPG	
SDWRKMBR	Group member management	SDWRKMBR	*LIBL	*MODAL	SVDRPG	
SDZOOMIMG		SDZOOMIMG	*LIBL	*NORMAL	SVDRPG	
TEST		*PGM	*LIBL	*NORMAL	SVDRPG	*DEFA

La fenêtre et le source rpg s'ouvrent :

The screenshot shows the SilverDev IDE interface. On the left, the source code of the TEST program is displayed in a green editor. The code includes comments and RPG instructions for various blocks. On the right, the 'Contexte' window is open, showing a table of programs and their properties. The table has columns for Programme, Description, Fenêtre, Bibliothèque, Mode affichage, Type, and Chemi. The 'TEST' program is highlighted in orange.

```

1 // *****
2 // * Title . . . :
3 // * Author . . .ADUVAL
4 // * Date . . .2020-08-28
5 // * Project . . :
6 // * Description :
7 // *
8 // *
9 // * -Technical data-----
10 // *
11 // * Main SDF . *LIBL/*PGM
12 // *
13 // *****
14
15 //BLOCK RPGSPCIH
16 // ----- RPGSPCIH : H specifications (Heading)
17
18 //BLOCK RPGSPCIF
19 // ----- RPGSPCIF : Files declarations (F Spec.)
20
21 //BLOCK RPGSPCID
22 // ----- RPGSPCID : Data descriptions (D Spec.)
23
24 //BLOCK RPGSPCII
25 // ----- RPGSPCII : Input descriptions (I Spec.)
26
27 //BLOCK RPGPARM
28 // ----- RPGPARM : Program parameters and declarations (KLIST)
29
30 //BLOCK RPGPGMSTART
31 // ----- RPGPGMSTART : Beginning of Initializations procedure
32
33 //BLOCK RPGBEFORECREATE
34 // ----- RPGBEFORECREATE : just before form creation
35
36 //BLOCK RPGAFTERCREATE
37 // ----- RPGAFTERCREATE : After form creation
38
39 //BLOCK RPGBEFORESHOW
40 // ----- RPGBEFORESHOW: Before show(F1)
41

```

Programme	Description	Fenêtre	Bibliothèque fenêtre	Mode
SDDMPRTT	Print Type	SDDMPRTT	*LIBL	*NOR
SDDMRCHEDT	Richedit example	SDDMRCHEDT	*LIBL	*NOMI
SDDMRICH		SDDMRICH	*LIBL	*NOMI
SDDMRAP	Relative positions	*PGM	*LIBL	*NOR
SDDMSAVE	Save File Transfer Tool	SDDMSAVE	*LIBL	*NOR
SDDMSAVF2	Select Library	SDDMSAVF2	*LIBL	*MOD
SDDMSAVF3		SDDMSAVF3	*LIBL	*MOD
SDDMSKINS		SDDMSKINS	*LIBL	*MOD
SDDMSPLA1		SDDMSPLA1	*LIBL	*NOR
SDDMSPLA2		SDDMSPLA2	*LIBL	*MOD
SDDMSPLF	spooled file list	SDDMSPLF	*LIBL	*NOMI
SDDMSPLF2	display one spooled file	SDDMSPLF2	*LIBL	*NOS
SDDMSPLF1	paramètres de sddmsplf1	*PGM	*LIBL	*MOD
SDDMSQL	dynamic sql	SDDMSQL	*LIBL	*NOMI
SDDMSRC	PG source visualisation	SDDMSRC	*LIBL	*NOMI
SDDMSTYLES		SDDMSTYLES	*LIBL	*MOD
SDDMTAB1	findows in tabsheets example	SDDMTAB1	*LIBL	*NOMI
SDDMTAB2	findows in tabsheets example	SDDMTAB2	*LIBL	*NOS
SDDMTAB3	findows in tabsheets example	SDDMTAB3	*LIBL	*NOS
SDDMTAB4	findows in tabsheets example	SDDMTAB4	*LIBL	*NOMI
SDDMTAK	uzzle	SDDMTAK	*LIBL	*NOMI
SDDMTHM	theme management	SDDMTHM	*LIBL	*NOMI
SDDMTSLMD	translate Message	SDDMTSLMD	*LIBL	*MOD
SDDMTSLMF	translate MsgF tool	SDDMTSLMF	*LIBL	*NOR
SDDMTSLMS	translate Message - Subst var	SDDMTSLMS	*LIBL	*MOD
SDDMTWAIN	CT wain component	SDDMTWAIN	*LIBL	*NOMI
SDDMUPDEV		SDDMUPDEV	*LIBL	*MOD
SDDMUPLOAD		SDDMUPLOAD	*LIBL	*NOMI
SDDMVIEW	Agenda	SDDMVIEW	*LIBL	*NOMI
SDDMXFER		SDDMXFER	*LIBL	*NOR
SDDMXFER2		SDDMXFER2	*LIBL	*NOMI



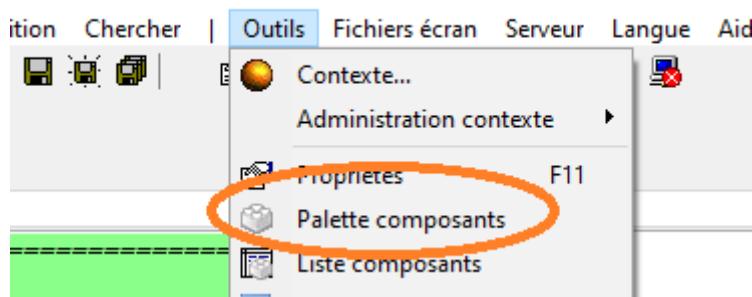
---

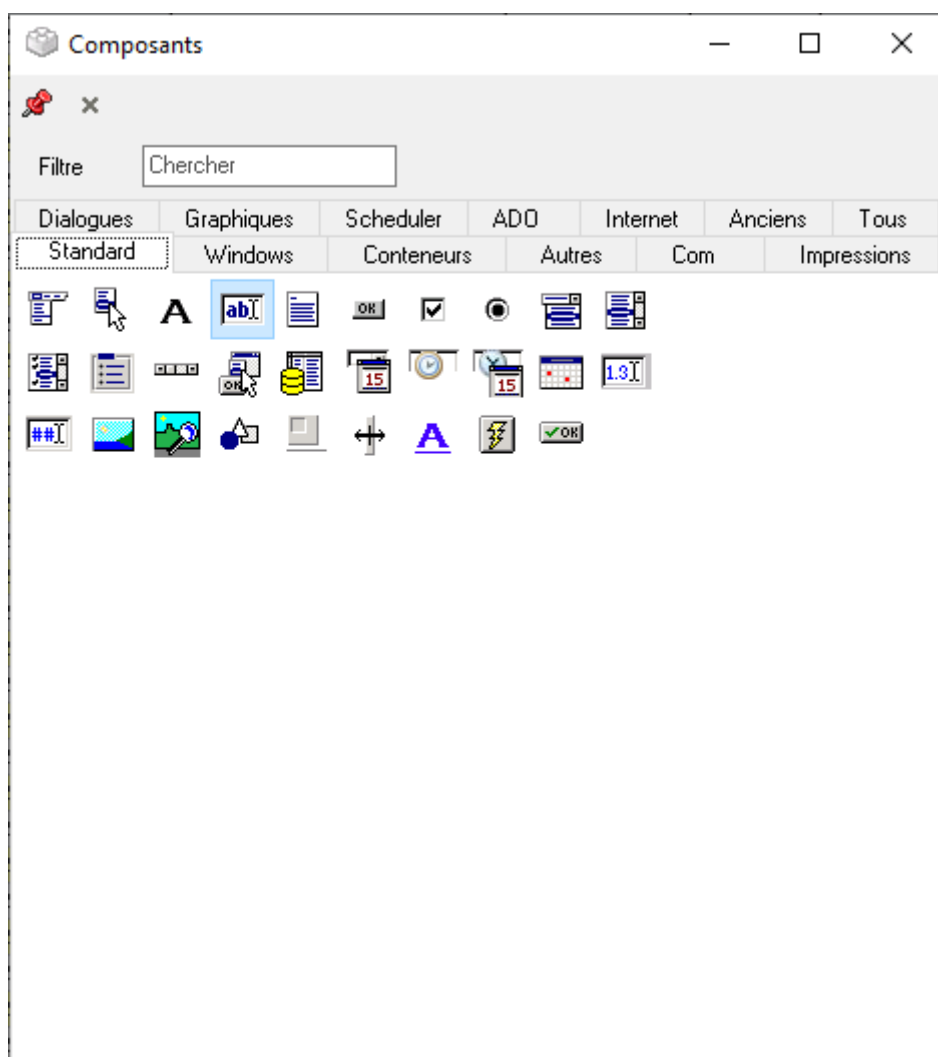
## Chapitre 4. Conception d'une fiche

---

### Palette de composants

Si la palette de composants n'est pas affichée, utilisez le menu outils/palette de composants :





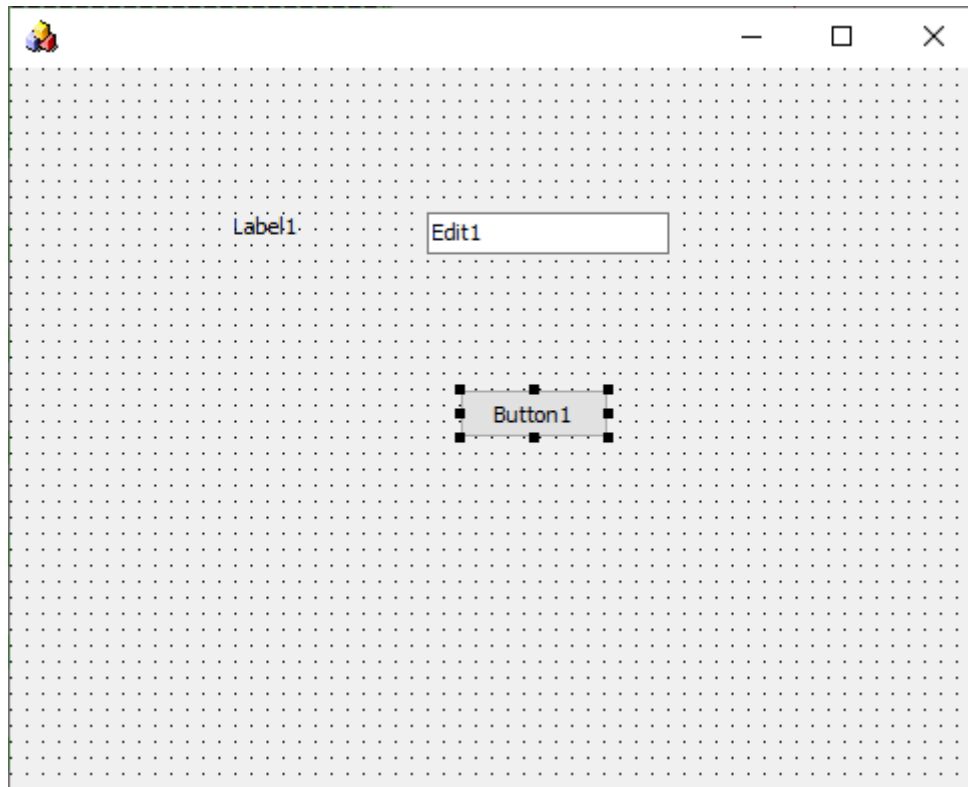
Sélectionnez un type de composant dans la palette et cliquez sur la fiche pour créer ce composant.

---

## Manipulation des composants

### Déplacer/Dimensionner un composant

Cliquez sur un composant pour le sélectionner.



Cliquez sur le composant sélectionné, maintenez le bouton gauche enfoncé, et déplacez le composant.

Pour redimensionner le composant, cliquez sur l'un des carrés noir qui entourent le composant.

### Déplacer/Dimensionner un composant au pixel près

Pour déplacer un composant, sélectionnez le, maintenez la touche contrôle enfoncée, puis utilisez les touche directionnelles pour le déplacer. Il est possible de faire cette manipulation lorsque plusieurs composants sont sélectionnés.

Pour dimensionner un composant, sélectionnez le, maintenez la touche shift enfoncée, puis utilisez les touche directionnelles pour le modifier. Il est possible de faire cette manipulation lorsque plusieurs composants sont sélectionnés.

### Revenir au composant conteneur

Lorsqu'un composant occupe tout l'espace du composant dans lequel il est contenu, il n'est plus possible de cliquer sur le conteneur pour le sélectionner. Il suffit alors d'utiliser la touche "echap" pour passer au composant conteneur.

### Sélectionner Plusieurs composants avec la souris

Pour sélectionner plusieurs composants en même temps, maintenez la touche contrôle enfoncée et en même temps, cliquez avec le bouton gauche de la souris, puis faites bouger la souris pour dessiner un carré sur la fiche. Relâchez, tous les composants situés dans le carré sont sélectionnés.

Une seconde solution est de maintenir la touche shift enfoncée, et de cliquer sur les composants désirés un à un.

Il est alors possible de définir les propriétés que ces composants ont en commun en même temps.

---

## Inspecteur de propriétés

L'inspecteur de propriétés permet de modifier les propriétés des composants qui sont sélectionnés sur la fiche en cours.

### Propriétés directement modifiables

Certaines propriétés sont directement modifiables dans l'inspecteur.

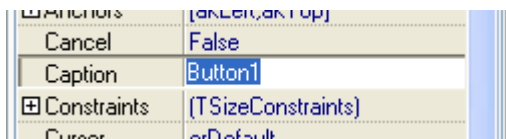


Figure 6

### Boîtes de dialogue

D'autres, nécessitent l'utilisation d'une boîte de dialogue.

Pour faire apparaître cette boîte de dialogue, cliquez sur le bouton avec trois points dans l'inspecteur. Ces propriétés apparaissent en bleu.



Figure 7



Figure 8

## Objets

Certaines propriétés peuvent être développées. Il s'agit en général de propriétés de type objet ou ensemble.

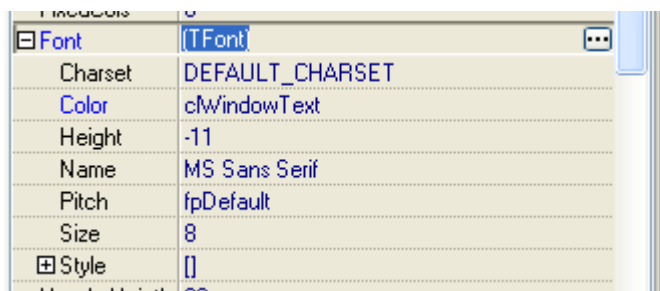


Figure 9

## Ensembles

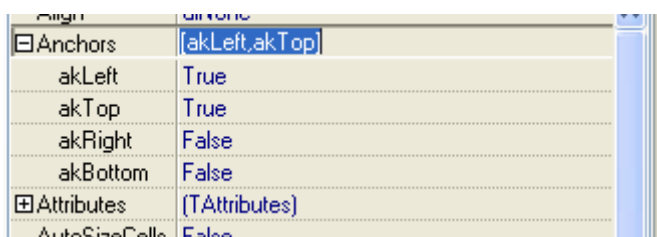


Figure 10

## Références

Les propriétés qui sont en rouge sont des propriétés qui font référence à un autre composant dans la fiche. Une liste déroulante apparaît et permet de choisir le composant.

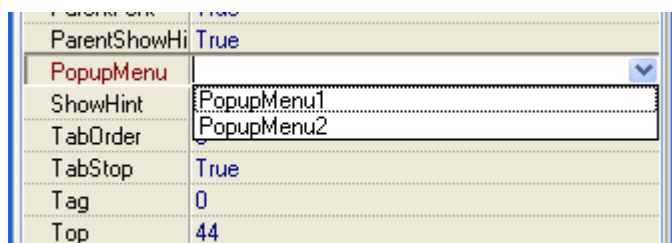


Figure 11

## Collections

Certaines propriétés sont des collections. C'est à dire une liste d'objets.

Par exemple, le composant CSFL possède une collection qui correspond aux colonnes du composant.

Pour modifier une collection, cliquez sur le bouton en face du nom de la propriété

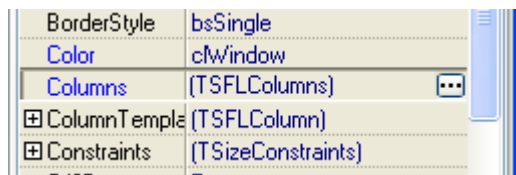


Figure 12

Les éléments de la collection sont alors affichés.

Sélectionnez un des éléments de la collection, et modifiez les propriétés de cet élément dans l'inspecteur.

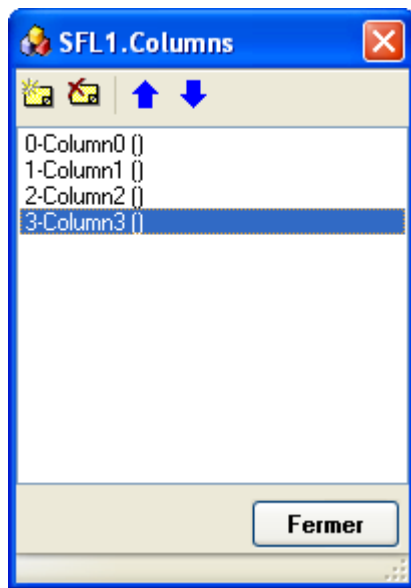


Figure 13

## Evenements

L'inspecteur de propriétés permet aussi de sélectionner les évènements qui devront être signalés au programme RPG.

Utilisez le second onglet et cochez les évènements qui devront être signalés.

Dans le programme, vous utiliserez la fonction `sdSetCallBack` pour relier ces évènements à des procédures RPG. Si vous utilisez le générateur, les appels aux fonctions `sdSetCallBack` seront automatiquement générés.

```
c          callp      sdSetCallBack (F1
c                               : 'Button1.OnClick'
c                               : %paddr (
c                               'BUTTON1_ONCLICK' ) )
```

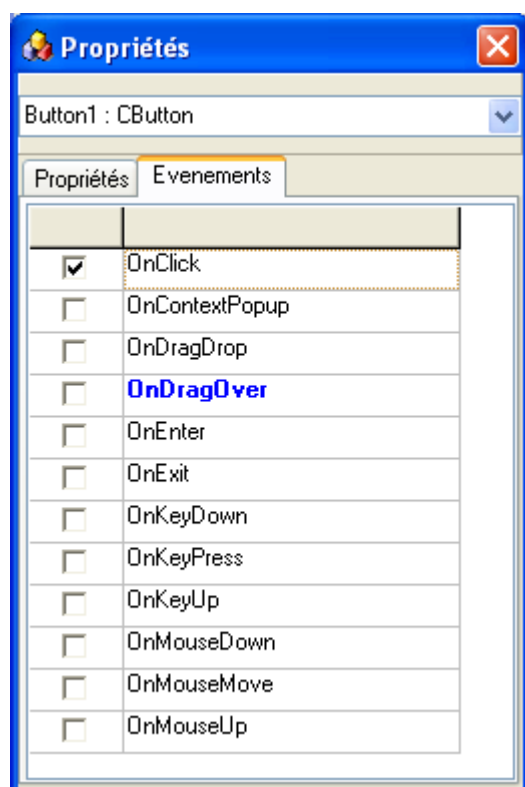


Figure 14

En face des coches, il est possible de saisir du code qui s'exécutera en local, nous verrons cette fonctionnalité dans le **Chapitre 20**

### Composant non visible.

Si un composant n'est pas visible dans le designer. Soit parce que sa propriété left est trop grand soit parce qu'il est sous un autre composant, vous pouvez utiliser l'arborescence ou l'inspecteur de propriété pour le sélectionner.

## Menu Edition

### Couper

Le menu "Edition/Couper" permet de supprimer les composants en cours sur la fiche active et les placer dans le presse-papiers

Les composants en cours sont repérés par des carrés noirs tout autour du composant.

### Copier

Le menu "Edition/Copier" permet de copier les composants en cours dans le presse-papiers

Les composants en cours sont repérés par des carrés noirs tout autour du composant.

Seules les propriétés des composants sont copiées, pas les évènements.



## Coller

Le menu "Edition/Coller permet de coller les composants copiés dans le presse-papiers sur la fiche en cours. Si au moment du collage, un composant pouvant contenir d'autres composants est sélectionné (CPanel, CGroupBox..), les composants seront copiés dans celui-ci.

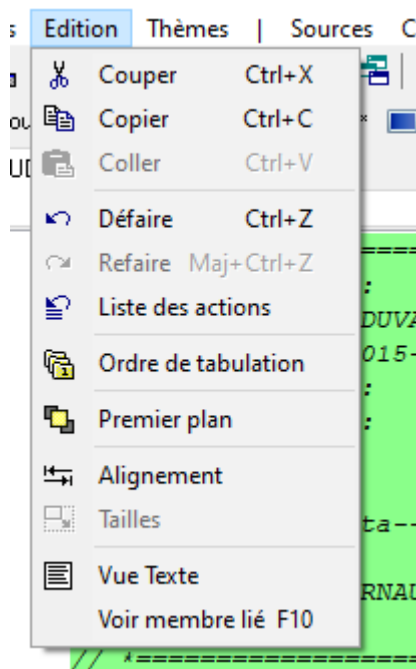
Tous les composants dans le presse papier dont le nom est le même qu'un composant existant seront automatiquement renommés.

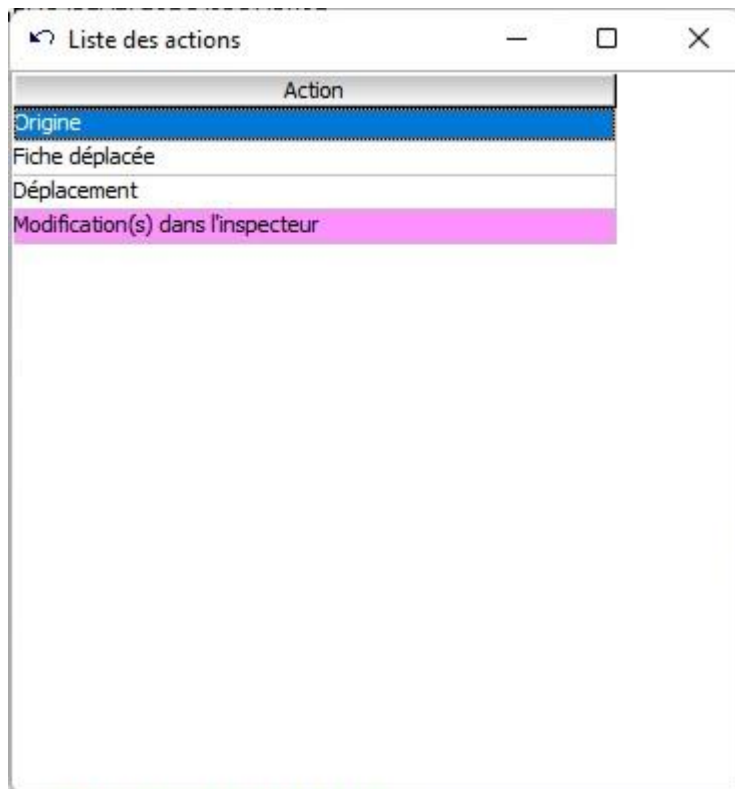
## Défaire

Utilisez le menu défaire pour revenir en arrière, le raccourci est ctrl+z

Remarque : Il est aussi possible de revenir en avant avec le menu refaire , raccourci Ctrl+maj+z

Il est possible d'afficher la liste des actions que l'on peut défaire :





Double cliquez sur une ligne pour ramener la fiche dans l'état après cette action.

Remarque, lorsque vous revenez à un état  $n$ , les états  $n+1$  etc sont encore disponible, mais à la prochaine modification de la fiche, l'état  $n+1$  est remplacé.

### Ordre de tabulation

Le menu "Edition/Ordre de tabulation" permet d'accéder à la boîte de dialogue suivante:



**Figure 15**

Cette boîte de dialogue permet de modifier l'ordre dans lequel les contrôles ont le focus lorsque l'utilisateur utilise la touche tabulation.

Ce menu est désactivé s'il y a plusieurs composants sélectionnés ou si le composant sélectionné ne peut pas contenir d'autres contrôles.

Cet éditeur modifie les propriétés TabOrder des composants.

Il est possible de faire en sorte que le focus ne passe pas par un composant en mettant la propriété TabStop de ce composant à false.

Dans ce cas, l'élément est grisé dans la liste.

Si un composant en contient d'autres, par exemple un CPanel. Lorsque le focus arrive sur ce composant, l'ordre de tabulation à l'intérieur de ce composant est utilisé, puis une fois que le dernier composant dans le CPanel est atteint, le focus réutilise l'ordre de tabulation du composant contenant le CPanel.

Le premier composant ayant le focus est donc le composant ayant la valeur TabOrder à 0. Mais lorsque qu'une fiche est cachée puis visible à nouveau, le focus ne repasse pas au composant ayant le taborder à 0. Le focus reste sur le composant qui avait le focus avant que la fenêtre soit cachée.

Pour remettre le focus sur un composant par programmation, utilisez la fonction `sdSetFocus` ou la propriété `ActiveControl`.

Pour modifier la propriété `activeControl` à l'exécution, utilisez la fonction `sdSetCtrl`.

Attention : Il n'est pas possible d'utiliser la fonction `sdSetFocus` sur un composant si la fenêtre de ce composant n'est pas visible. Appelez la fonction `sdSetFocus` après un

sdShow. Si vous utilisez la fonction sdShowModal, appelez la fonction sdSetFocus dans l'événement OnShow de la fenêtre ou utilisez la propriété ActiveControl.

### Arrière plan

Le menu "Edition/Arrière plan" place un contrôle fenêtré derrière tous les autres contrôles fenêtrés de son parent ou place un contrôle non fenêtré derrière tous les autres contrôles non fenêtrés de son parent.

### Premier plan

Le menu "Edition/Premier Plan" place le contrôle en avant de tous les autres contrôles du contrôle parent.

### Alignement

Le menu "Edition/Alignement" permet d'accéder à la boîte de dialogue suivante :

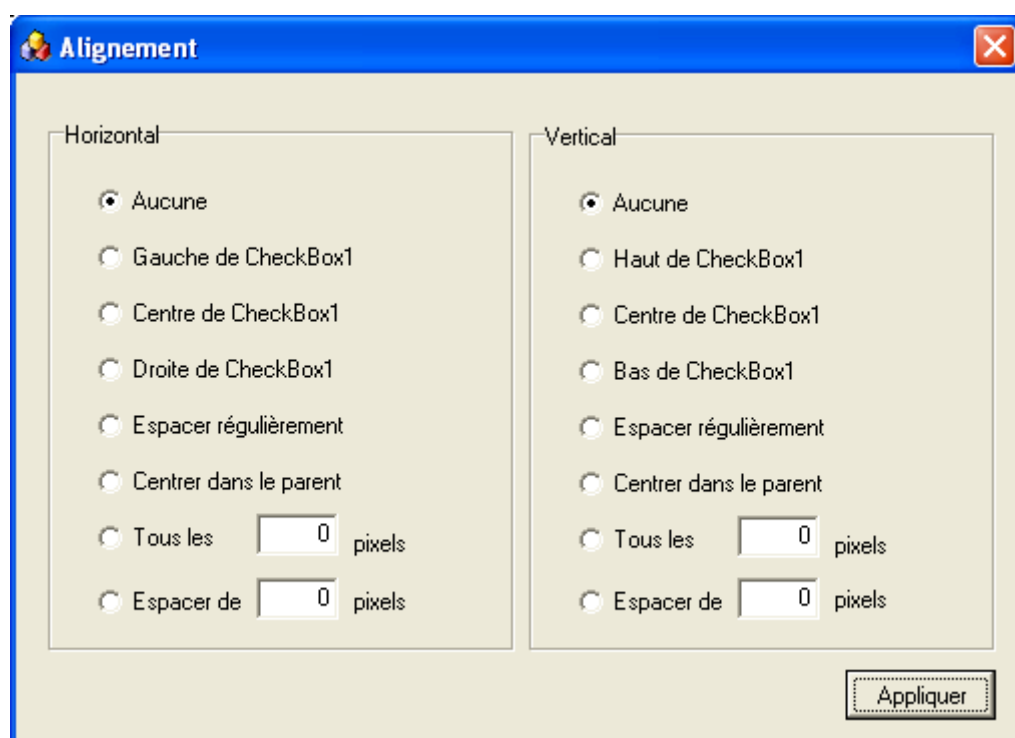


Figure 16

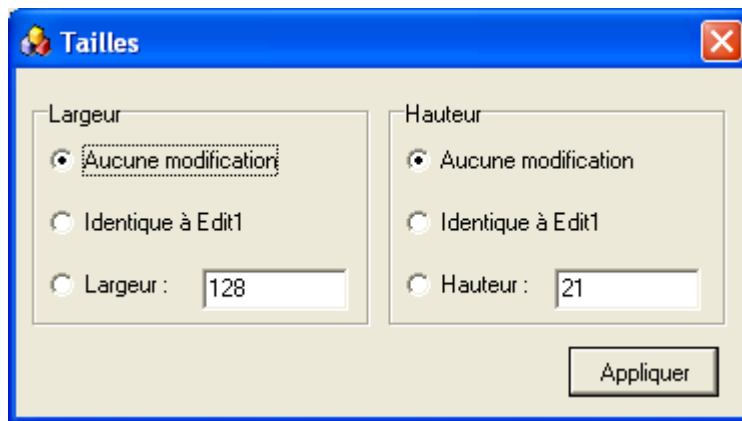
Cette boîte de dialogue permet de repositionner automatiquement les contrôles sélectionnés.

Elle n'est proposée que s'il y a plusieurs composants sélectionnés et qu'ils sont tous contenus dans le même parent.

Les modifications se font par rapport à l'un des composants. Lorsque plusieurs composants sont sélectionnés, l'un d'entre eux est entouré de noir et les autres de gris. C'est celui entouré en noir qui sert de référence.

### Tailles

Le menu "Edition/Tailles" permet d'accéder à la boîte de dialogue suivante :

**Figure 17**

Cette boîte de dialogue permet de redimensionner automatiquement les contrôles sélectionnés.

Cette boîte de dialogue n'est proposée que s'il y a plusieurs composants sélectionnés et qu'ils sont tous contenus dans le même parent.

Les modifications se font par rapport à l'un des composants. Lorsque plusieurs composants sont sélectionnés, l'un d'entre eux est entouré de noir et les autres de gris. C'est celui entouré en noir qui sert de référence.

## Vue Texte

La vue texte permet de voir la fiche sous la forme d'un texte. Cela peut être pratique pour modifier plusieurs propriétés.

Si vous modifiez le texte, la fiche sera modifiée à condition que vous ne fassiez pas d'erreur dans la syntaxe.

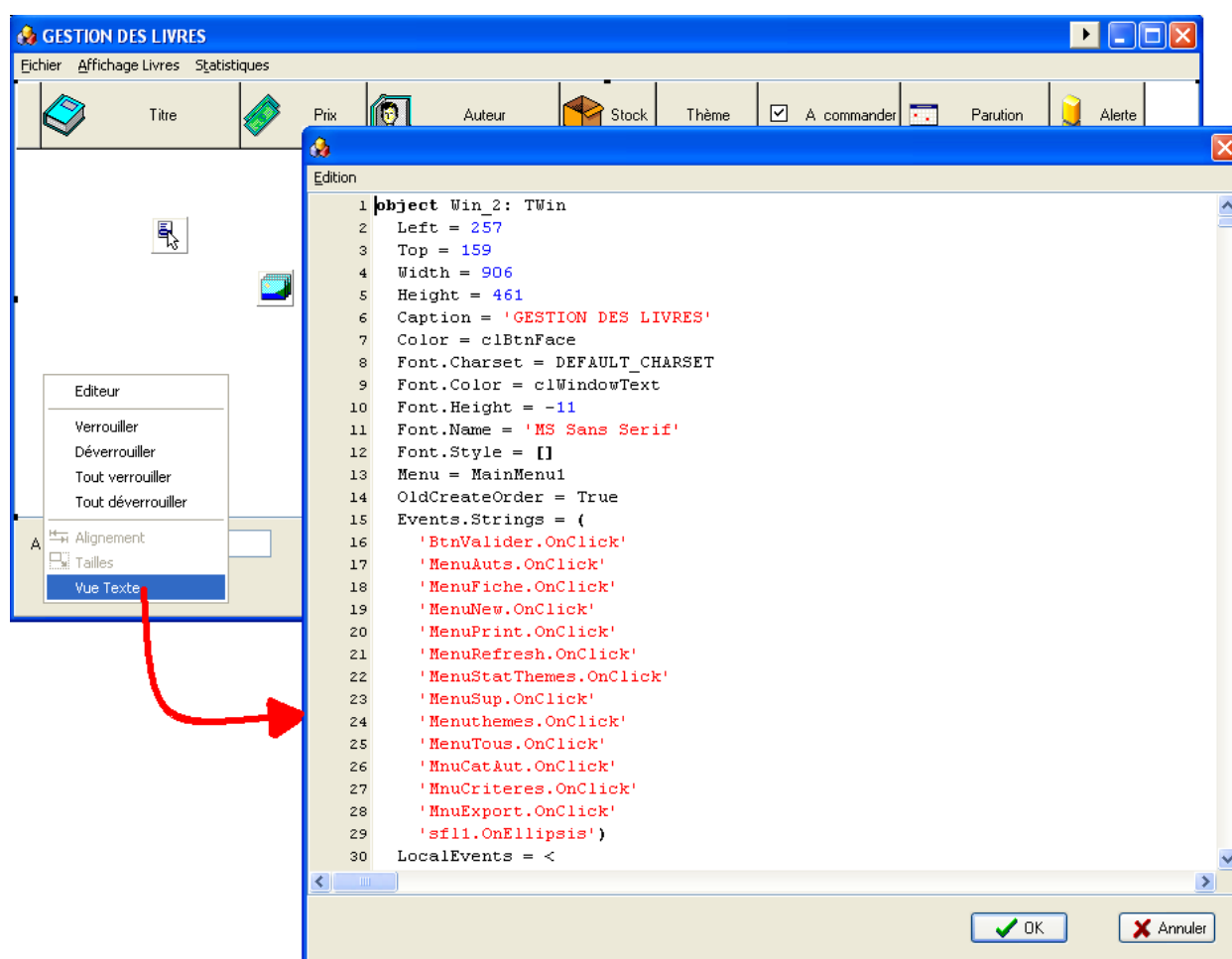


Figure 18

## Verrouillage des composants

Le click droit permet d'accéder au menu Verrouiller/Déverrouiller.

Le fait de verrouiller les composants empêche de pouvoir déplacer ceux ci avec la souris. Il est toujours possible de modifier les composants via l'inspecteur de propriété. Cela évite de déplacer accidentellement un composant après avoir passé du temps à aligner tous les composants. Les composants verrouillés sont encadrés en rouge au lieu d'être en noir.

## Thèmes

Lorsque vous déposez des composants sur une fiche, vous êtes souvent amené à modifier certaines propriétés de ce composant en utilisant l'inspecteur de propriétés.

Vous êtes alors souvent amené à affecter les même valeurs à un certain nombre de propriétés. Les thèmes vous évite cette tâche répétitive.

Un thème permet d'enregistrer pour chaque composant les valeurs de ses propriétés, et de les ré-appliquer à d'autres composants.

Il est possible d'enregistrer différents thèmes.

### Activer Thème

Pour activer un thème, utilisez le menu Thème/Activer thème.

Une boîte de dialogue affiche la liste des thèmes existants sur votre disque.

Sélectionnez un des thèmes pour l'activer.

Pour sélectionner un thème, double cliquez sur le nom d'un des thèmes.

Le nom du thème en cours apparaît dans la barre d'état du designer.

Tous les composants que vous poserez se verront appliquer le thème en cours.

*Remarque 1 : Chaque thème est en fait un sous répertoire du répertoire C:/program files/experia/designer/thèmes*

*Remarque 2 : Vous pouvez supprimer ou créer un thème à partir de cette boîte de dialogue.*

*Remarque 3 : Vous pouvez utiliser les thèmes sans qu'aucun thème ne soit activé.*

*(Voir les paragraphes suivants)*

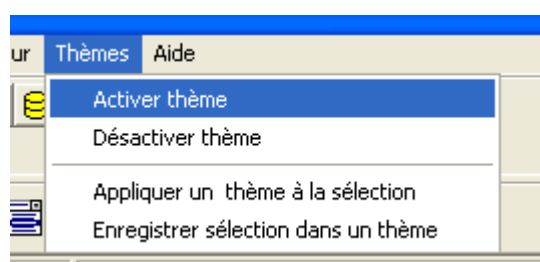


Figure 19

### Appliquer un thème à la sélection

Appliquez un autre thème que celui en cours en utilisant le menu « Thèmes/Appliquer un thème à la sélection ».

Une boîte de dialogue apparaît et vous permet de choisir le thème à appliquer aux composants sélectionnés.

Ceci ne change pas le thème en cours.

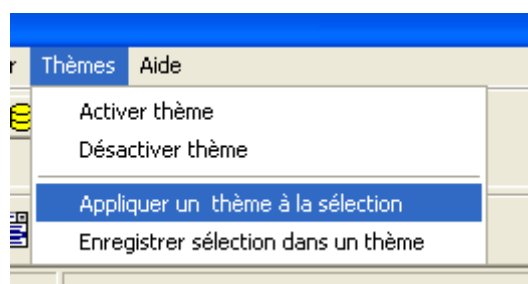


Figure 20

### Enregistrer sélection dans un thème

Enregistrez les propriétés des composants sélectionnés dans un thème.

Une boîte de dialogue apparaît et vous permet de choisir le thème dans lequel s'enregistrent les propriétés. Ceci ne change pas le thème en cours.

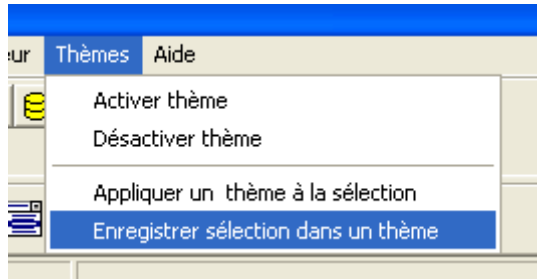


Figure 21

---

## Verrouillage des sources

Lorsque vous ouvrez une fiche dans le designer, le source est verrouillé. Il n'est donc pas possible que deux développeurs ouvrent le même source écran en même temps.

---

## Aide

### Accès depuis le designer

L'aide sur les composants silverdev est accessible depuis le designer via le menu « Aide ». Il est aussi possible d'utiliser la touche F1. Si vous êtes sur une fiche, l'aide concernant le composant sélectionné apparaîtra. Si vous êtes dans l'inspecteur de propriété, l'aide sur la propriété sélectionnée apparaîtra.

### Sommaire

Le sommaire présente les rubriques de l'aide classées par catégorie



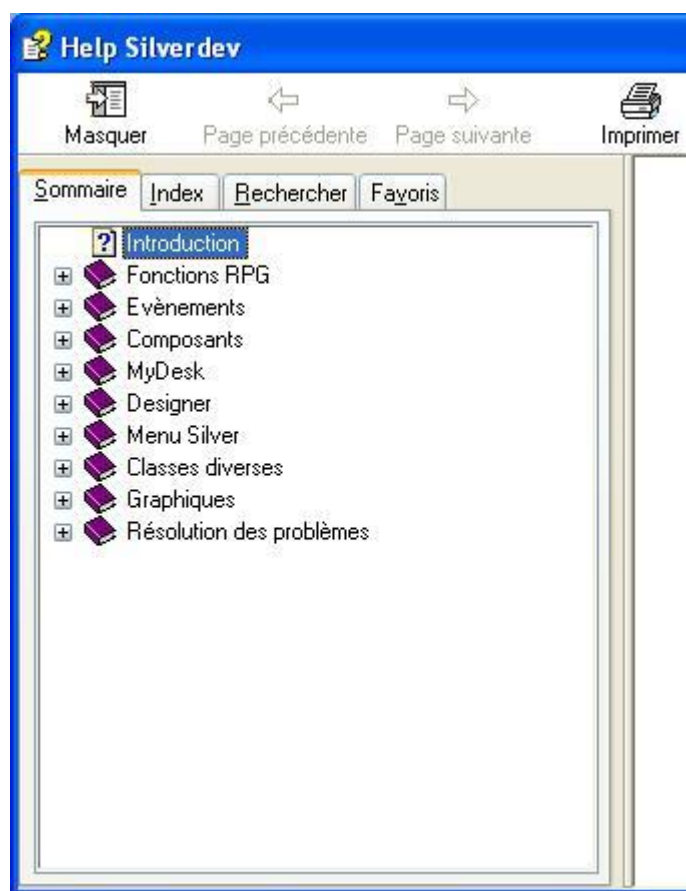


Figure 22

## Index

L'index permet de rechercher une aide sur un mot clef.

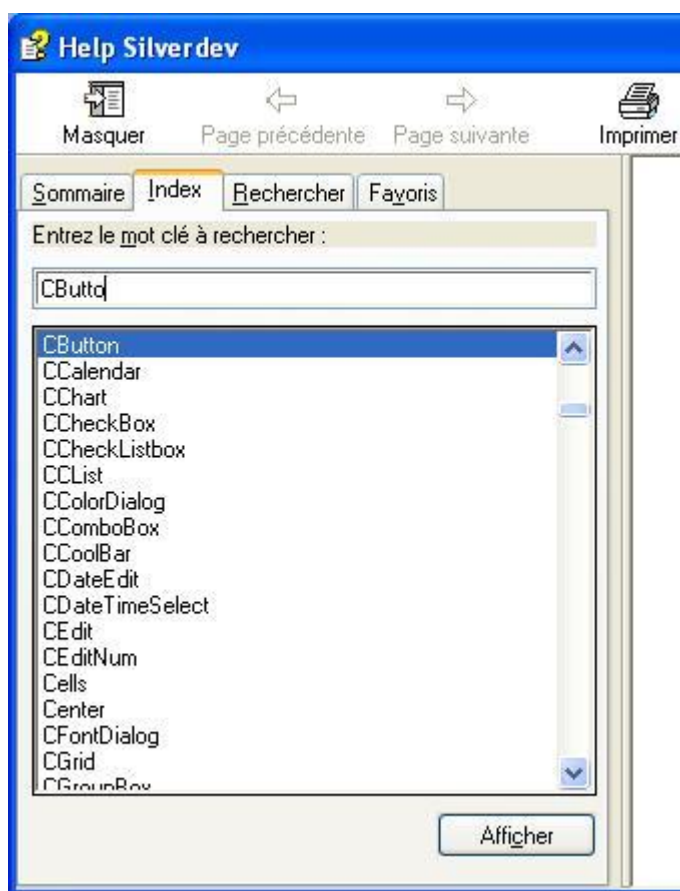


Figure 23

## Recherche

L'onglet recherche permet de rechercher toutes les rubriques dans lesquelles apparait un mot.

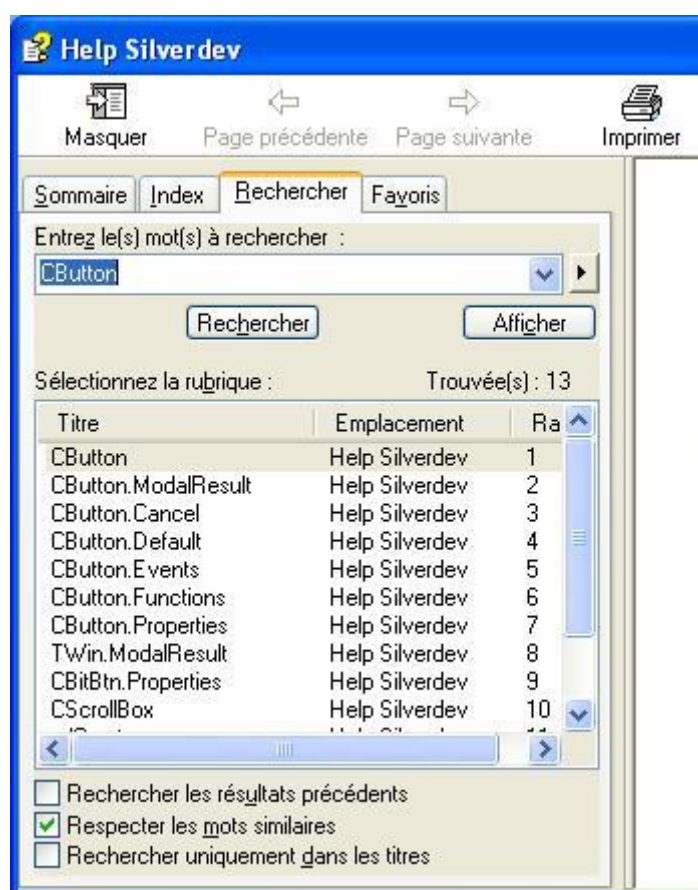


Figure 24

## Page d'aide

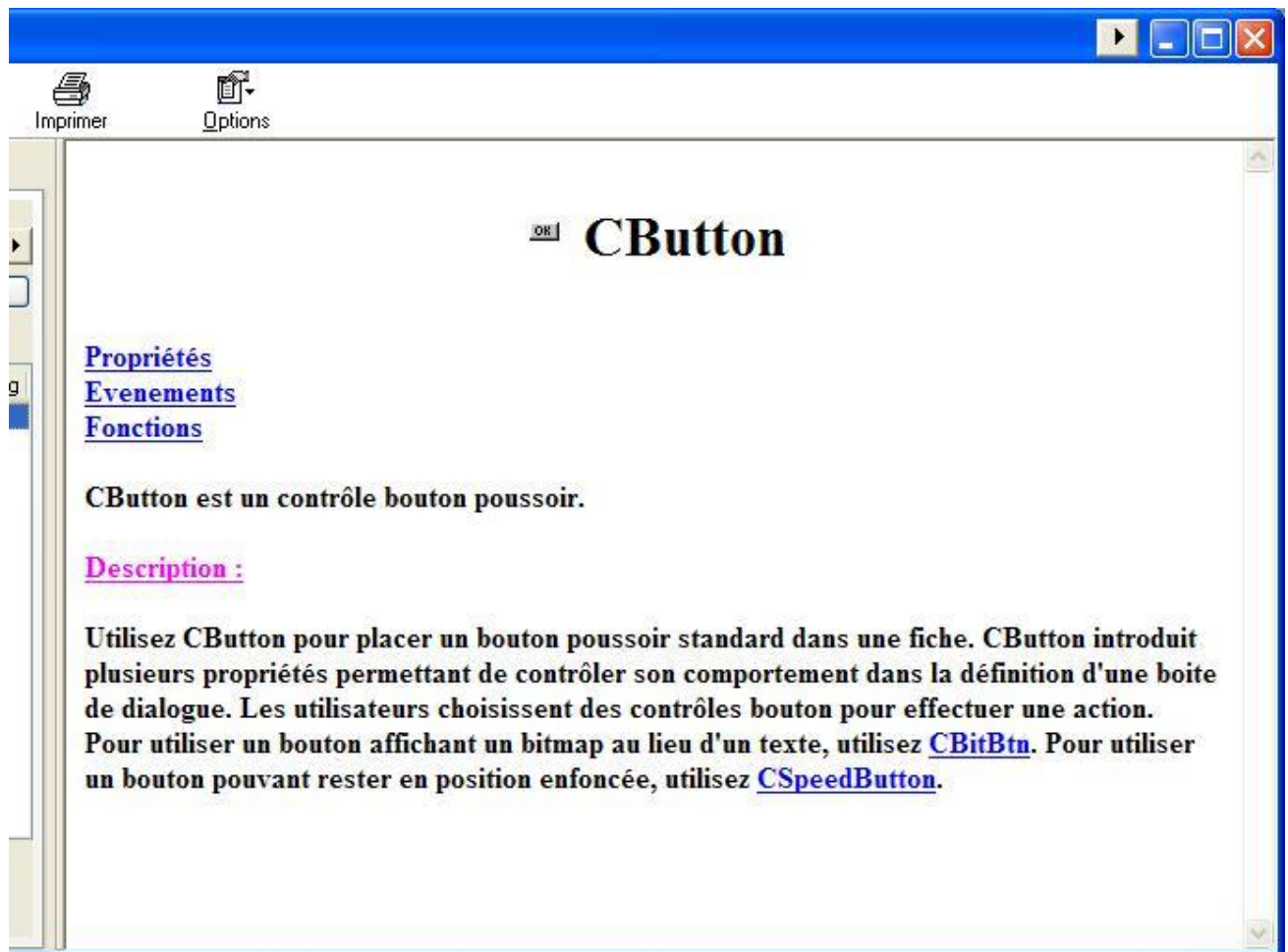


Figure 25

### Aide depuis le réseau

Si le fichier d'aide (fichier chm) est lancé depuis le réseau, le contenu peut être verrouillé pour des raisons de sécurité.

Pour déverrouiller le contenu, modifier l'entrée dans la base de registre

```
REGEDIT4 [HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\HTMLHelp]
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\HTMLHelp\1.x\ItssRestrictions]
"MaxAllowedZone"=dword:00000003
```

Pour modifier l'entrée, vous pouvez utiliser l'utilitaire regedit, ou créer un fichier .reg contenant le texte ci-dessus et double cliquer dessus

Il est aussi possible d'après la documentation de microsoft de corriger le problème en modifiant la valeur UrlAllowedList, mais l'auteur de ces lignes n'a pas réussi

---

## Menu Fiches

### Nouvelle fiche

Le menu "Fiches/Nouvelle fiche" permet de créer une nouvelle fiche.

Il est possible de créer une nouvelle fiche, même si d'autres fiches sont en cours de travail. Designer permet de travailler avec plusieurs fiches en même temps.

### Ouvrir fiche

Le menu "Fiches/Ouvrir fiche" permet d'ouvrir une fiche existante. Les sources des écrans sont stockés sur l'ifs. Il n'est pas nécessaire d'avoir accès à l'ifs par le réseau. En revanche, il faut que le serveur soit démarré et que le designer soit connecté avec le serveur.

Si vous essayez d'ouvrir une fiche sans être connecté, le designer vous proposera une boîte de dialogue pour vous connecter.

Il est possible d'ouvrir plusieurs écrans sources en même temps.

Si l'écran source que vous demandez est déjà ouvert, celui ci sera réactivé. Il ne peut être ouvert deux fois.

L'ouverture d'un fichier source dans le designer verrouille ce fichier.

Un source ne peut pas être ouvert par deux utilisateurs en même temps dans le designer.

### Fiches récentes

Afin de retrouver plus rapidement les écrans sur lesquels vous avez précédemment travaillé, les chemins des 20 derniers écrans sur lesquels vous avez travaillé sont mémorisés. Pour ré-ouvrir un écran, double cliquez sur son nom.

Utilisez le menu Supprimer pour enlever les écrans de la liste de vos récents. (Les écrans ne sont pas supprimés sur l'ifs)

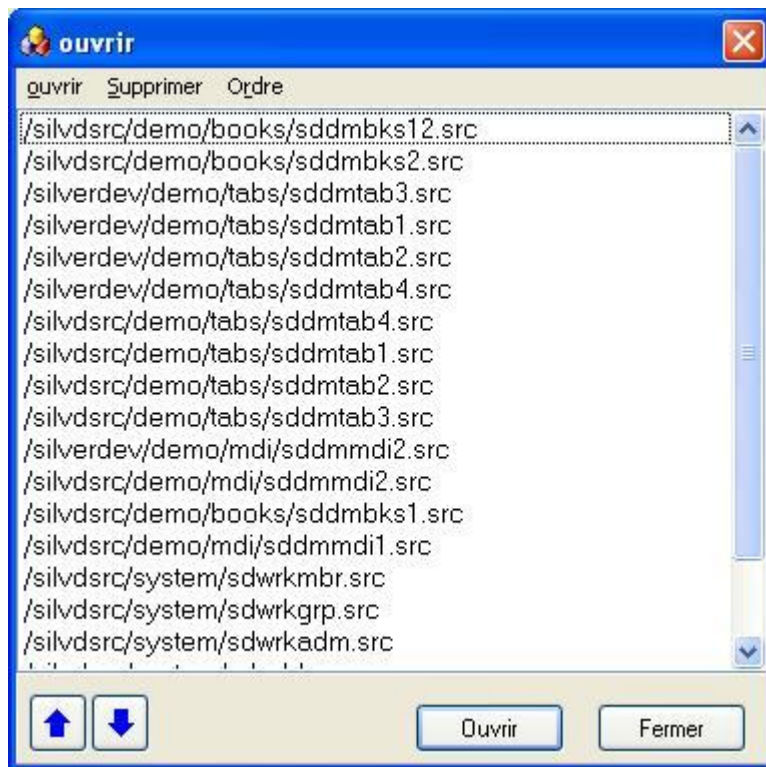


Figure 26

### Nouveau à partir d'un modèle

Le menu "Fichier/Nouveau à partir d'un modèle" permet de créer une nouvelle fiche en se basant sur une fiche existante.

### Fiches ouvertes

Le menu "Fiches ouvertes" permet d'activer une des fiches ouvertes. Celle qui est active est repérée en gras.



Figure 27

## Enregistrer

Le menu "Fichier/Enregistrer" permet de sauvegarder une fiche sur l'ifs.

Il n'est pas nécessaire d'avoir accès à l'ifs par le voisinage réseau du pc. En revanche, il faut que le serveur soit démarré et que le designer soit connecté avec le serveur.

Le chemin sur lequel la fiche sera enregistrée est affiché en haut du designer.

Si la fiche en cours n'a jamais été sauvegardée, une boîte de dialogue vous sera proposée comme si vous aviez utilisé le menu "Fichier/Enregistrer sous".

## Enregistrer sous...

Le menu "Fichier/Enregistrer sous" permet de choisir l'emplacement où la fiche en cours sera sauvegardée.

Si vous ouvrez une fiche à un emplacement et que vous faites "Enregistrer sous", toutes les prochaines sauvegardes seront faites à ce nouvel emplacement.

## Tout enregistrer

Le menu "Fichier/Tout enregistrer" permet de sauvegarder toutes les fiches ouvertes.

A chaque fois qu'une fenêtre n'a pas de nom de source associé, Designer vous demandera où sauvegarder la fiche.

## Compiler

Le menu "Fichier/Compiler" permet de générer l'objet écran sur l'iSeries/400.

Cette opération nécessite d'être connecté au serveur SILVERDEV.

Le designer vous demandera toujours où vous voulez compiler l'écran.

Ce menu n'est accessible que si la fiche en cours a un nom de source et n'est pas modifiée.

Une boîte de dialogue vous est proposée. Par défaut la zone Fichier prend le nom du fichier source. La bibliothèque ne prend pas de valeur par défaut, mais d'une compilation à l'autre même pour des écrans différents, le texte saisi est mémorisé. (Mais pas entre deux lancements du Designer)

Le bouton à droite de la zone permet de sélectionner une bibliothèque.

Si vous commencez à saisir une bibliothèque, seules celles commençant par les mêmes lettres seront affichées lors de l'appui sur le bouton « ... ».



Figure 28

*Remarque :*

*Par défaut, la case à cocher « Ecraser le fichier existant » est décochée.  
Si vous souhaitez que cette case soit cochée par défaut, allez dans le menu « préférences/config »*

### Contrôle de la compilation

Si vous souhaitez contrôler la compilation, vous pouvez ajouter un exit program à l'exit point SVDCTRLCPLSDF à l'aide de la commande WRKREGINF.

Le programme doit avoir les paramètres suivants :

C	*entry	plist		
C		parm	JobUser	10
C		parm	Lib	10
C		parm	UsrSpc	10
C		parm	Ret	1
C		parm	Texte	256

Les trois premiers paramètres sont en entrées, les deux suivants en sortie.

Si le paramètre Ret vaut 'N' la compilation sera abandonnée et le texte de la variable Texte sera affiché dans le designer.

### Fermer

Ferme la fiche en cours.

Si la fiche en cours a été modifiée, une boîte de dialogue vous demande si vous voulez la sauvegarder.

Le bouton "OUI" permet de sauvegarder la fiche avant de la fermer. Le bouton "NON" ferme la fiche sans la sauvegarder. Le bouton "ANNULER" annule l'opération de fermeture.

### Tout fermer

Ferme toutes les fiches ouvertes.

Pour chaque fiche modifiée, le designer vous demande si vous voulez la sauvegarder.



Si vous choisissez "OUI", le designer sauvegarde la fiche et continue de fermer les fiches.  
Si vous choisissez "NON" le designer ferme la fiche sans la sauvegarder, et continue la fermeture des fiches.  
Si vous choisissez "ANNULER", Designer arrête la fermeture des fiches.

## Préférences

Le menu "Configuration/Préférences" permet d'afficher la boîte de dialogue suivante

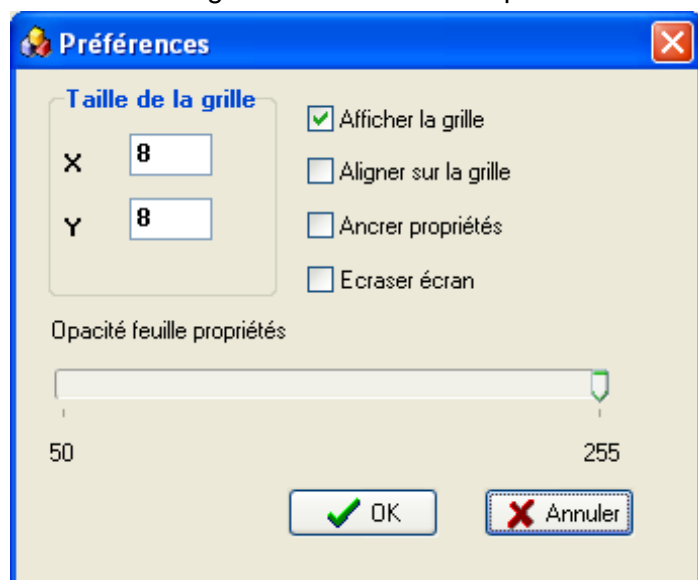


Figure 29

Les zones X et Y permettent de modifier le quadrillage présent sur les fiches.  
La zone "Afficher la grille" permet d'afficher ou non le quadrillage présent sur les fiches.  
La zone "Aligner sur la grille" permet de modifier le comportement du designer.  
Si la case est cochée, alors les composants qui sont créés sont alignés sur les traits les plus proches de l'endroit où le composant est déposé.  
La zone "Ancrer propriétés" vous permet d'indiquer si vous souhaitez que l'inspecteur de propriétés soit toujours au-dessus des autres fenêtres.  
La zone "Opacité" vous permet de rendre l'inspecteur de propriétés transparent. (Cette fonctionnalité ne marche que sous windows 2000 ou windows XP).  
La zone « Ecraser écran » permet de déterminer si la case à cocher « Ecraser écran » est cochée par défaut lorsque vous compilez un écran.

---

## Déconnexions

Lorsque vous travaillez avec le designer et qu'une déconnexion a lieu sur le réseau, vous ne serez pas prévenu automatiquement.  
Vous serez prévenus au moment où vous essaieriez de sauvegarder le source.  
Vous aurez alors le message suivant :

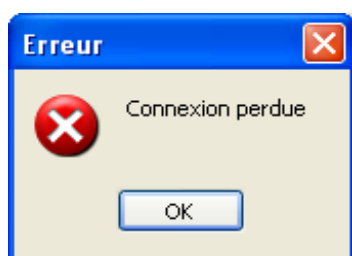


Figure 30

Dans ce cas, utilisez le menu déconnexion puis reconnectez vous.  
Attention le job de l'ancienne connexion est toujours actif, et verrouille votre source.

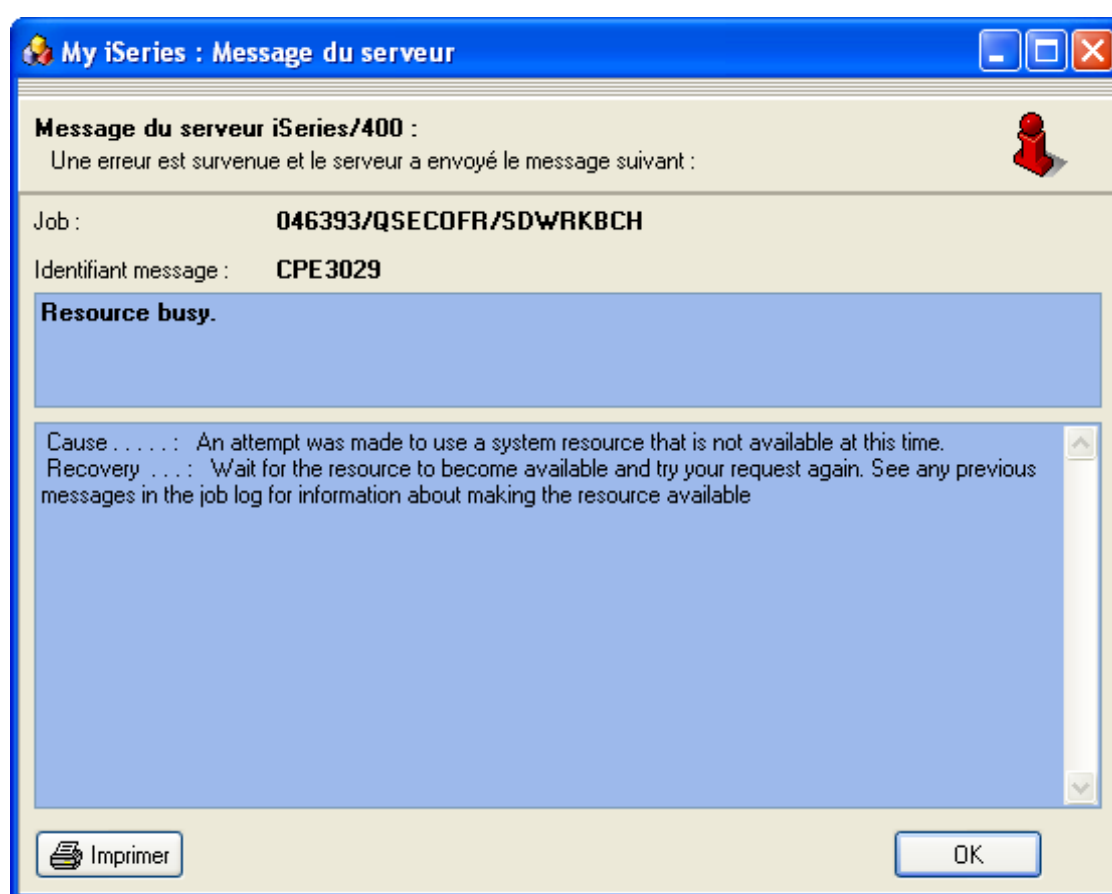


Figure 31

Il vous faut arrêter ce job pour libérer le source.

---

## Chapitre 5. Editeur de sources

---

### Menu membres

L'éditeur peut être utilisé pour une source non silverdev en utilisant le menu Membres. Choisissez le menu Membres/Ouvrir pour ouvrir une source.



Figure 32

---

### Fenêtre associée

Si la source est ouverte depuis le contexte, une fenêtre est associée à la source.

Si la source est ouverte depuis le menu Membres, il est encore possible de lui rattacher une fenêtre depuis le menu Membres/Lier à un écran.

Lorsqu'une fenêtre et une source sont liées, si vous double-cliquez dans l'inspecteur de propriétés sur un événement, l'éditeur prend le focus au niveau du gestionnaire de l'événement. Si le gestionnaire n'existe pas, il est créé dans la source.

Pour basculer entre la source et la fenêtre, utilisez la touche F10.

## Compilation

### Commande

Si le source a été ouvert depuis le menu membre, la commande de compilation dépend du type du source.

Si le source a été ouvert depuis un contexte, la commande de compilation dépend de la configuration du contexte.

Dans tous les cas, le programme peut être compilé avec le menu compilation.

### Contrôle d'intégrité

Lorsque que le source est lié à une fenêtre, l'éditeur vérifie au moment de la compilation que tous les événements de la fenêtre ont un gestionnaire d'événement et que tous les gestionnaires d'événement correspondent à un événement de la fenêtre.

Si ce n'est pas le cas, un message d'information apparaît, mais n'empêche pas la compilation.

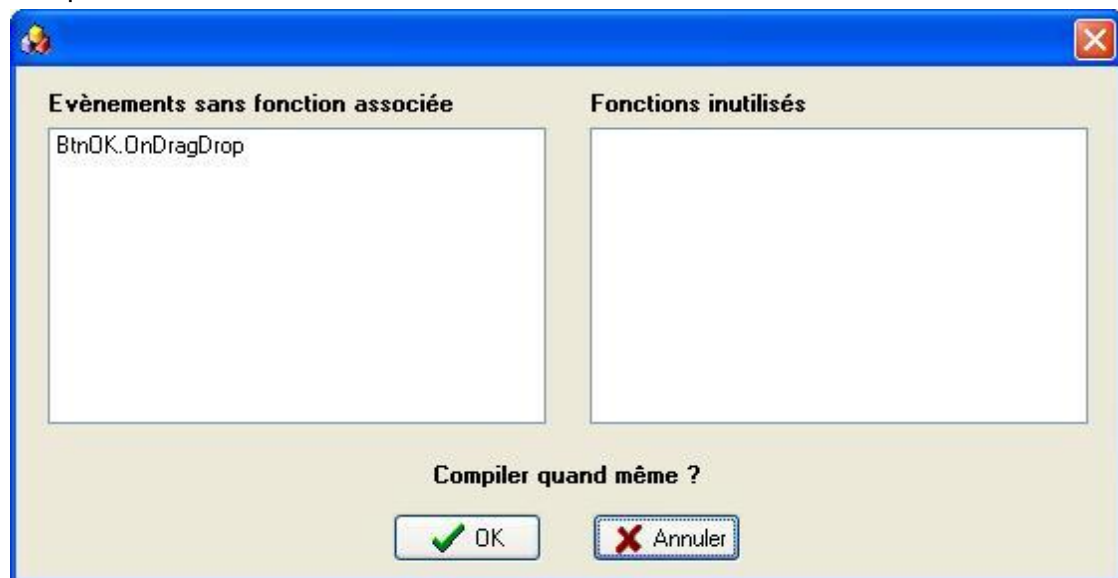


Figure 33

### Résultats de compilation, Spools et travaux soumis

Si la compilation est soumise, le travail soumis est affiché. Il est alors possible d'afficher les spools du job soumis par un click droit.

Si la compilation n'est pas soumise, l'éditeur affiche les résultats de compilation.

Identifiant	Ligne	Colonne	Niveau	Message	Fichier source
<b>RNS9308</b>	<b>0</b>	<b>0</b>	<b>50</b>	<b>Arrêt de la compilation. Erreurs de gravité 30 détectées dans</b>	<b>ARNAUD/QRPGLESRC(ABC2)</b>
<b>RNF7030</b>	<b>77</b>	<b>47</b>	<b>30</b>	<b>Le nom ou l'indicateur FORMCOUNT n'est pas défini.</b>	<b>ARNAUD/QTXTLESRC(ABC2)</b>
<b>RNF5377</b>	<b>77</b>	<b>47</b>	<b>20</b>	<b>Fin de l'expression escomptée.</b>	<b>ARNAUD/QTXTLESRC(ABC2)</b>
RNF7031	88	7	0	Le nom ou l'indicateur WIN n'est pas référencé.	ARNAUD/QTXTLESRC(ABC2)
RNF7031	70	7	0	Le nom ou l'indicateur PARAMETERS n'est pas référencé.	ARNAUD/QTXTLESRC(ABC2)
RNF7031	87	7	0	Le nom ou l'indicateur PARAMETERS n'est pas référencé.	ARNAUD/QTXTLESRC(ABC2)
RNF7031	89	7	0	Le nom ou l'indicateur EVT n'est pas référencé.	ARNAUD/QTXTLESRC(ABC2)
RNF7031	60	7	0	Le nom ou l'indicateur EVT n'est pas référencé.	ARNAUD/QTXTLESRC(ABC2)
RNF7031	59	7	0	Le nom ou l'indicateur WIN n'est pas référencé.	ARNAUD/QTXTLESRC(ABC2)
RNF7031	58	7	0	Le nom ou l'indicateur PARAMETERS n'est pas référencé.	ARNAUD/QTXTLESRC(ABC2)
RNF7031	72	7	0	Le nom ou l'indicateur EVT n'est pas référencé.	ARNAUD/QTXTLESRC(ABC2)
RNF7031	71	7	0	Le nom ou l'indicateur WIN n'est pas référencé.	ARNAUD/QTXTLESRC(ABC2)
RNF7031	1276	7	0	Le nom ou l'indicateur SDMETHOD n'est pas référencé.	SILVERDEV/H(SILVERDEV)
RNF7031	623	7	0	Le nom ou l'indicateur SDMOVENODE n'est pas référencé.	SILVERDEV/H(SILVERDEV)
RNF7031	692	7	0	Le nom ou l'indicateur SDLOADF... n'est pas référencé.	SILVERDEV/H(SILVERDEV)
RNF7031	774	7	0	Le nom ou l'indicateur SDMERGE n'est pas référencé.	SILVERDEV/H(SILVERDEV)
RNF7031	447	7	0	Le nom ou l'indicateur SDNEW/M&P n'est pas référencé.	SILVERDEV/H(SILVERDEV)

Figure 34

Les erreurs sont en gras, et les messages d'information sont en police normale.

Double cliquez sur une des lignes pour ouvrir le source à l'emplacement de l'erreur.

Utilisez les boutons  et  pour afficher respectivement les spools et les message de log qui ont été créés pendant la compilation.

#### Remarque :

Pour un source de type SQLRPGLE, les numéros de ligne des résultats de compilation font référence au fichier temporaire (QTEMP/QSQLTEMP1) créé par le compilateur. Modifier ce source est inutile puisqu'il sera recréé à la prochaine compilation.

## Complétion de code

### Fonctions

Utilisez le raccourci Ctrl+Espace pour afficher la liste des fonctions silverdev.

La liste déroulante permet d'insérer une fonction. Cette liste déroulante n'affiche que les fonctions commençant comme le mot en cours d'édition.

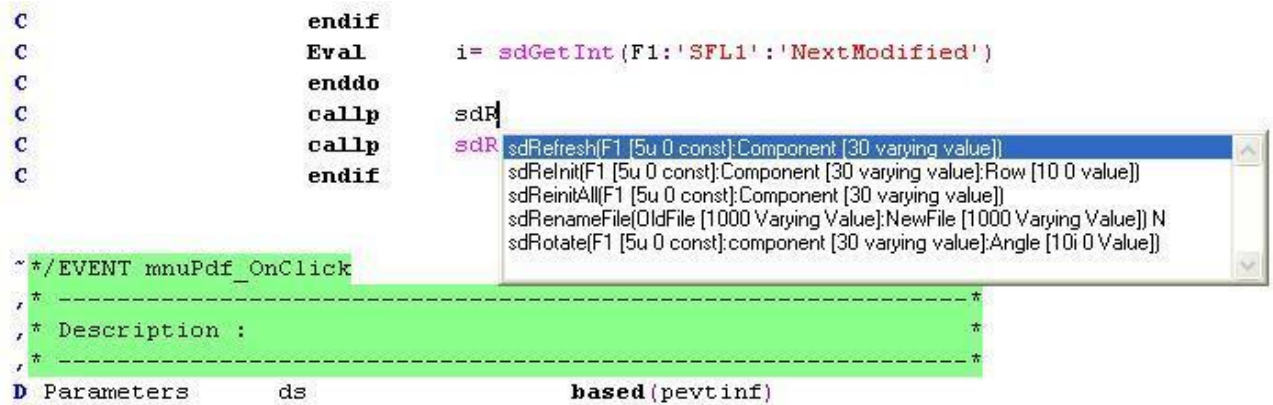


Figure 35

## Arborescence fenêtres

Utilisez la touche F2 pour faire apparaître un arbre affichant les objets et les propriétés d'une fenêtre. Si une fenêtre est liée à un source, cette fenêtre apparaît dans l'arborescence.

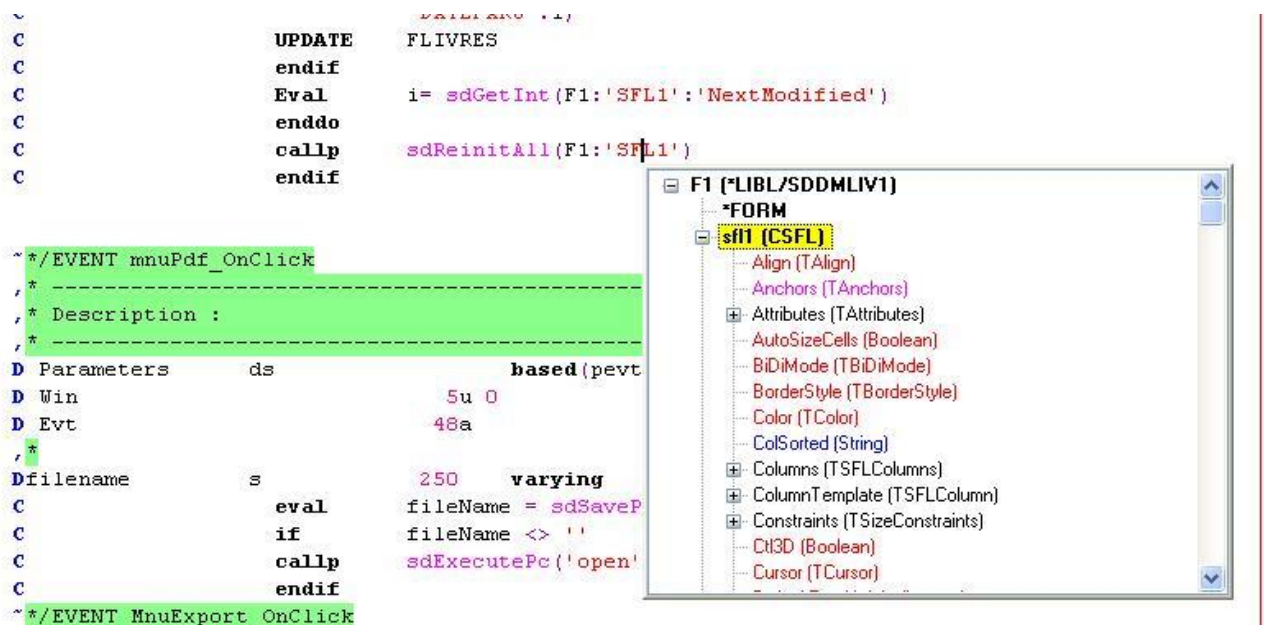


Figure 36

Il est possible d'ajouter d'autre fenêtres en utilisant le menu "Source/Charger un écran compilé" ou "Source/Charger un écran en design". Faites F5 ou utilisez le menu contextuel pour rafraîchir un nœud.

*Remarque :*

*Ne confondez pas la fenêtre associée au source avec les fenêtres chargées.*

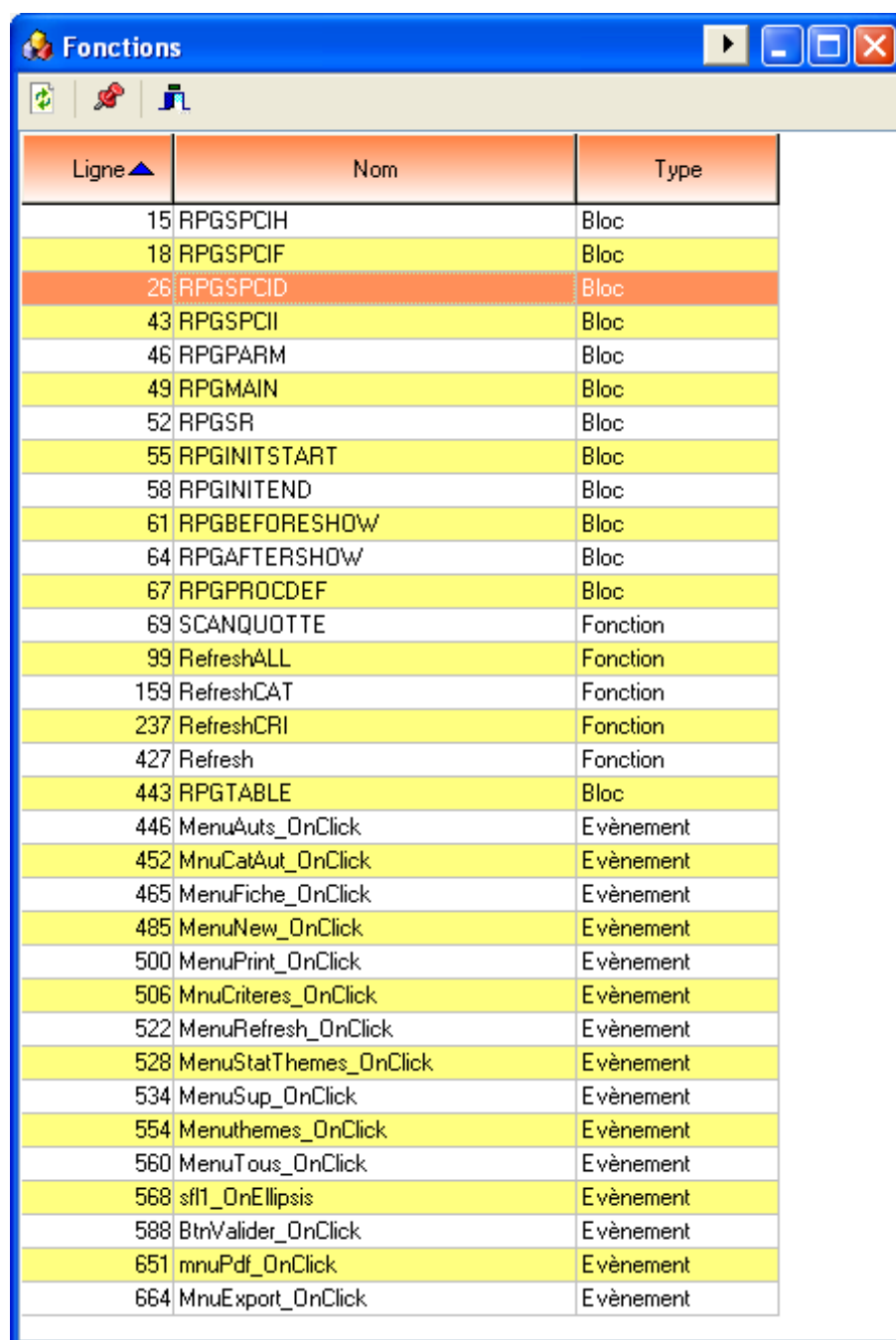
*Il ne peut y avoir qu'une seule fenêtre associée, mais il peut y avoir plusieurs fenêtres chargées. La fenêtre associée est automatiquement chargée.*

---

## Liste des fonctions et moments

Pour afficher la liste des fonctions dans un source RPGLE, utilisez le menu "Chercher/Liste fonctions".

Double cliquez sur une ligne de la liste pour déplacer le curseur au niveau de la définition de la fonction.



Ligne ▲	Nom	Type
15	RPGSPCIH	Bloc
18	RPGSPCIF	Bloc
26	RPGSPCID	Bloc
43	RPGSPCII	Bloc
46	RPGPARM	Bloc
49	RPGMAIN	Bloc
52	RPGSR	Bloc
55	RPGINITSTART	Bloc
58	RPGINITEND	Bloc
61	RPGBEFORESHOW	Bloc
64	RPGAFTERSHOW	Bloc
67	RPGPROCDEF	Bloc
69	SCANQUOTTE	Fonction
99	RefreshALL	Fonction
159	RefreshCAT	Fonction
237	RefreshCRI	Fonction
427	Refresh	Fonction
443	RPGTABLE	Bloc
446	MenuAuts_OnClick	Evènement
452	MnuCatAut_OnClick	Evènement
465	MenuFiche_OnClick	Evènement
485	MenuNew_OnClick	Evènement
500	MenuPrint_OnClick	Evènement
506	MnuCriteres_OnClick	Evènement
522	MenuRefresh_OnClick	Evènement
528	MenuStatThemes_OnClick	Evènement
534	MenuSup_OnClick	Evènement
554	Menuthemes_OnClick	Evènement
560	MenuTous_OnClick	Evènement
568	sf11_OnEllipsis	Evènement
588	BtnValider_OnClick	Evènement
651	mnuPdf_OnClick	Evènement
664	MnuExport_OnClick	Evènement

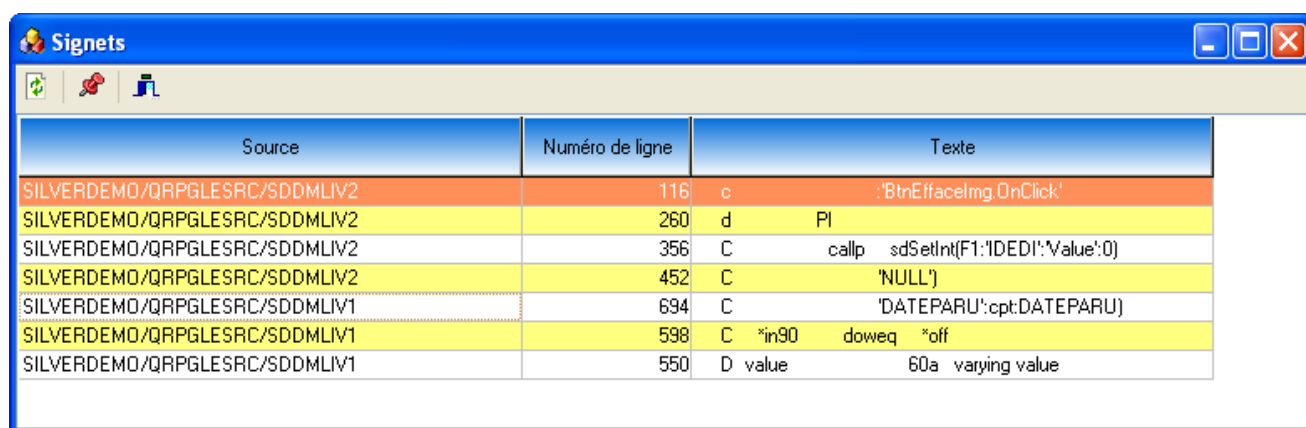
Figure 37

## Signets

Les signets permettent de marquer une ligne de texte, afin de la retrouver rapidement. Chaque source peut avoir jusqu'à 10 signets. Utilisez les raccourcis Maj + Ctrl + n pour ajouter un signet (n est le numéro du signet) et Ctrl + n pour aller au signet.

Pour afficher les signets, utilisez le menu "Chercher/Voir signets".





Source	Numéro de ligne	Texte
SILVERDEMO/QRPGLESRC/SDDMLIV2	116	c :BtrEfacelmg.OnClick'
SILVERDEMO/QRPGLESRC/SDDMLIV2	260	d PI
SILVERDEMO/QRPGLESRC/SDDMLIV2	356	C callp sdSetInt(F1:'IDED!':Value:0)
SILVERDEMO/QRPGLESRC/SDDMLIV2	452	C 'NULL')
SILVERDEMO/QRPGLESRC/SDDMLIV1	694	C 'DATEPARU':cpt:DATEPARU)
SILVERDEMO/QRPGLESRC/SDDMLIV1	598	C *in90 doweq *off
SILVERDEMO/QRPGLESRC/SDDMLIV1	550	D value 60a varying value

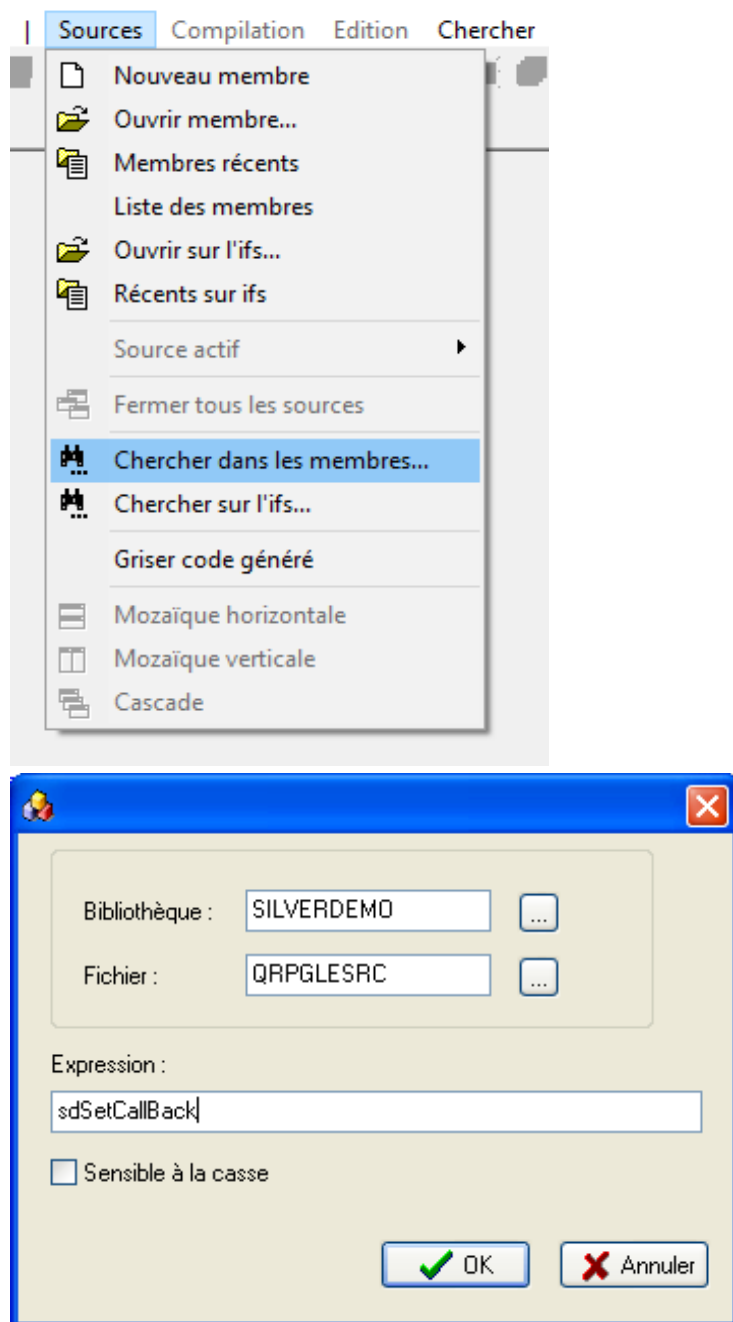
Figure 38

## Raccourcis

Ctrl + Espace	Liste des fonctions Silverdev
F2	Arborescence de fenêtres
F10	Voir la fenêtre associée
F8	Faire apparaître/disparaître la règle
Ctrl + n	Aller au bookmark n
Maj + Ctrl + n	Ajouter le bookmark n
Maj + Ctrl+ N	Selection normale
Maj + Ctrl + C	Selection en colonne
Maj + Ctrl + L	Selection en ligne
Maj + Ctrl + I	Indentation de bloc
Maj + Ctrl + U	Désintention de bloc
Maj + Ctrl + B	Trouver la parenthèse correspondante

## Recherche d'une expression dans un source

Utilisez le menu Sources/Chercher dans les membres

**Figure 39**

Appuyez sur Echap pour arrêter la recherche.

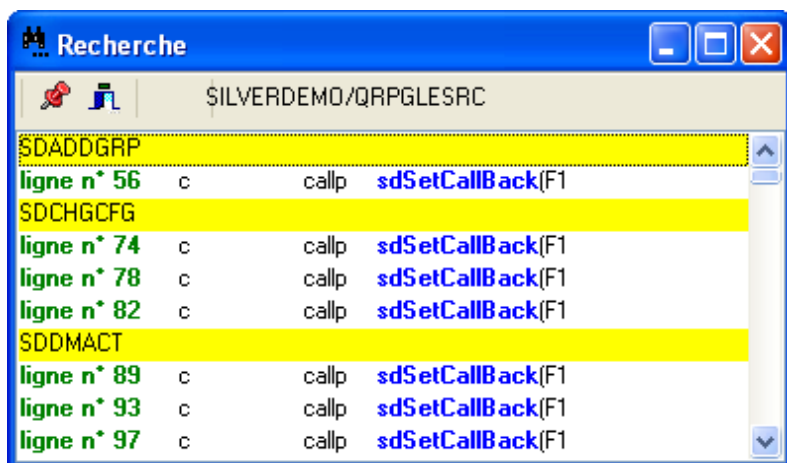


Figure 40

Double cliquez sur une ligne pour ouvrir le membre.

## Chapitre 6. Générateur

### Source RPG IV

Le code saisi par le développeur et le code généré sont stockés dans des sources différents (respectivement QTXTLERSC et QRPGLERSC).

Ainsi, à chaque lancement de la génération, le code du développeur n'est pas écrasé.

L'affichage des fenêtres et les associations d'évènement aux fonctions (Callback) sont générés. Le développeur peut alors se concentrer sur les actions à réaliser lors des évènements.

De plus, il est possible de demander de générer des fonctions de lecture et d'écriture. A cette fin, certains composants possèdent des propriétés spécifiques au générateur. Ces propriétés sont regroupées dans une propriété objet nommée OptionsGeneration. Voir **Erreur ! Source du renvoi introuvable.**

Lors de la génération, d'un source, le générateur s'appuie sur la description de la fenêtre (indiquée dans option 8) et sur le membre source de QTXTLERSC.

Le membre du fichier RPG généré dans QRPGLERSC aura le même nom que le membre QTXTLERSC.

#### **\*/BLOCK**

Les différentes parties du source de QTXTLERSC sont identifiées par des commentaires spéciaux : **\*/BLOCK**

Les blocks correspondent aux différentes sections traditionnelles d'un programme RPG.

Un commentaire **\*/BLOCK** est suivi d'un espace et du nom du Block

Nom de Block	Description
RPGSPCIH	Spécifs H
RPGSPCIF	Spécifs F
RPGSPCID	Spécifs D
RPGSPCII	Spécifs I
RPGPARM	Début de programme, sert à définir les paramètres
RPGPGMSTART	Intructions de début de programme.
RPGSR	Sous routines
RPGBEFORECREATE	Début de la procédure d'init.
RPGAFTERCREATE	Après création de la fenêtre et call backs.
RPGBEFORESHOW	Juste avant l'appel à sdShow ou sdShowmodal . Idéal pour rafraichir le contenu d'un écran.

	Remarque : ce moment peut être utilisé même si le mode d'affichage de la fenêtre est *NOSHOW.
RPGAFTERSHOW	Juste après l'appel à sdShow ou sdShowModal.
RPGPROCDEF	Définition de procédures "utilisateurs" (non événements)
RPGTABLE	Données des tables CTDATA

Exemple :

```
*/BLOCK RPGSpCiD
D Cpt          S          5u 0
```

## Gestionnaires d'évènements : \*/EVENT

En plus des blocs \*/BLOCK, il y a les blocs \*/EVENT pour l'ajout de code dans les procédures des évènements. On parle dans ce cas de « gestionnaire d'évènement ». Le générateur crée un bloc \*/EVENT pour chaque événement de l'écran coché dans le designer.

### Exemple :

```
*/EVENT SFL1_OnChangeCellValue
* -----*
* Description :                               *
* -----*
D Parmameters      ds                      based(pevtinf)
D Win              5u 0
D Evt              48a
* Modifiable
D Allow            N
* Non modifiable
D ColName          100a   varying
D Row              10i 0
D Value            100a   varying
*
```

#### Remarque :

Si l'évènement possède des paramètres, ceux ci sont ajoutés par le générateur

#### Remarque :

Le générateur produit les noms de bloc \*/EVENT de la fiche principale.  
La longueur totale d'un gestionnaire d'évènement est limitée à 41 caractères.

Comme pour les \*/BLOCK, les \*/EVENT sont insensibles à la casse.

### Exemple :

```
*/BLOCK RPGSpclD
D Cpt              S                      5u 0
*/EVENT Timer1_OnTimer
* -----*
* Description :                               *
* -----*
D Parmameters      ds                      based(pevtinf)
D Win              5u 0
D Evt              48a
```

```
C          Eval          Cpt = Cpt + 1
C          Callp          SdSetString(F1:'Label1':'Caption':NOMCLI)
  */EVENT ButtonRefresh_OnClick
* -----*
* Description :                                           *
* -----*
D Parmameters      ds          based(pevtinf)
D Win              5u 0
D Evt              48a
C          Callp          Timer1_OnTimer(Pform)
```

## Protection de QRPGLSRC

Le générateur s'appuie sur le source de qtxtlesrc pour générer un source dans qrpglesrc. Le générateur n'écrase pas le source de qrpglesrc si celui ci n'avait pas été généré par silverdev.

```

Message . . . . : Le membre source existe déjà, mais n'est pas un source
                  généré par SILVERDEV : REPLACE(*YES) non pris en compte. Génération annulée.
Source member already exists, but has not been recongnized as generated by
SilverDev. REPLACE(*YES) ignored. To protect your sources, generation
process has been aborted. Give another member name and retry
0  07/11/07 10:35:52,100640 GSVDESCRT TTOBJ *STMT GSVDESCRT
From module . . . . . : GSVDESCRT
From procedure . . . . . : GSVDESCRT
Statement . . . . . : 250
To module . . . . . : GSVDESCRT
To procedure . . . . . : _QRNP_PEP_GSVDESCRT
Statement . . . . . : *N
Message . . . . : Source generation Failed *****.
Cause . . . . . : Ce message est utilisé par les programmes d'application
                  comme message général d'arrêt programme.

```

## PSVDCMP

Le fichier PSVDCMP permet de définir les fonctions qui seront générées par le générateur. Ce fichier n'a pas normalement à être modifié.

Exemple :

Component Type:	CEDITNUM
Component Default property:	VALUE
Component Set Function:	sdSetNum
Component Get Function:	sdGetNum
Component Level:	01
Component Default gen:	Y

Cet enregistrement indique que pour les composants de type CEditNum, silverdev générera le code callp sdSetNum(F1 :'editnum1' :'value' :editnum1) dans la fonction d'écriture et le code eval editnum1=sdGetnum(F1 :'editnum1' :'value') dans la fonction de lecture.



## Source généré

### Eléments de base d'un programme

Un programme Silverdev commence toujours par :

```
H BNDDIR('SILVERDEV') DFTACTGRP(*no)
```

Ceci afin de lier le programme au service programme SILVERDEV/SDSRVPGM où sont définies les fonctions utiles.

Dans une spécification D, il faut ajouter la ligne suivante :

```
/COPY H,SILVERDEV
```

Le membre SILVERDEV/H/SILVERDEV contient les prototypes des fonctions définies dans SILVERDEV/SDSRVPGM.

Le programme est constitué des lignes suivantes :

```
C          callp      sdStart(%paddr('INIT'))
C          eval       *inlr = *on
```

Tout le reste du programme est constitué de sous-procédures. Notamment la procédure Init qui est la procédure appelée par sdStart.

Dans la procédure Init, faites un appel à la fonction sdCreateForm qui enverra une première fiche. Le programme s'arrêtera lorsque l'utilisateur fermera la première fiche(ou lorsque vous la fermerez par programmation).

Après la création d'une fiche, il faut associer les évènements qui vont être reçus à une fonction que vous créerez et que l'on appellera gestionnaire d'évènement. Cette association se fait à l'aide de la fonction sdSetCallBack.

En définitif, la fonction Init ressemblera à ceci :

```
P Init          B
D              PI
C              eval      F1 =sdCreateForm('*LIBL/LIVRES')
C              callp     sdSetCallBack(F1 :
C                      'BtnValider.OnClick':%paddr('VALIDATION'))
P              E
```

Remarque 1: La variable F1 est déclarée comme ceci :

```
D F1          S          5u 0
```

Remarque 2: Si vous souhaitez effectuer des traitements juste après avoir envoyé la première fenêtre (Initialisation de champs par exemple) il est possible de le faire dans la fonction Init après les appels à sdSetCallback.

Dans l'exemple précédent, la fonction sdSetCallBack associe le gestionnaire d'évènement VALIDATION à l'évènement BtnValider.OnClick.

Remarque 3: Le nom de la fonction *VALIDATION* doit être écrit en majuscule.

Pour connaître la liste des événements qui seront signalés, utilisez le menu "voir/Récapitulation" dans le designer.

La fonction *VALIDATION* est une sous procédure classique RPG ILE qui s'exécute à chaque fois que l'utilisateur clique sur le bouton BtnValider.

Toutes les fonctions nécessaires pour manipuler les composants situés sur la fiche sont fournies dans le service programme SDSRVPGM. Voir la liste dans le fichier d'aide Silverdev.hlp.

Le programme se termine côté client lorsque la fenêtre principale est fermée (soit par l'utilisateur, soit par code). La fenêtre principale est la première fenêtre envoyée au client.

### **sdStart**

Lorsque la fonction sdStart est appelée pour la première fois, cette fonction appelle la fonction dont l'adresse lui est passée en paramètre puis entre dans une boucle qui se termine lorsque la partie cliente lui signale que la fenêtre principale a été fermée.

Lorsque la fonction sdStart est appelée ensuite (dans un second programme par exemple) la fonction ne fait que appeler la fonction dont l'adresse lui est passée en paramètre puis se termine. Ce qui signifie que lors de l'appel d'un second programme, celui ci est chargé en mémoire, puis le programme se termine.

Donc si celui ci est dans le groupe d'activation \*NEW, le groupe d'activation est détruit. Le programme est alors déchargé de la mémoire.

Remarque :

***Ne compilez jamais vos programmes dans le groupe d'activation \*NEW.***

*Si votre programme est appelé par un autre, vous auriez l'erreur suivante:*

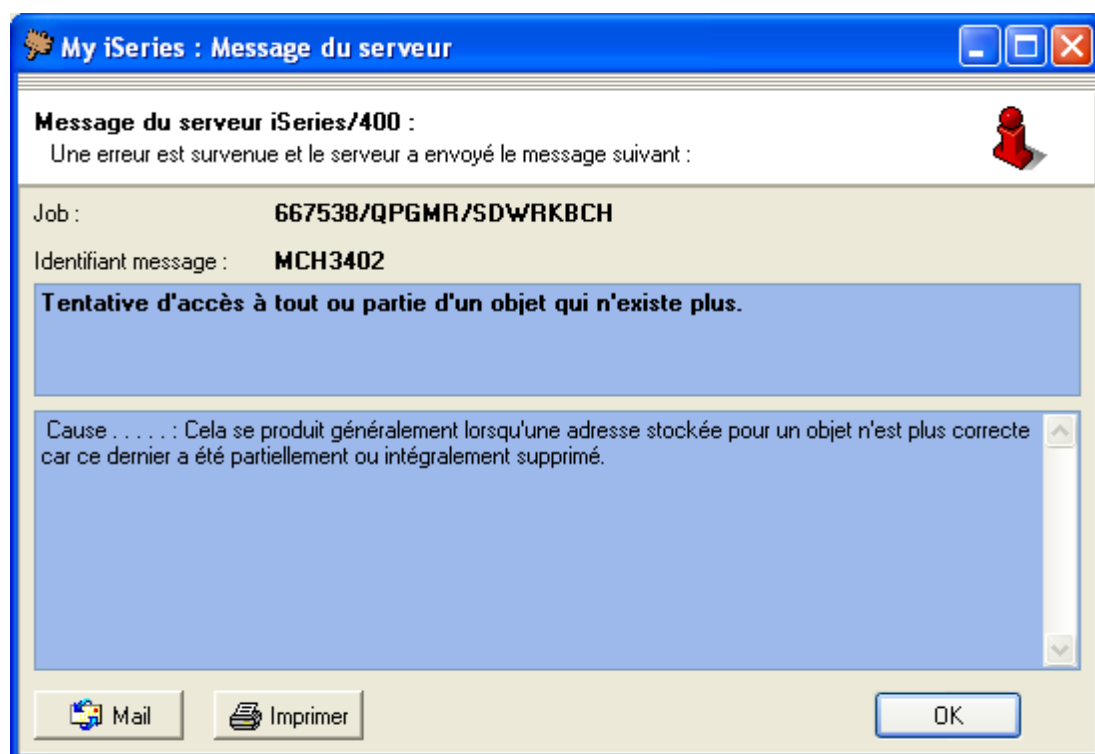


Figure 41

Pour exemple, recompilez le programme sddmthm avec l'instruction `actgrp(*NEW)`

## Chapitre 7. Mode RT ou LR

Le choix du mode LR dans un programme silverdev n'est pas anodin.

Si vous choisissez le mode LR, il faut libérer la fenêtre dans l'évènement onclose afin de libérer la mémoire utilisée par la fenêtre côté client. Appelez la fonction sdFreeForm pour informer la partie serveur que la fenêtre n'existe plus côté client. (Ne pas confondre sdFree et sdFreeForm)

```
D Parameters      ds          based(pevtnf)
D Win              5u 0
D Evt              48a
,* Variable
D Action           10i 0
,* 0 : ne rien faire
,* 1 : Cacher la fenêtre
,* 2 : Libérer la fenêtre
,* 3 : Minimiser la fenêtre
,*
c                  eval      action = 2
c                  callp     sdFreeForm(Win)
```

### Résumé :

	RT	LR
<b>Non modal</b>	<p>Une fenêtre est créée au premier appel du programme.</p> <p>A chaque appel du programme, la même fenêtre est utilisée.</p>	<p>Ce cas est de loin le plus complexe.</p> <p>Une fenêtre est créée à chaque appel du programme. Il est alors possible d'avoir plusieurs instances de fenêtre en même temps. Il ne faut donc pas utiliser la variable F1 qui n'enregistre que la valeur de la dernière fenêtre créée.</p> <p>Utilisez le paramètre win dans les évènements.</p> <p>Détruisez la fenêtre dans l'évènement onClose en mettant le paramètre action à 2. (Afin d'éviter les fuites de mémoire côté client)</p> <p>Appelez aussi la fonction sdFreeForm pour informer le serveur que la fenêtre n'existe plus côté client. (Afin d'éviter les fuites de mémoire</p>

		<p>côté serveur et d'éviter les erreurs avec les fonctions de type sdGetForm)</p> <p>Si vous avez des informations à stocker pour chaque fenêtre, il faudra les stocker dans un tableau. Voir Chapitre 20</p>
<b>Modal</b>	<p>Une fenêtre est créée au premier appel du programme.</p> <p>A chaque appel du programme, la même fenêtre est utilisée.</p>	<p>Chaque appel du programme crée une nouvelle fenêtre.</p> <p>Puisque la fenêtre est modale, vous ne pouvez pas rappeler le programme tant que la fenêtre est affichée. Il n'y a donc qu'une seule instance de la fenêtre en même temps. Vous pouvez donc utiliser la variable F1 qui contient la valeur de la dernière fenêtre créée.</p> <p>Détruisez la fenêtre dans l'évènement onClose en mettant le paramètre action à 2. (Afin d'éviter les fuites de mémoire côté client)</p> <p>Appelez aussi la fonction sdFreeForm pour informer le serveur que la fenêtre n'existe plus côté client. (Afin d'éviter les fuites de mémoire côté serveur et d'éviter les erreurs avec les fonctions de type sdGetForm)</p>

## Chapitre 8. Manipulation des composants par programme

Les composants peuvent être modifiés par les fonctions fournies dans le service programme sdsrvpgm.

### sdSetString, sdGet

Les propriétés de type string peuvent être modifiées par la fonction sdSetString.

```
d sdSetString      pr
d  form              5u 0
d  component         30   varying value
d  propertyPath     256a  varying value
d  value            9999a  varying value
d  Trim              N    value options(nopass)
```

Exemple la propriété text d'un composant CEdit :

```
C          callp      SdSetString(F1:'TITRE':
C          'Text':'Objectif lune')
```

Remarque : si la fonction doit s'appliquer sur une propriété de la fenêtre, le paramètre component doit prendre la valeur '\*FORM'

la propriété peut être interrogée à l'aide de la fonction sdGet :

```
d sdGet            pr          9999a  varying
d  form              like(tWHandle)
d  component         30   varying value
d  propertyPath     256a  varying value
```

```
C          Eval      titre=sdGet(form:'Titre  ':'Text')
```

### SdSetNum, sdGetNum

```
d sdSetNum          pr
d  Pform              5u 0
d  Component         30   varying value
d  Property          256  varying value
d  Valeur            30s 5  value
```

d	sdGetNum	pr	30s	5	
d	Pform		5u	0	
d	Component		30		varying value
d	Property		256		varying value

Pour les propriétés de type numérique, utilisez les fonctions sdSetNum et sdGetNum

C	callp	SdSetNum(F1: 'IDEDI': 'Value': 0)
---	-------	-----------------------------------

C	eval	PRIX=sdGetNum(F1: 'PRIX': 'Value')
---	------	------------------------------------

### sdSetBool, sdgetBool

Pour les propriétés de type booléen, utilisez les fonctions sdSetBool et sdGetBool

d	sdSetBool	pr		
d	form			like(tWHandle)
d	component		30	varying value
d	propertyPath		256a	varying value
d	value		n	value

d	sdGetBool	pr	N	
d	form			like(tWHandle)
d	component		30	varying value
d	propertyPath		256a	varying value

C	callp	sdSetBool(F1: 'Button1': 'visible': *off)
---	-------	---

C	if	sdGetBool(F1: 'Date': 'ValidDate')=*off
---	----	---

### sdSetDateTime, sdGetDateTime

Pour les propriétés de type TDateTime, utilisez les fonctions sdSetDateTime et sdGetDateTime.

Le service programme SDSRVPGM fournit plus de 300 fonctions pour manipuler les composants.

## Chapitre 9. Debug côté serveur

### Débug côté serveur avec le designer

Designer permet de déboguer un programme qui s'exécute dans un autre job.

Il est possible de déboguer tous types de programmes, y compris des programmes non silverdev.

Remarque : Il n'est possible de déboguer qu'un job à la fois.

#### Lancement du debug

Utilisez le menu Outils/Debug/Travaux Silverdev.

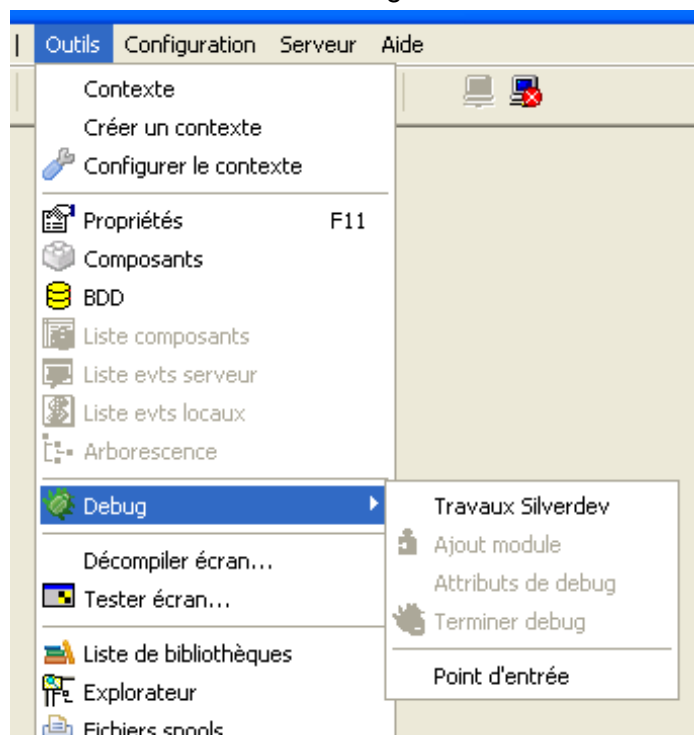


Figure 42

L'écran suivant affiche les jobs du serveur Silverdev.



Travaux

Utilisateur

Travail : SDWRKBCH/SVDSVR/\*ALL Type : Etat : \*ACTIVE

Rang	Nom job	Utilisateur	Numéro travail	Etat	Etat actif	Fonction	Type fonction	File d'attente	Commande silverdev	Utilis
6	SDWRKBCH SVDSVR		091493	*ACTIVE	TIMW	SDWRKBCH	P	*DEV		
5	SDWRKBCH SVDSVR		091492	*ACTIVE	TIMW	SDWRKBCH	P	*DEV		
4	SDWRKBCH SVDSVR		091482	*ACTIVE	SELW	SDWRKBCH	P	*DEV	/Examples/sddmbks	ADUVAL
3	SDWRKBCH SVDSVR		091359	*ACTIVE	SELW	SDWRKBCH	P	*DEV	*MYDESK	ADUVAL
2	SDWRKBCH SVDSVR		091338	*ACTIVE	RUN	SDWRKBCH	P	PRTARNAUD ARNAUD	*DESIGNER	ADUVAL
1	SDWRKBCH SVDSVR		021948	*ACTIVE	TIMW	SDWRKBCH	P	*DEV		

Figure 43

Remarque :

Pour modifier la liste des jobs affichés, utilisez le bouton

Faites un click droit sur le job à déboguer, et cliquez sur "Déboguer"

Utilisateur

Travail : SDWRKBCH/SVDSVR/\*ALL Type : Etat : \*ACTIVE

Rang	Nom job	Utilisateur	Numéro travail	Etat	Etat actif	Fonction	Type fonction	File d'attente	Commande silverdev	Utilis
6	SDWRKBCH SVDSVR		091493	*ACTIVE	TIMW	SDWRKBCH	P	*DEV		
5	SDWRKBCH SVDSVR		091492	*ACTIVE	TIMW	SDWRKBCH	P	*DEV		
4	SDWRKBCH SVDSVR		091482	*ACTIVE	SELW	SDWRKBCH	P	*DEV	/Examples/sddmbks	ADU
3	SDWRKBCH SVDSVR		091359	*ACTIVE	SELW	SDWRKBCH	P	*DEV	*MYDESK	ADU
2	SDWRKBCH SVDSVR		091338	*ACTIVE	RUN	SDWRKBCH	P	PRTARNAUD ARNAUD	*DESIGNER	ADU
1	SDWRKBCH SVDSVR		021948	*ACTIVE	TIMW	SDWRKBCH	P	*DEV		

Context menu for job 4 (Rank 4):

- Déboguer
- Rafraichir F5
- Fichiers spoule
- Supprimer les fichiers spoule

Figure 44

Un onglet s'ouvre dans le designer. Son label correspond au numéro du job qui est débogué.  
Une fenêtre « attributs de debug » s'ouvre. Les options choisies peuvent être modifiées plus tard.

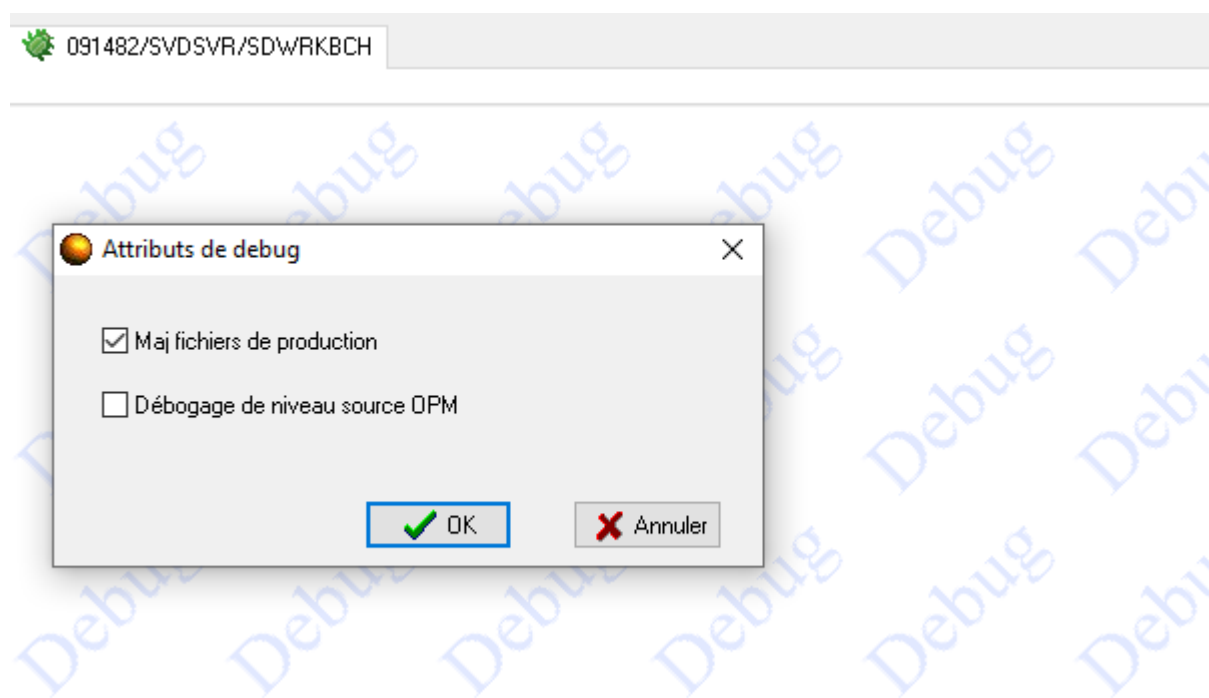
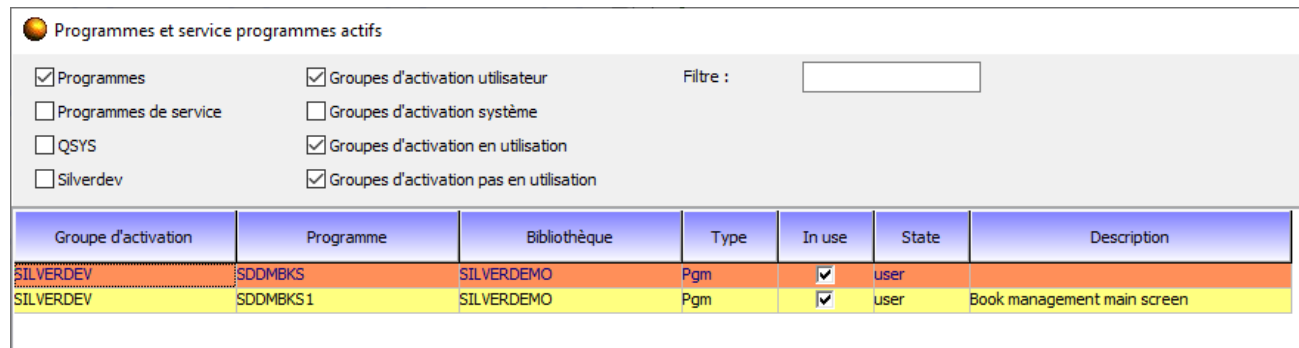


Figure 45

### Choix du programme

Une fenêtre avec la liste des programmes chargés en mémoire s'ouvre.

Si le programme à déboguer est en mémoire, double cliquez sur une ligne correspondant à ce programme.



Si le programme à déboguer n'est pas encore en mémoire, utilisez le bouton

Pgm/Srvpgm...

Figure 46

Vous arrivez alors à la fenêtre suivante :

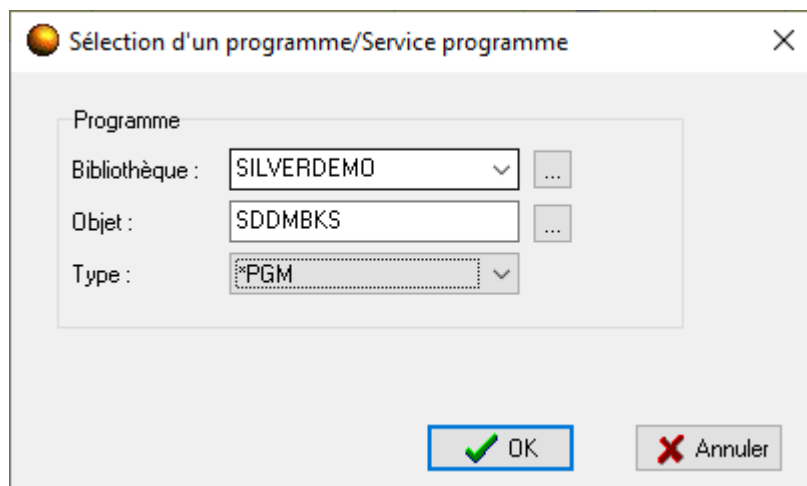


Figure 47

### Choix du module

Si le programme (ou service programme est composé de plusieurs modules) la liste des module est affichée :

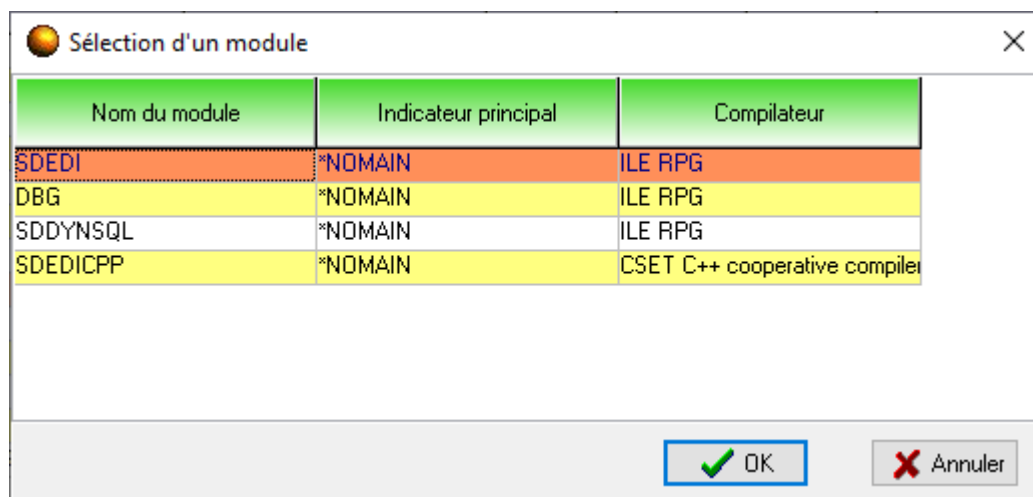


Figure 48

Dans notre exemple (silverdemo/sddmbks) il n'y a qu'un seul module, la fenêtre de selection d'un module n'est donc pas affichée.

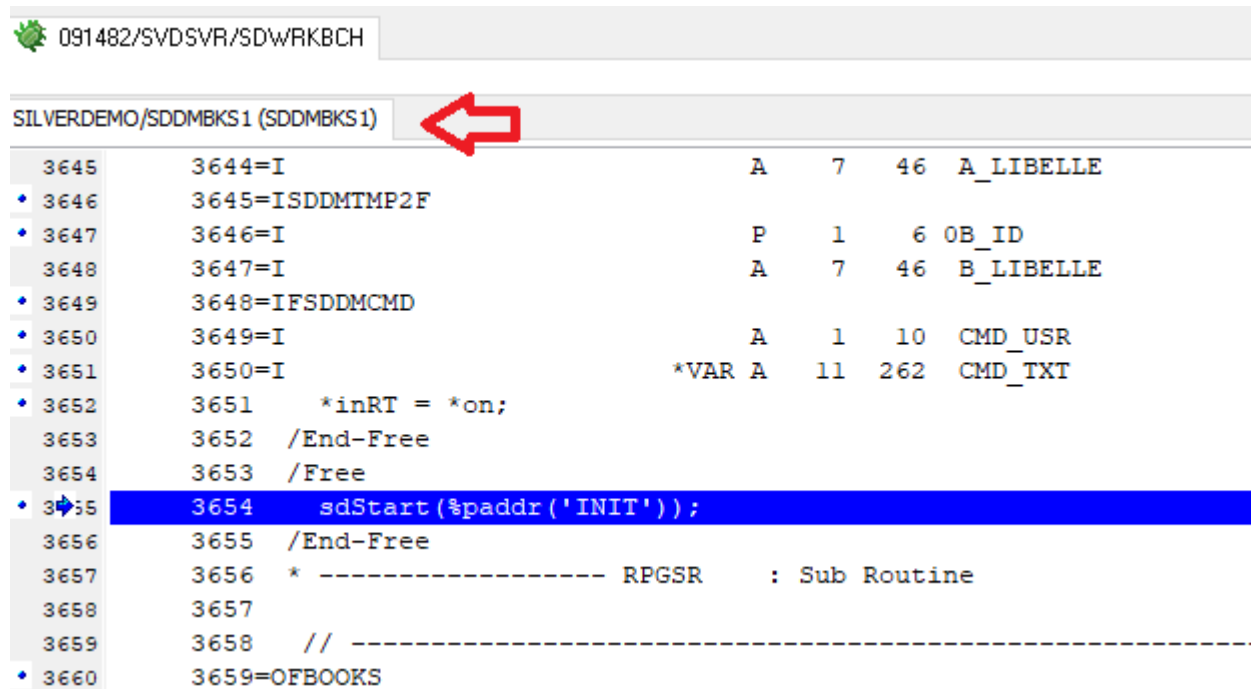
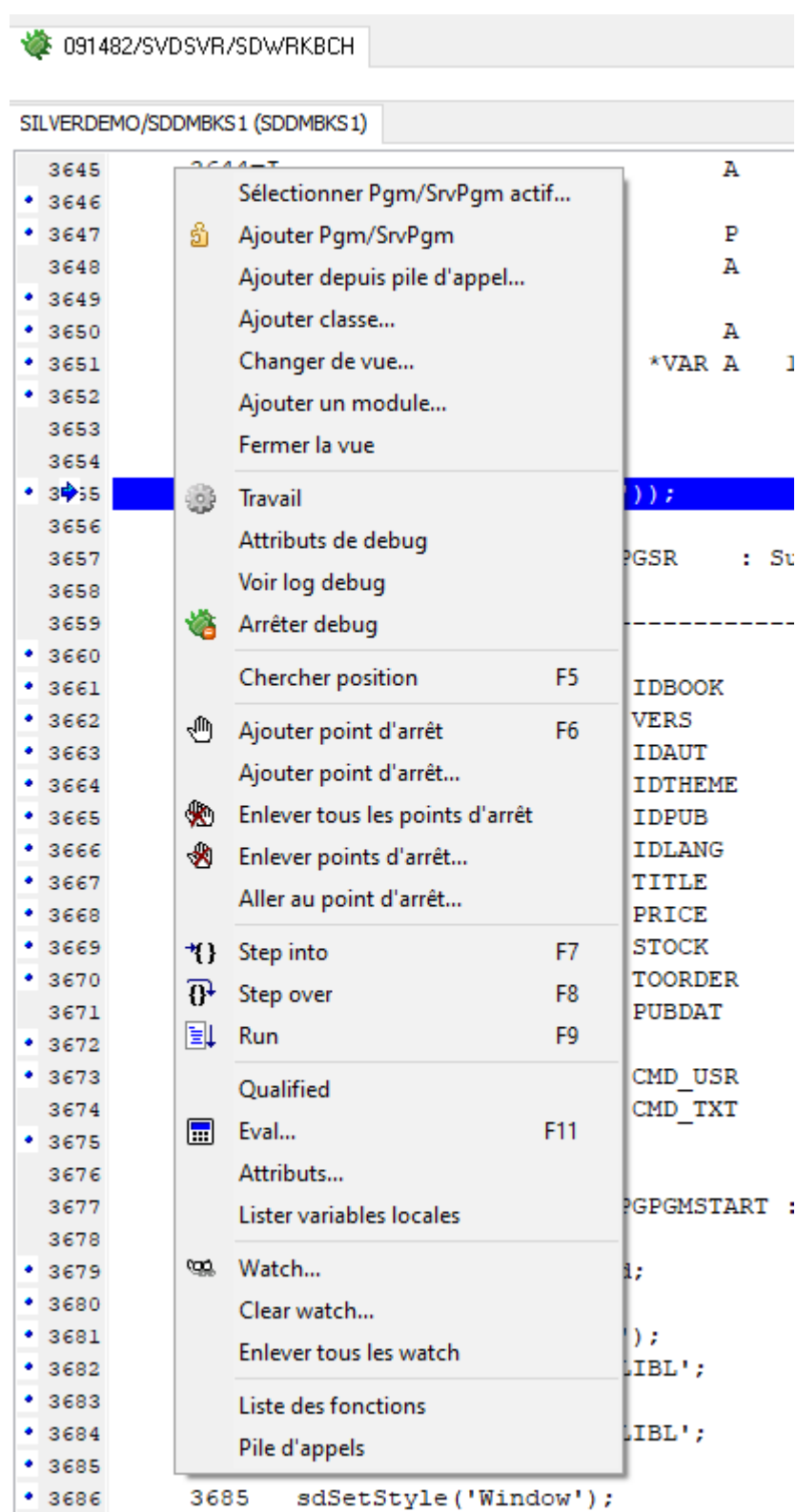


Figure 49

L'onglet précédemment ajouté contient à son tour un onglet. D'autres onglets s'ouvriront lorsqu'on ajoutera des modules.

### Menu contextuel

Effectuez un clic droit pour afficher un menu contextuel.



Description des options du menu contextuel :

Chercher position	Place le curseur à l'endroit où se trouve le programme. La ligne est
-------------------	--

	affichée en bleu
Ajouter point d'arrêt	Ajout direct d'un point d'arrêt à la ligne où se trouve le curseur.
Ajouter point d'arrêt...	Affiche un prompt pour ajouter un point d'arrêt conditionnel.
Enlever tous les points d'arrêt	Enlève tous les points d'arrêt.
Enlever points d'arrêts...	Affiche une grille avec tous les points d'arrêts. Sélectionnez une ou plusieurs lignes dans la grille et cliquez sur ok
Allez au point d'arrêt...	Affiche une grille avec tous les points d'arrêt. Sélectionnez une ligne et cliquez sur ok
Run	Permet de reprendre l'exécution du programme lorsqu'il est sur un point d'arrêt.
Step into	Effectue un pas à pas en entrant dans les fonctions.
Step over	Effectue un pas à pas sans entrer dans les fonctions.
Soumettre commande debug...	Affiche un prompt qui permet d'exécuter n'importe quelle instruction de debug. Le résultat de l'instruction est affichée dans une fenêtre outil. Voir chapitre Instructions de debug
Changer de vue...	Affiche la grille de sélection d'une vue comme lors du lancement du debug. Pour un module, une seule vue peut être affichée à la fois. Lors du changement de vue, la liste des points d'arrêt est conservée.
Changer de module...	Affiche la liste des modules du programme (ou service programme) dans la vue en cours. Il est possible d'afficher en même temps plusieurs modules d'un même programme (ou service programme) Pour afficher un module d'un autre programme, utilisez le menu principal Debug/Sélection programme
Qualified	Permet d'exécuter la commande QUAL pour la ligne où se trouve le curseur. L'instruction QUAL permet de déterminer le domaine de validité pour l'instruction eval.
EVAL	Permet d'afficher le contenu de la valeur de l'expression sous le curseur.
Attributs	Permet d'afficher le type d'une variable
Lister variables locales	Permet d'afficher toutes les variables locales
Watch	Utilisez Watch pour ajouter une variable à surveiller. Le programme s'arrête lorsqu'une variable à surveiller est modifiée.
Clear Watch...	Affiche une grille de toutes les variables de guêt et permet d'en supprimer
Enlever tous les watch...	Enlève toutes les variables de guêt.
Liste des fonctions	Affiche une fenêtre avec la liste des fonctions du module affiché. Double cliquez dans cette fenêtre pour déplacer le curseur au début de cette fonction.

*Remarque : lorsque vous sélectionnez une option de ce menu, la commande équivalente est affichée dans la zone de commandes de debug.*

### Zone de commandes de debug

En bas du texte, se trouve une zone permettant de lancer une commande de debug.

Pour voir les commandes de debug, voir chapitre Instructions de debug.

### Clicks et Doubles clicks

Pour gagner du temps, il est possible de cliquer dans la gouttière pour ajouter/supprimer un point d'arrêt.

Si vous double cliquez sur le texte, un eval de la zone sélectionnée sera effectuée.

Si vous cliquez sur un champ composé tel que Ds1.Field1, si vous cliquez sur Ds1, la zone Ds1 sera évaluée, si vous cliquez sur Field1, la zone Ds1.field1 sera évaluée.

### Log de débog

La fenêtre de log de débog affiche le résultat des commandes de débog.

Notamment le résultat des commandes EVAL :

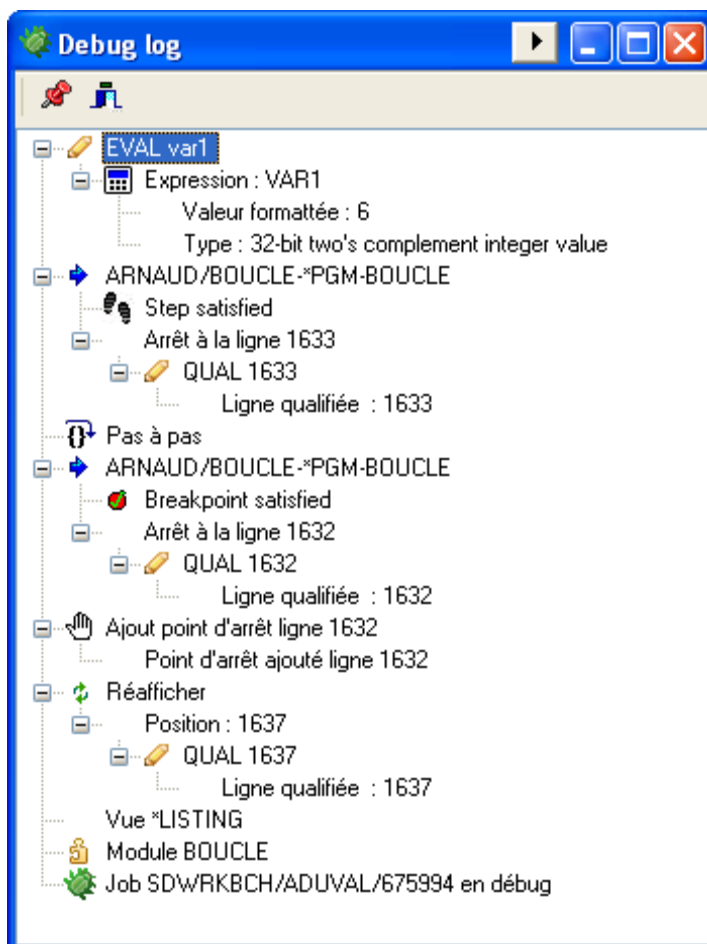


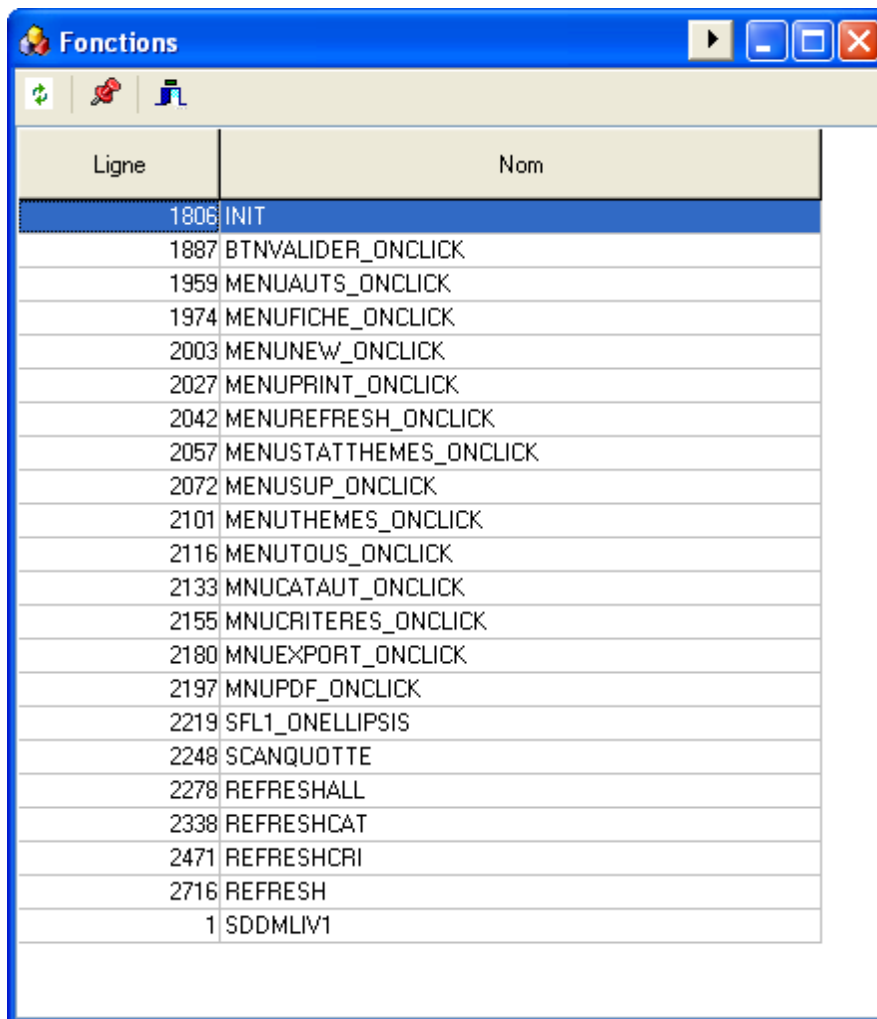
Figure 50

Cette fenêtre est automatiquement ouverte lors du lancement du débog.

### Liste des fonctions

Une fenêtre affiche la liste des fonctions.

Double cliquez sur une fonction pour déplacer le curseur au début de cette fonction dans la vue.



Ligne	Nom
1806	INIT
1887	BTNVALIDER_ONCLICK
1959	MENUAUTS_ONCLICK
1974	MENUFICHE_ONCLICK
2003	MENUNEW_ONCLICK
2027	MENUPRINT_ONCLICK
2042	MENUREFRESH_ONCLICK
2057	MENUSTATTHEMES_ONCLICK
2072	MENUSUP_ONCLICK
2101	MENUTHEMES_ONCLICK
2116	MENUTOUS_ONCLICK
2133	MNUCATAUT_ONCLICK
2155	MNUCRITERES_ONCLICK
2180	MNUEXPORT_ONCLICK
2197	MNUPDF_ONCLICK
2219	SFL1_ONELLIPSIS
2248	SCANQUOTE
2278	REFRESHALL
2338	REFRESHCAT
2471	REFRESHCRI
2716	REFRESH
1	SDDMLIV1

**Figure 51**

Cette fenêtre est automatiquement ouverte lors du lancement du debug.

## Ajouter un module

Pour ajouter un module, utilisez le menu Outils/debug/Ajouter module :



**Figure 52**



### Choix de la vue

Plusieurs vues peuvent être associées à une source, le débogueur choisit automatiquement la vue la plus complète. Il est possible plus tard de changer de vue.

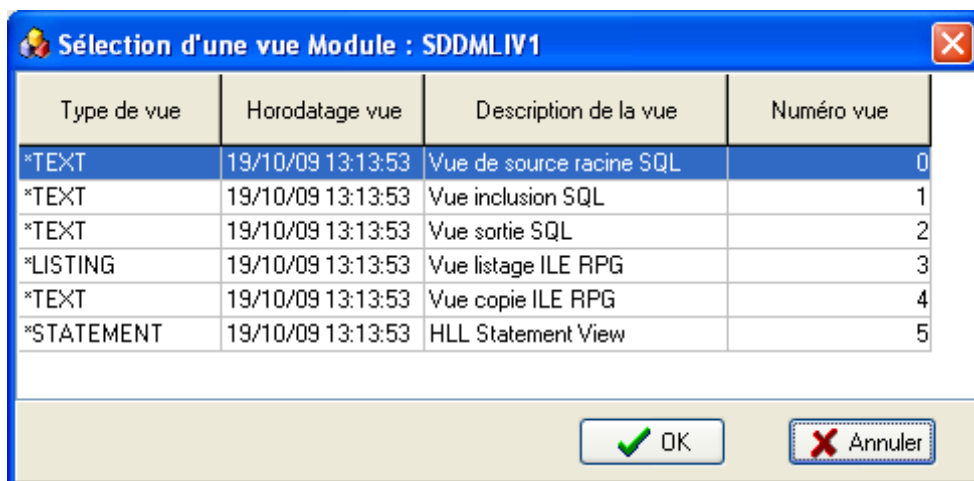


Figure 53

Le nombre de vues disponibles dépend de la manière dont le programme a été compilé. (option dbgview)

Par exemple, la vue \*Listing est disponible si le module a été compilé avec l'option DBGVIEW \*LIST ou DBGVIEW \*ALL

### Fonctionnalités dans une vue

Ici, nous avons choisi la vue \*LISTING.

Les points bleus dans la marge indiquent que la ligne est exécutable.

Utilisez la touche F5 pour retrouver l'emplacement où le programme s'est arrêté.

Cliquez dans la marge pour ajouter ou supprimer un point d'arrêt.

Pour plus d'options, utilisez le clic droit.

### Terminer le débog

Pour arrêter le débog, utilisez le menu principal "Debug/Terminer debug"

Toutes les fenêtres de débog sont automatiquement fermées sauf la fenêtre de log.

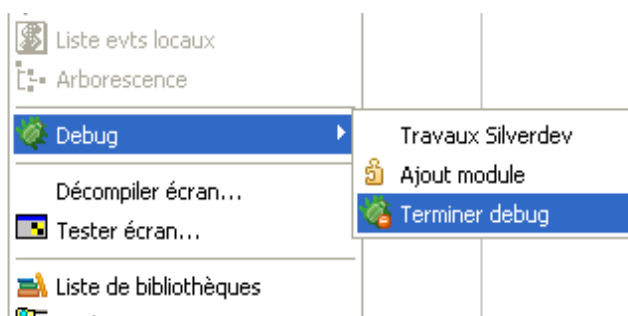


Figure 54

## Attributs de debug

Pour modifier à tout moment les attributs de débog, utilisez le menu Debug/Attributs de debug.

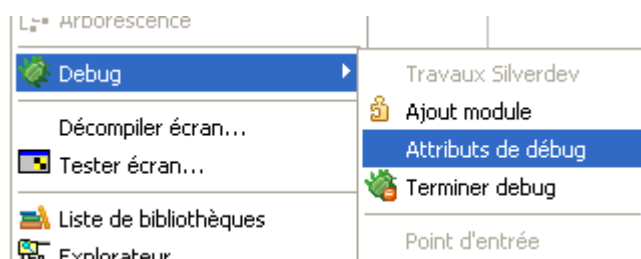


Figure 55

La fenêtre suivante permet de modifier les attributs de debug.

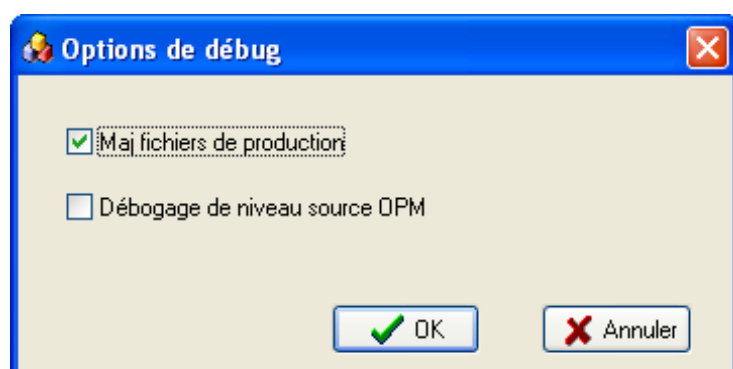


Figure 56

## Relancer un programme silverdev

Lorsqu'une chaîne de programme silverdev est en exécution, il est possible d'éviter de relancer tous les programmes depuis le début lorsque vous modifiez un programme.

Le programme que vous souhaitez relancer doit être dans un groupe d'activation nommé et de préférence ne contenant que ce programme.

Dans le débogueur, dans la fenêtre de sélection d'un programme actif, effectuez un click droit, et choisissez "Réclamer le groupe d'activation"

Attention, si des fenêtr correspondant à ce programme sont encore visible, il ne faut plus les utiliser.

Vous ne pouvez pas réclamer le groupe d'activation Silverdev, et vous ne pouvez pas réclamer un groupe d'activation si il contient un programme qui est dans la pile d'appel. (Premier programme ou programme avec fiche modale).

## Point d'entrée

Pour débbuger un job qui n'est pas encore démarré, utilisez le menu Debug/Point d'entrée.

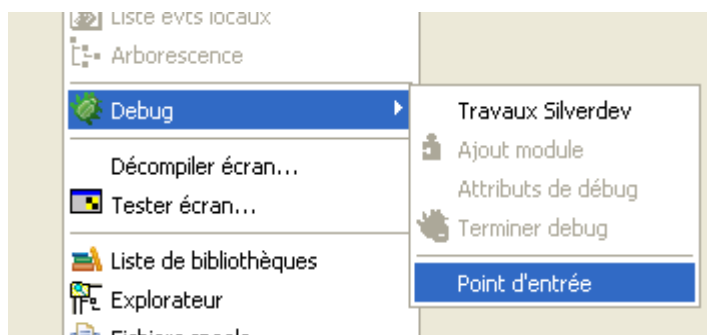


Figure 57

Les fenêtres de choix de programme, de module et de vue s'affichent.

Lorsque vous cliquez dans la gouttière, ce n'est pas une instructino break qui est exécutée, mais une instruction sbreak.

La fenêtre de log du debug affiche :

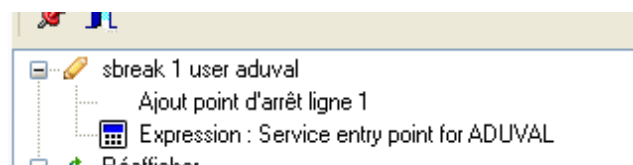


Figure 58

Lorsque l'utilisateur saisi dans la commande SBREAK lance le programme en debug, une fenêtr apparaît.

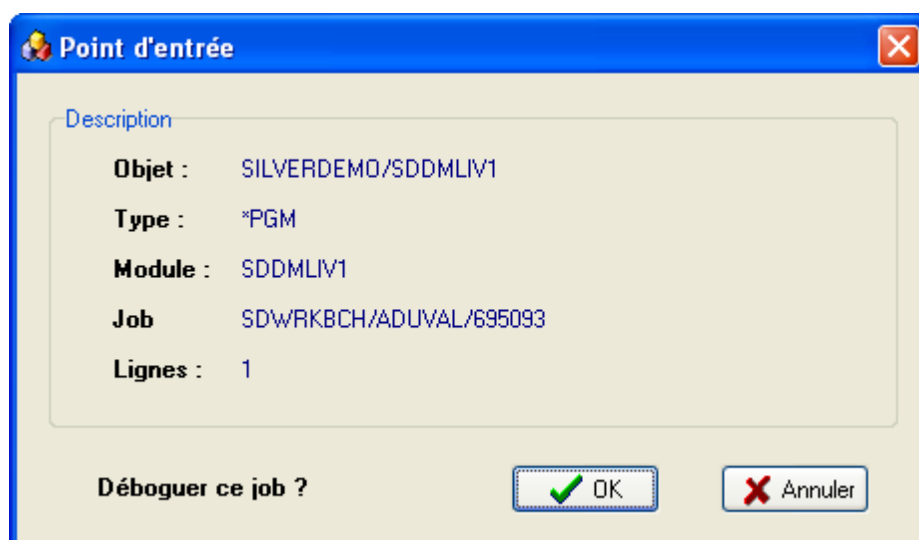


Figure 59

Vous avez une minute pour cliquer sur le bouton ok.

Si vous cliquez sur le bouton ok, le nouveau job est en debug.

Instructions de debug).

Lorsque vous sélectionnez une option dans le menu déroulant ou double cliquez, la commande associée s'affiche dans cette zone.

Vous pouvez passer de la zone de texte à la zone de commande de debug avec la touche Tabulation.

## Debug en 5250

Repérez le job à déboguer avec la commande wrksdjob :

```

Work with SilverDev jobs
27/02/23 09:04:04

SilverDev server is running

2=Change 3=Hold 4=End 5=Work with 6=Release 8=Work with spooled files
10=Job log 11=Call stack 12=Job locks 13=Library list...

Current
Opt Job user prf Stt Function
-- 092614/SVDSVR/SDWRKBCH ADUVAL SELW *SilverDev-MyDesk
-- 092619/SVDSVR/SDWRKBCH ADUVAL SELW /Examples/sddmbks

```

Saisissez l'option 18 devant le job à déboguer. Cela est équivalent à la commande STRSRVJOB

Utilisez ensuite la commande STRDBG :

```

                                Start Debug (STRDBG)

Type choices, press Enter.

Program . . . . . sddmbks1 Name, *NONE
Library . . . . . silverdemo Name, *LIBL, *CURLIB
                  + for more values

Default program . . . . . *PGM Name, *PGM, *NONE
Maximum trace statements . . . . . 200 Number
Trace full . . . . . *STOPTRC *STOPTRC, *WRAP
Update production files . . . . . *yes *NO, *YES
OPM source level debug . . . . . *NO *NO, *YES
Service program . . . . . *NONE Name, *NONE
Library . . . . . Name, *LIBL, *CURLIB
                  + for more values

```

Saisissez le nom du programme à déboguer. Notez que l'on retrouve l'option « update production files »

Vous êtes ensuite dans le débogueur 5250.

```

                                Display Module Source

Program:  SDDMBKS1      Library:  SILVERDEMO      Module:  SDDMBKS1
3658      3657      sdStart(%paddr('INIT')) ;
3659      3658      /End-Free
3660      3659      * ----- RPGSR      : Sub Routine
3661      3660
3662      3661      // -----
3663      3662=0FB00KS
3664      3663=0                                IDBOOK                6P PACK      10,
3665      3664=0                                VERS                    17P PACK      20,
3666      3665=0                                IDAUT                     23P PACK      10,
3667      3666=0                                IDTHEME                   29P PACK      10,
3668      3667=0                                IDPUB                     35P PACK      10,
3669      3668=0                                IDLANG                    40A CHAR
3670      3669=0                                TITLE                     140A CHAR      10
3671      3670=0                                PRICE                     143P PACK      5,
3672      3671=0                                STOCK                     146P PACK      5,

More . . .

Debug . . .

```

Lorsque vous avez terminé le debug, utilisez les commandes ENDDBG et ENDSRVDBG.

## Point d'entrée

Si vous voulez déboguer un job avant que l'application soit lancée, commencez par la commande

```
STRDBG PGM(SILVERDEMO/SDDMBKS1)
```

Ajoutez un point d'arrêt avec la commande SBREAK:

```
SBREAK 1 user myuser
```

Quand le programme est démarré par l'utilisateur identifié dans la commande SBREAK, une fenêtre est affichée :

### Display Program Messages

```
Subsystem library list entry QSYS2924 in library list.
Job 327117/ADUVAL/QPADEV0009 started on 23/04/10 at 14:27:28 in subsystem QB
Service Entry Point has stopped at line 193 in program SILVERDEMO/SDDMBKS1
```

Vous pouvez utiliser la touche F1 pour plus de détails :

### Additional Message Information

```
Message ID . . . . . : CPI1903      Severity . . . . . : 10
Message type . . . . . : Notify
Date sent . . . . . : 23/04/10      Time sent . . . . . : 14:40:23

Message . . . . . : Service Entry Point has stopped at line 192 in program
SILVERDEMO/SDDMBKS1 in job 327008/ADUVAL/SDWRKBCH.
Cause . . . . . : Service Entry Point has stopped at line 192 in program
SILVERDEMO/SDDMBKS1 in job 327008/ADUVAL/SDWRKBCH.
Recovery . . . . . : This program must be debugged from a servicing job. Do a
Start Service Job (STRSRVJOB JOB(327008/ADUVAL/SDWRKBCH)). Then do Start
Debug (STRDBG) on the spawned job from where the STRSRVJOB was done. Set a
local breakpoint at or after the Service Entry Point. Return to the original
job and press enter to release the spawned job.
```

Dans une autre session 5250, utilisez la commande :

```
STRSRVJOB 680493/QSECOFR/SDWRKBCH
```

Et

```
STRDBG PGM(SILVERDEMO/SDDMBKS1)
```

Remarque :

La commande **SBREAK** a été introduite dans la version **V5R2M0**

## Instructions de debug

Les instructions suivantes sont accessibles dans le débogueur du designer ou dans le débogueur 5250.

### ATTR

→ ATTR → variable →

L'instruction ATTR permet d'afficher les attributs d'une variable.

Le dernier appel à l'instruction QUAL détermine la localisation de la variable.

### Break

→ BREAK → position →  
 → BREAK → position └ WHEN → expression ┘ →

L'instruction break permet d'ajouter un point d'arrêt.

Le paramètre de condition est optionnel.

Un point d'arrêt peut être remplacé par un autre. Le point d'arrêt le plus récent à une position est le point d'arrêt actif.

### Sbreak

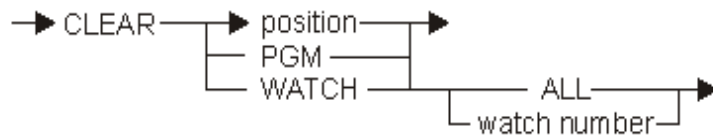
→ SBREAK → position └ USER → userid ┘ →

L'instruction sbreak permet d'entrer un point d'arrêt d'entrée de service.

UserId détermine le profil sous lequel le job doit s'exécuter pour activer le point d'entrée. Si UserId n'est pas défini, le profil du job est utilisé.

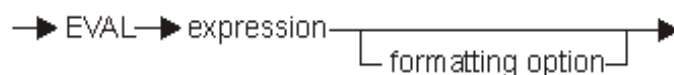
Un service breakpoint et un breakpoint ne peuvent pas exister en même temps à la même position.

### Clear



L'instruction Clear permet de supprimer un point d'arrêt ou une variable de guêt.

### EVAL



L'instruction eval permet d'afficher le contenu d'une valeur.

Les variables peuvent être affichées lorsque le programme est suspendu. Le programme est suspendu lorsqu'il rencontre un point d'arrêt, qu'il exécute un pas à pas, ou qu'une variable de guêt est modifiée.

Le format affiché dépend du paramètre formatting option

:c	Chaine de caractères ebcdic
:x	Format hexadecimal. Pour un pointeur, c'est la valeur déréférencée qui est affichée.
:s	Chaine de caractères (seulement pour modules compilés en langage C ou C++)
:f	Chaine de caractères (seulement pour modules compilés en langage C ou C++)
:a	Chaine de caractère ascii
:u	Format unicode

Le paramètre formatting option peut être suivie d'une longueur pour indiquer le nombre d'octets à afficher. Si ce nombre est omis, les valeurs par défaut sont les suivantes :



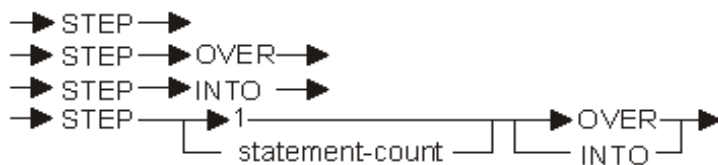
:c	1
:x	Longueur de la variable
:s	30
:f	1024
:a	1024
:u	1024

## QUAL

→ QUAL → position →

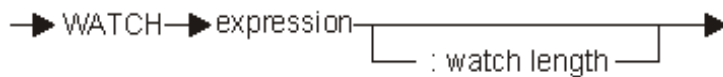
Cette instruction détermine la localité pour les instructions évaluées qui suivent.

## Step Statement



L'instruction Step permet d'avancer en pas à pas dans le programme.

## Watch

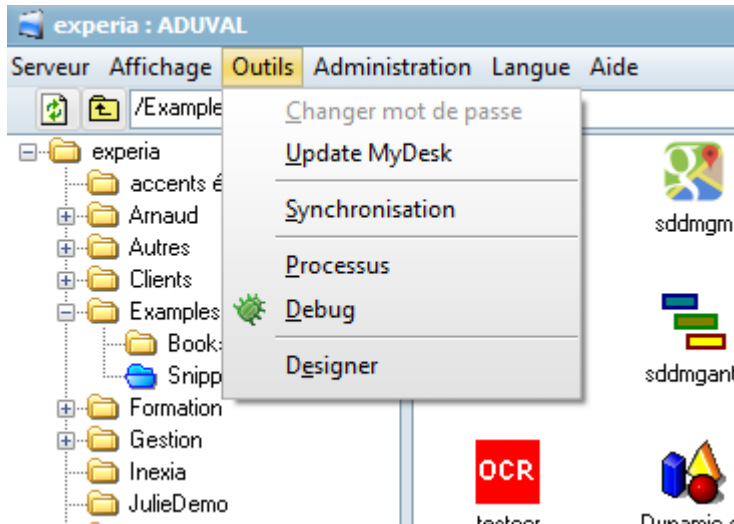


L'instruction Watch permet d'ajouter une variable de guet.

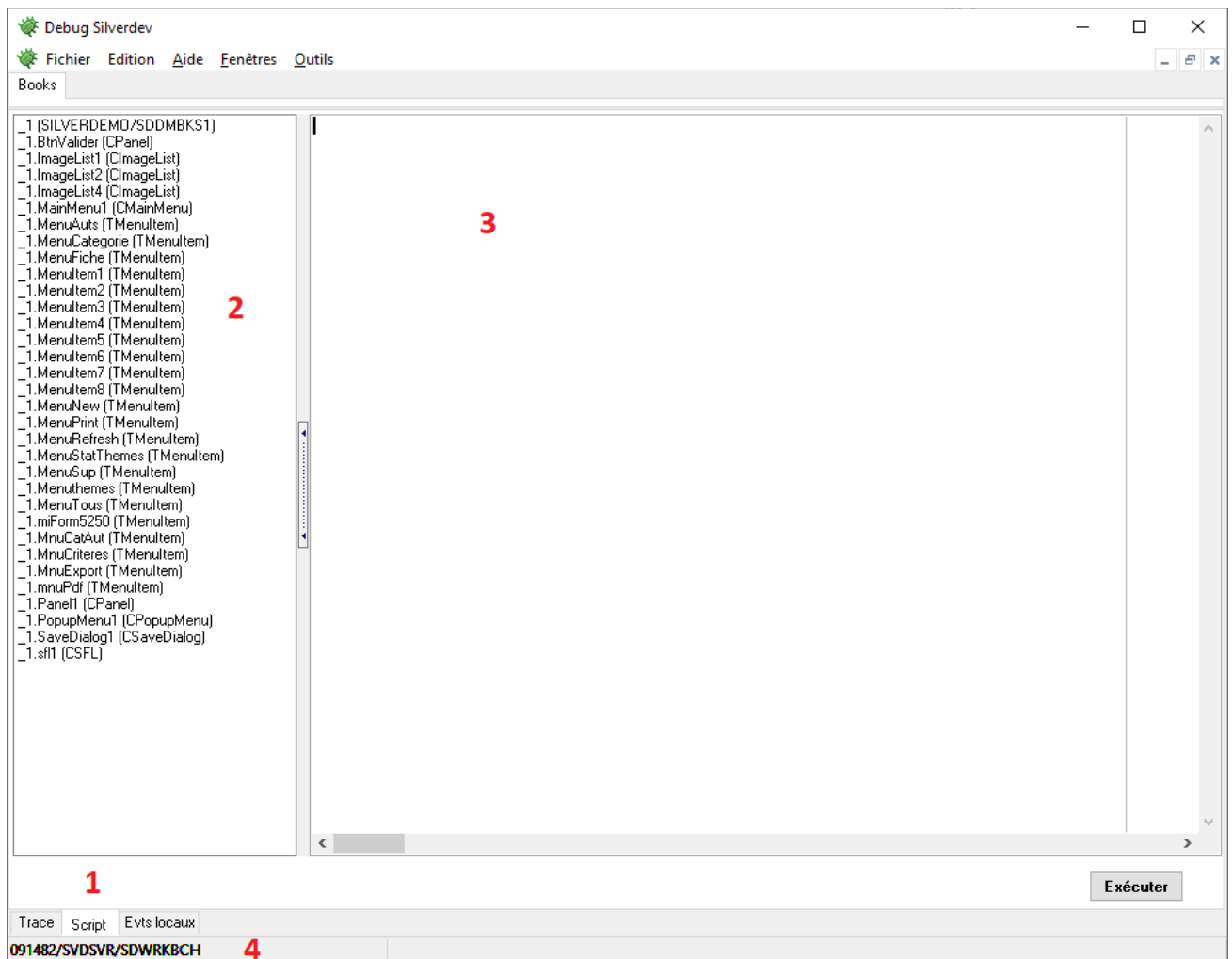
Lorsqu'une variable de guet est modifiée, le programme est suspendu.

## Chapitre 10. Debug côté client

Vous pouvez lancer l'application de debug depuis MyDesk :



Une fenêtre contenant des onglets s'ouvre. Pour chaque application silverdev qui s'exécute, un onglet est ouvert



En (4) on retrouve le numéro du job associé

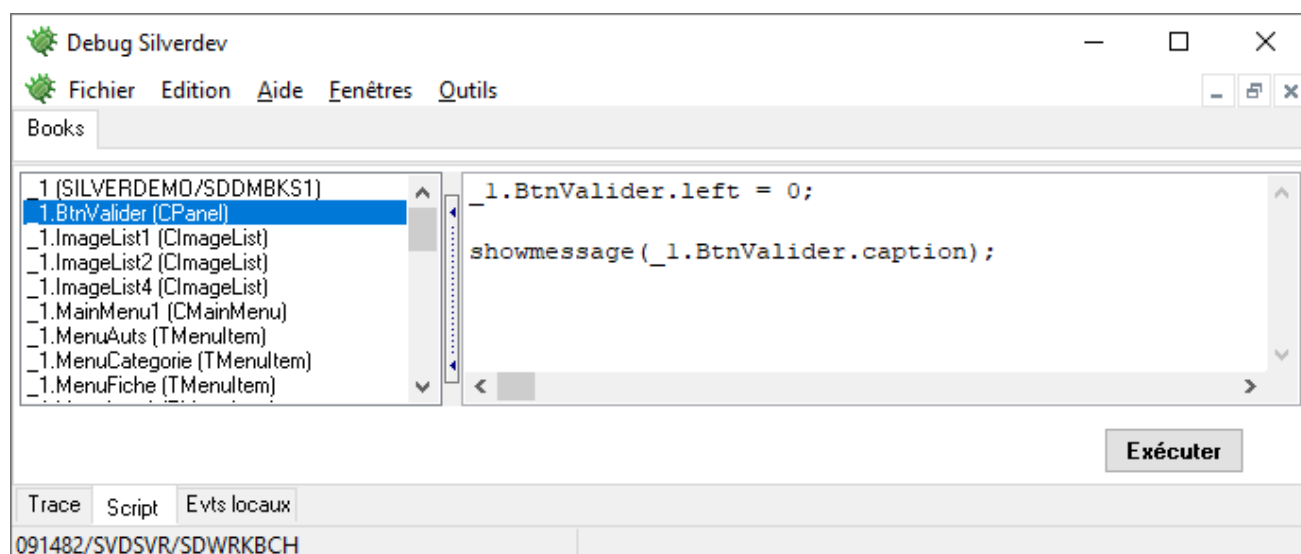
## Onglet Script

On retrouve en bas trois onglets (1) L'onglet Script est ouvert par défaut.

Dans la zone (2) la liste des fiches et de leurs composants est affichée.

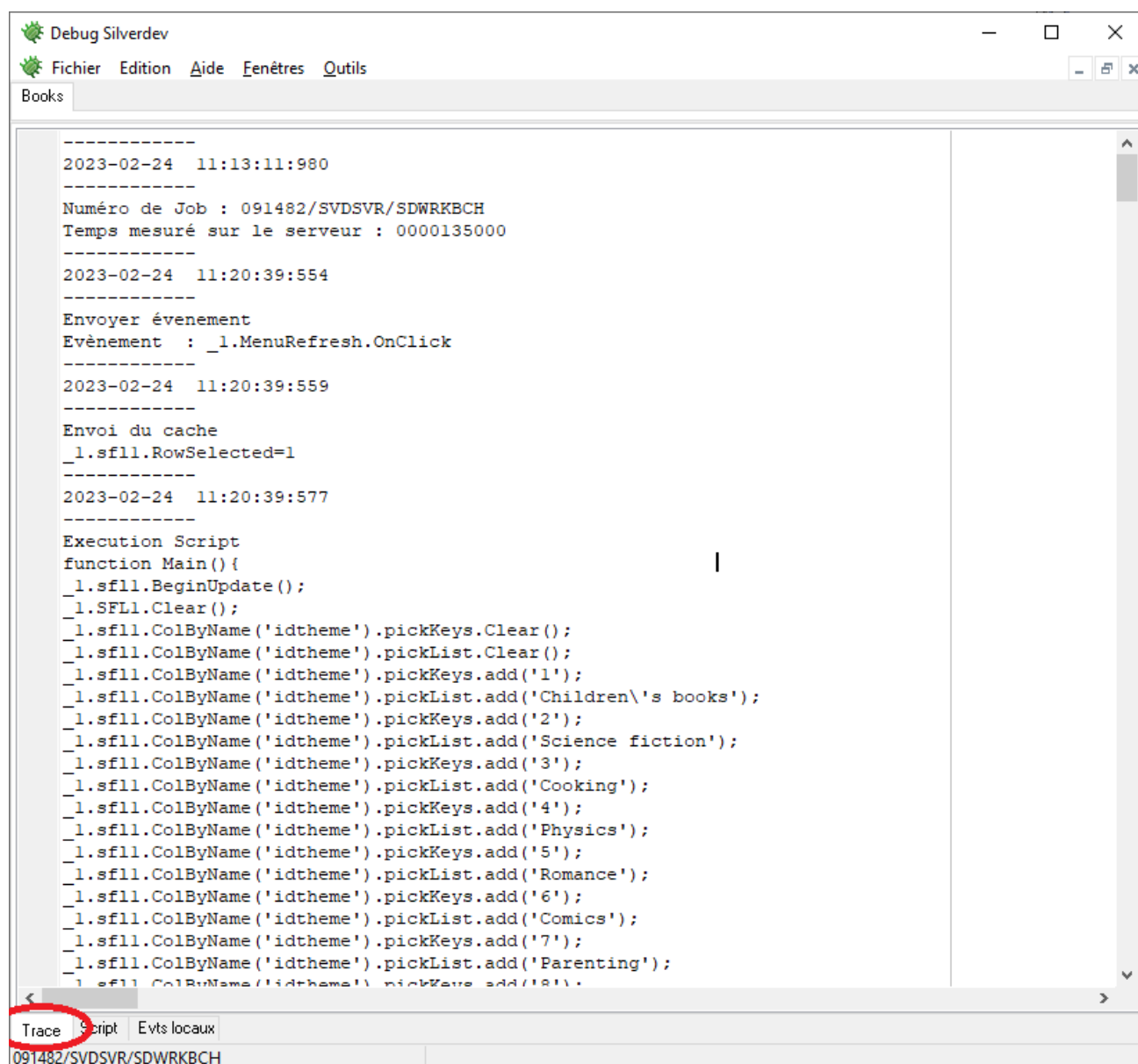
Dans la zone (3) il est possible d'écrire du code pour effectuer des tests :

(Vous pouvez double cliquer sur la partie (1) pour ajouter le nom d'un composant dans la partie (3))



## Onglet Trace :

L'onglet trace permet de voir tout ce qui a été envoyé depuis le serveur

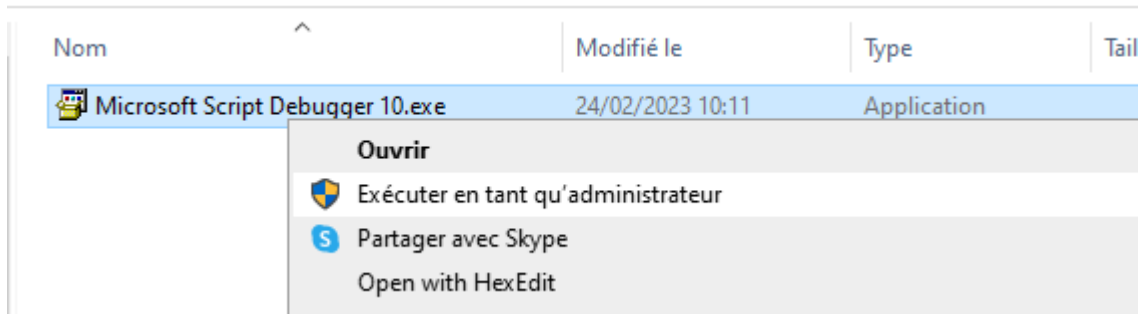


## Microsoft Script Debugger

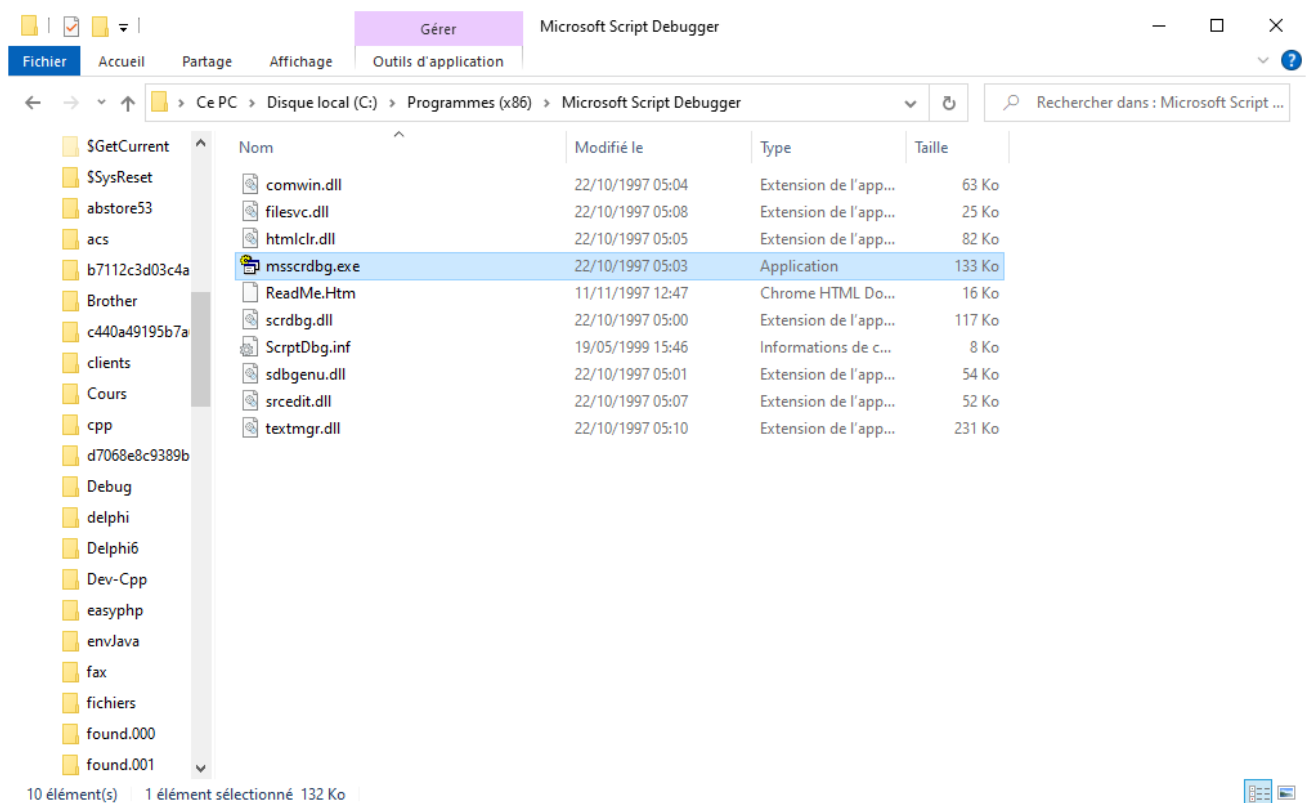
Dans le dossier de la release, il y a un répertoire nommé tools.

Il permet d'installer le microsoft Script Debugger.

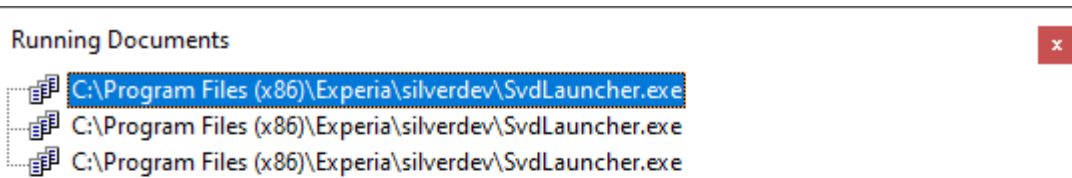
Lancez le programme d'installation en étant administrateur sur votre poste.



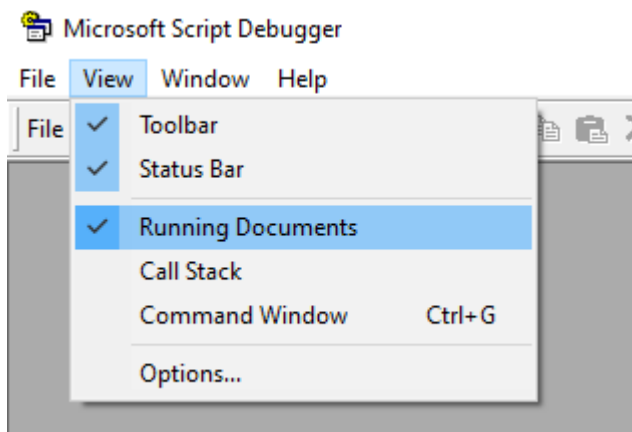
Ensuite lancez le programme installé



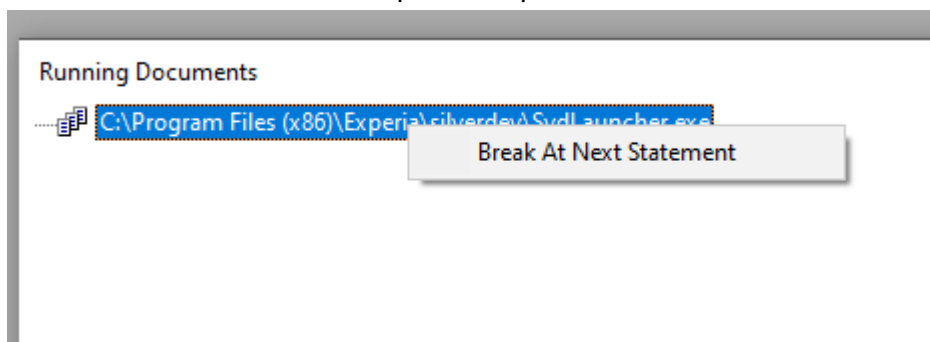
Le programme détecte toutes les instances de SvdLauncher :



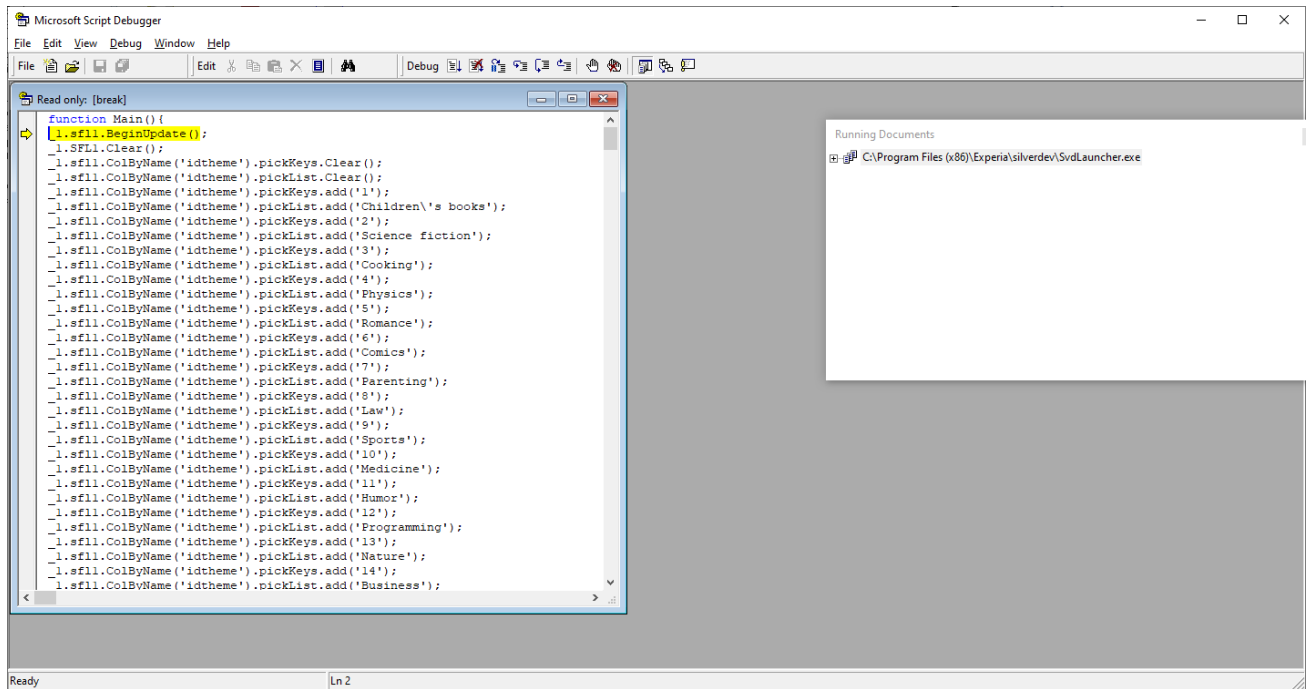
Remarque : si la fenêtrés ci-dessus n'est pas visible, cochez l'option Running documents :



Effectuez un click droit sur le process qui vous intéresse :



Déclenchez un évènement, le code envoyé depuis le serveur se trouve en debug :



Vous pouvez faire du pas à pas avec la touche F8.

A noter que les évènements locaux ne peuvent pas être déboguer de cette manière.



# Chapitre 11. Erreurs

## Erreurs côté serveur

Lorsqu'une erreur non monitorée survient dans votre programme, le programme se termine.  
Le job associé est en MSGW.

L'utilisateur est prévenu par une boîte de dialogue.

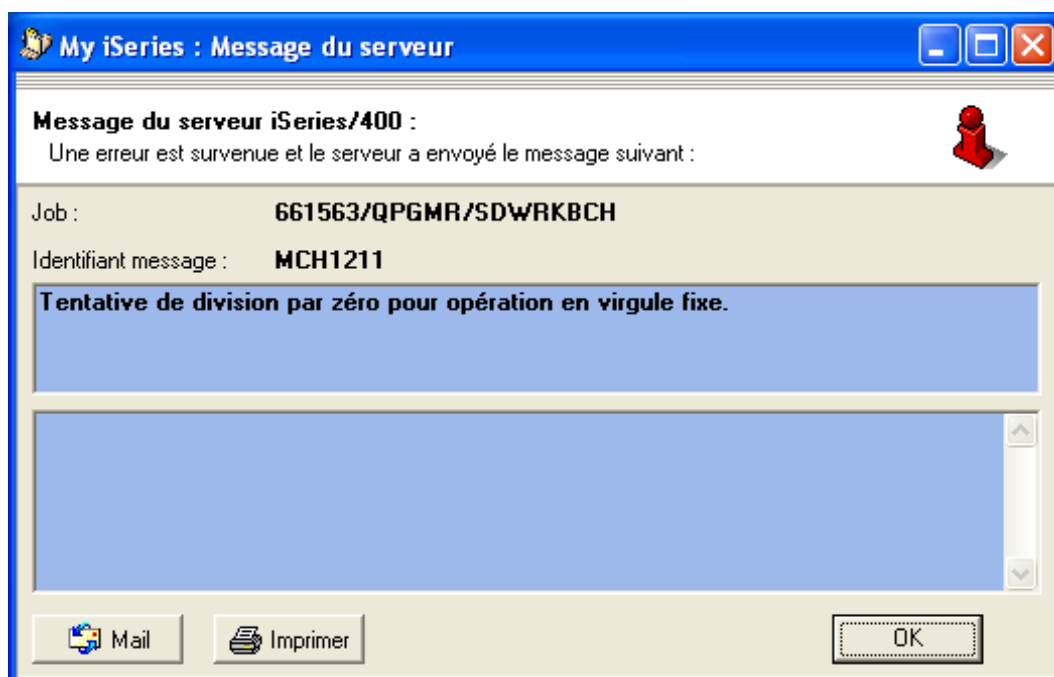


Figure 60

Le bouton mail permet d'envoyer le rapport à une adresse qui est stockée dans le champ CFMAILADM.

## Erreurs côté client

Si une erreur se produit côté client, un message SVD0716 est envoyé au programme.

```

Travail 754601/QSECOFR/SDWRKBCH démarré le 13/10/07 à 14:23:35 dans le
sous-système SILVERDEV de SILVERDEV ; soumis le 13/10/07 à 14:23:35.
-* Connexion depuis l'adresse ip : 192.168.0.169
La bibliothèque SILVEDEV existe déjà dans la liste des bibliothèques.
-* Utilisateur en cours : ADUVAL
-* Fonction ou commande demandée : call arnaud/screen1
Rapport d'erreur Instruction : _1.Button1.widthz=10; Description : Cet
objet ne gère pas cette propriété ou cette méthode
Erreur d'exécution de script
SilverDev s'est arrêté en erreur
Erreur. SVD0112 non intercepté par SCREEN1 à la spécif 0000001377, inst
x'0000'.
(C G D F) Appel à SDSTART terminé en erreur.

```

Pour que le message SVD0716 ne soit pas envoyé au programme, utilisez la fonction `sdSetClientErrorHandler` dont le prototype est le suivant :

```

d sdSetClientErrorHandler...
d                                pr
d  proc                          *   procptr value
D  Type                          10   value options(*nopass)

```

Le premier paramètre est un pointeur vers une procédure qui sera appelée en cas d'erreur sur le client.

Utilisez le second paramètre pour déterminer les erreurs interceptées.

'\*LOCAL' pour les erreurs dans un événement local

'\*SERVER' pour les erreurs dans un événement serveur.

'\*ALL' pour toutes les erreurs clients.

La valeur par défaut du second paramètre est '\*ALL'

La fonction gestionnaire d'erreur doit avoir le prototype suivant :

```

d ClientErrorHandler...
d                                pr
d msg                          500   varying value

```

*Remarque :*

*Pour les événements serveur il est aussi possible d'intercepter l'erreur comme sur l'exemple suivant :*

```

c                                monitor
c                                callp   sdSetString(f1:'sf11':'toto':'')

```

c	callp	sdApplySet
c	on-error	
c	endmon	

### Debugger just in time :

Si un débogueur just in time javascript est installé sur le poste vous aurez la fenêtre suivante qui s'affichera, cliquez sur non pour obtenir la boîte de dialogue précédente.

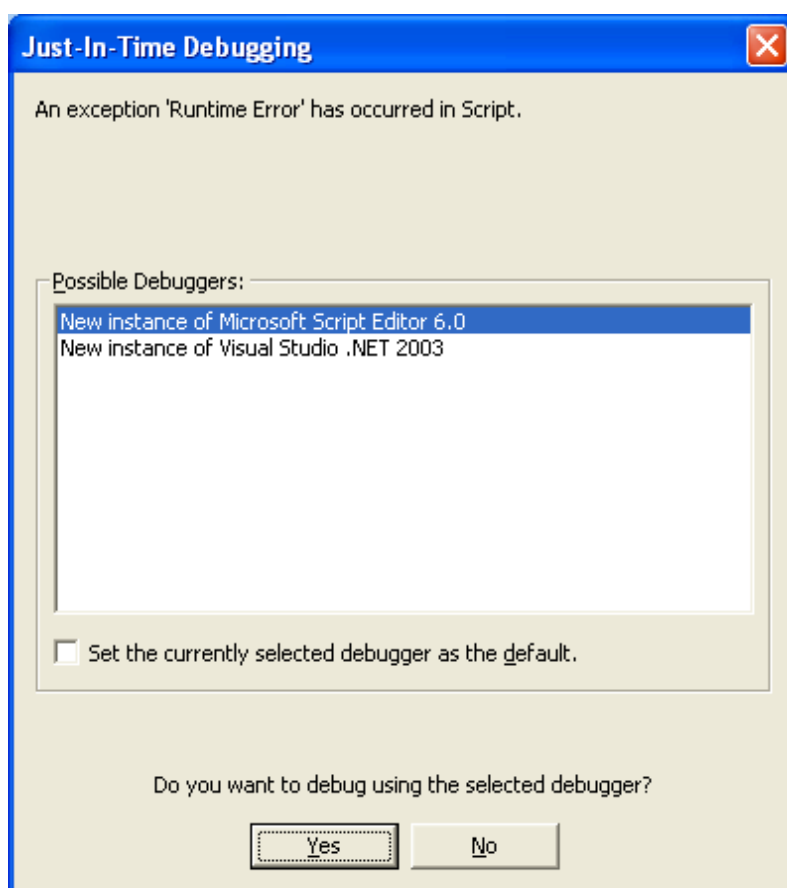


Figure 61

---

## Chapitre 12. JOB programmes en MSGW

Lorsqu'un programme est en erreur, le job est en MSGW. Certains fichiers peuvent alors être lockés.

### Première solution :

Modifiez la jobd silverdev/silverdev :

```
CHGJOB JOB(SILVERDEV/SILVERDEV) INQMSGRPY(*DFT)
```

### Seconde solution :

Lorsqu'un programme silverdev est en msgw, l'exit program associé à l'exit point SVDEXCEPT est appelé.

Vous pouvez associer un exit program avec la commande addexitpgm.

```
ADDEXITPGM EXITPNT(SVDEXCEPT) FORMAT(EXCEPT01) PGMNBR(*LOW) PGM(MYLIB/MYEXITPGM)
```

L'exit program doit avoir la liste de paramètres suivante :

C	*entry	plist		
C		parm	jobname	10
C		parm	jobUser	10
C		parm	jobNbr	6
c		parm	stop	1

Le paramètre Stop peut être modifié avec la valeur 'Y', ce qui arrêtera le job.

### Troisième solution :

C	*pssr	begsr
C		return
C		endsr

## Chapitre 13. Sources écrans

Les sources des écrans sont sauvegardés sur l'ifs. Il est possible d'ouvrir ces sources à l'aide d'un éditeur de texte. Cette possibilité est très utile dans certains cas.

- Vous souhaitez voir les propriétés qui ont été modifiées. Seules les propriétés différentes des valeurs par défaut sont enregistrées dans le fichier.
- Vous souhaitez modifier une propriété sur plusieurs écrans. Il vous est alors possible d'effectuer une recherche avec un éditeur classique pour la modifier directement dans le source. (Dans ce cas, pensez à compiler l'écran avec le designer)
- Dans un cas extrême, il est possible de remplacer un composant par un autre, par exemple un CButton par un CBitBtn. Dans ce cas attention aux propriétés de l'ancien composant que ne possède pas le nouveau. (voir Remarque)

Remarque :

*Faites très attention à ce que vous faites en modifiant le source à la main.*

*Une erreur de syntaxe pourrait empêcher l'ouverture de la fiche dans le designer.*

Par exemple si vous faites une erreur sur un type de composant :

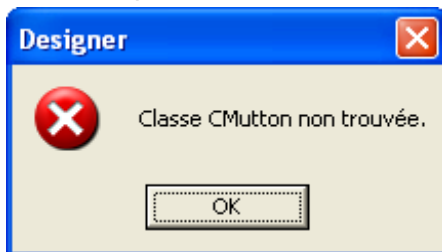


Figure 62

ou sur un nom de propriété :

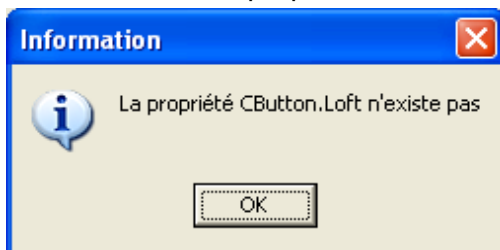


Figure 63

Dans le second cas, la fiche pourra parfois être ouverte dans le designer, mais la propriété mal orthographiée prendra la valeur par défaut. (0 pour un entier, false pour un boolean et une chaîne vide pour un string)

**Exemple de source :**

```
object Win: TWin
  Left = 354
  Top = 291
  Width = 809
  Height = 400
  Color = clBtnFace
  Font.Charset = DEFAULT_CHARSET
  Font.Color = clWindowText
  Font.Height = -11
  Font.Name = 'MS Sans Serif'
  Font.Style = []
  OldCreateOrder = False
  Events.Strings = (
    'Button1.OnClick')
  PixelsPerInch = 96
  TextHeight = 13
  object Button1: CButton
    Left = 40
    Top = 80
    Width = 75
    Height = 25
    Caption = 'Button1'
    TabOrder = 0
  end
  object Edit1: CEdit
    Left = 32
    Top = 32
    Width = 121
    Height = 21
    Color = clOlive
    TabOrder = 1
    Text = 'Edit1'
  end
end
```

---

# Chapitre 14. Impressions

## Impression simple

Dans ce chapitre, nous distinguerons les impressions simples des impressions de type liste. Dans une impression simple, chaque bande est imprimée une seule fois, dans les impressions de type liste, certaines bandes sont imprimées plusieurs fois.

Les composants de l'onglet impressions permettent de créer des impressions ainsi que des aperçus avant impression.

La maquette d'une impression sera faite sur une fiche. La fiche devra être non visible, (l'utilisateur n'a pas besoin de voir la maquette) cependant, elle devra être créée à l'aide de la fonction `sdCreateForm`.

Sur cette fiche, nous déposerons un composant `CReport` et des composants `CRepBand`. Sur les composants `CRepBand`, nous déposerons des composants de type `CRepLabel`, `CRepImage`, `CRepCheckbox`, `CRepShape`, `CRepSysField`, `CRepChart` et `CrepMemo`.

Les instructions `sdPrint` et `sdPreview` permettant de lancer une impression feront référence au composant `CReport`.

## Aperçu avant impression

Pour avoir un aperçu avant impression, utilisez la fonction `sdPreview`.

Une fenêtre affiche alors l'aperçu. Cette fenêtre permet à l'utilisateur de naviguer à travers les différentes pages, d'effectuer des zooms, ainsi que de choisir les pages à imprimer.

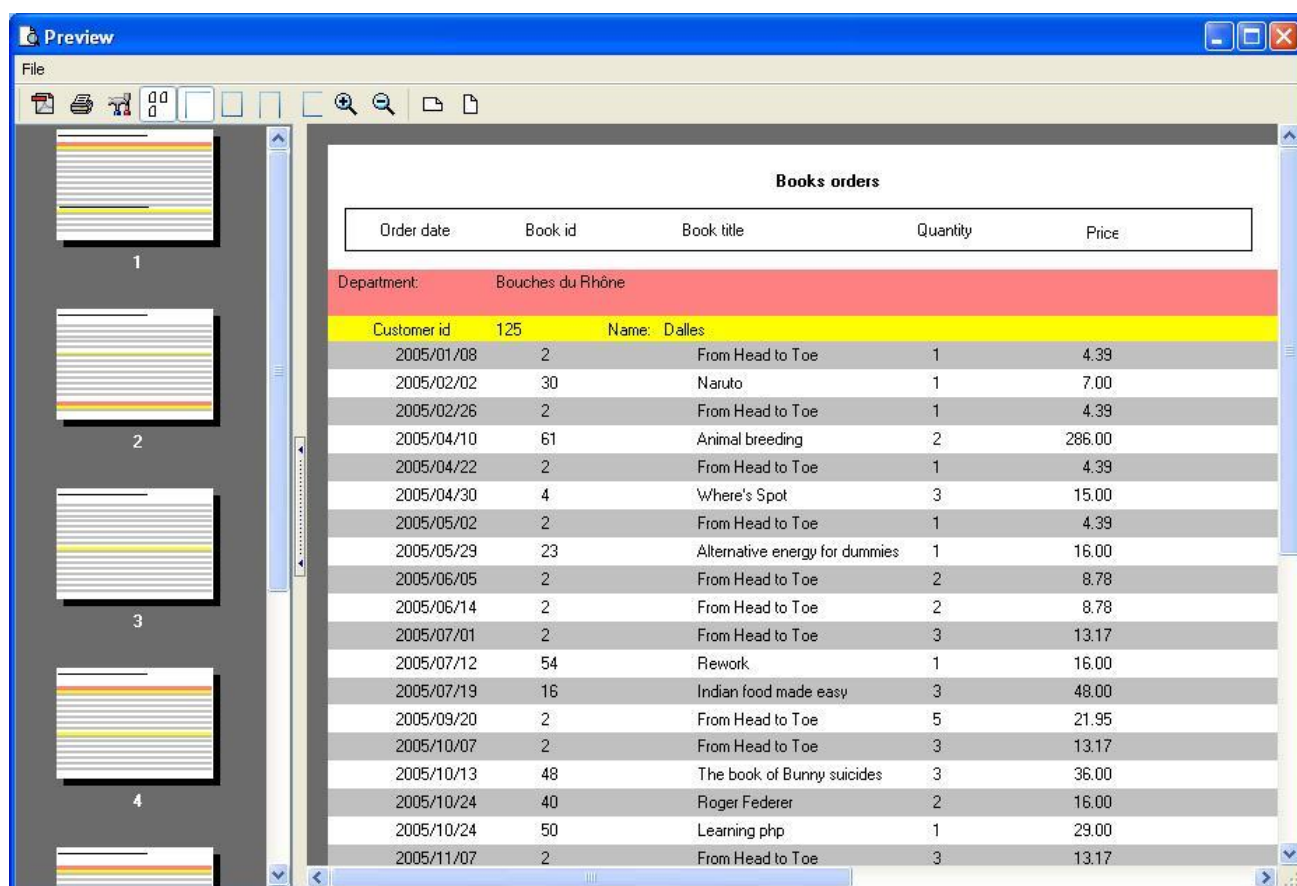


Figure 64

## Orientation

Le composant CReport a une propriété Orientation qui peut prendre 3 valeurs :

PoLandscape	Impression Paysage
PoPortrait	Impression portrait
PoUser	Valeur choisie par l'utilisateur dans la boîte de dialogue d'impression (sdPrintDlg)

## Imprimer une liste

Si vous souhaitez imprimer une liste dont le nombre d'éléments est inconnu à la conception, modifiez la propriété BandType du composant CRepBand en alr\_Detail.

Pour les composants CRepLabel, CRepImage, CRepCheckBox, CRepChart et CRepMemo, il est possible d'ajouter une liste de valeurs avec les fonctions sdAddValue, sdAddImg, sdAddLines, sdAddChart. Pour chaque valeur ajoutée avec ces fonctions, la bande sera imprimée une fois. Le composant se charge des changements de page.



Avant de recharger la liste de valeurs, commencez par vider toutes les valeurs en mémoire avec la fonction `sdClearData`.

## CRepLabel

Le composant `CRepLabel` est le composant le plus courant à utiliser dans une impression. Pour modifier sa valeur à l'exécution, dans le cas d'une impression simple, modifiez sa propriété `caption`. Dans le cas d'une impression de type liste, utilisez la fonction `sdAddValue`.

## CRepMemo :

Pour le composant `CRepMemo`, pour chaque occurrence alimentez les lignes d'un composant de type `CMemo` avec les fonctions classiques `sdSlAdd`, puis assignez la valeur de l'occurrence avec la fonction `sdAddLines`.

```
p Commentaires...
p          B
d          PI
C          callp      sdSlClear(F2:'Memo1':'lines')
C      A_IDLIVRE      SETLL      SDDMCMLV
C      A_IDLIVRE      READE      SDDMCMLV          5
C          DOW        NOT *IN58
C          callp      sdSlAdd(F2:'Memo1':'lines':
C                      %trim(E_LINE))
C      A_IDLIVRE      READE      SDDMCMLV          5
C          ENDDO
C          callp      SDADDLINES(F2:'RepMemo1':F2:'Memo1.lines
p          E
```

## CRepChart :

Pour le composant `CRepChart`, alimentez une composant de type `CChart`, puis assignez la valeur de l'occurrence avec la fonction `sdAddChart`.

```
p Graphe...
p          B
d          PI
DNB          s          10  0
C      Clef1          KLIST
C          KFLD          A_IDLIVRE
C          KFLD          ANNEE          4  0
C          KFLD          MOIS          2  0
C          eval          ANNEE=2005
C          callp      sdSeriesClear(F2:'Series1')
```

```

C          eval      MOIS=1
C          dow       MOIS <=12
C          eval      NB=0
C      CLEF1  SETLL    SDDMORDER1
C      CLEF1  READE    SDDMORDER1
C          dow       not *in57
C          eval      NB=NB+D_QUANTITE
C      CLEF1  READE    SDDMORDER1
C          enddo
C          if        NB>0
C          callp     sdAddPie(F2:'Series1':NB:MONTH(MOIS))
C          endif
C          eval      MOIS=MOIS+1
C          enddo
C          callp     sdAddChart(F2:'RepChart1':F2:'Chart1')
P          E

```

## Formatage d'une occurrence

Pour modifier l'apparence d'une valeur (couleur de fond, police, etc...) utilisez dans les fonctions sdAddValue les paramètres optionnels 4 et 5 qui font référence à un composant de type CRepAttributes. L'occurrence sera imprimée avec les propriétés du composant CRepAttributes.

```

C          if        A_STOCK =0
C          callp     sdAddValue(F2:'STOCK':%char(A_STOCK)
C                      *ON:F2:'Attr1')
C          else
C          callp     sdAddValue(F2:'STOCK':%char(A_STOCK)
C                      *ON:F2:'Attr1')
C          endif

```

### Remarque :

*Dans le cas d'une impression simple, modifiez directement les propriétés du composant.*

## Ruptures

Pour organiser une rupture dans une impression de type liste, suivez la procédure suivante :

- Créez une bande de type alr\_détail et une bande de type alr\_link
- Ajoutez un élément dans la collection Groups de la bande détail.
- Renseignez la propriété DataSource de cet élément pour déterminer le champ sur lequel a lieu la rupture. La propriété DataSource doit pointer sur un champ de la bande détail

- Renseignez les propriétés GroupFooter et GroupHeader de cet élément pour déterminer les bandes imprimées lors d'une rupture.
- Alimentez uniquement les champs sur la bande de type detail.

Exemple 1 :

Posez 2 bandes, une de type alr\_Detail et une de type alr\_Link que nous appellerons "BandDetail" et "BandHeader".

Posez sur BandDetail un composant de type CRepLabel nommé NOCLI.

Posez sur BandHeader un composant de type CRepLabel nommé NOCLI2.

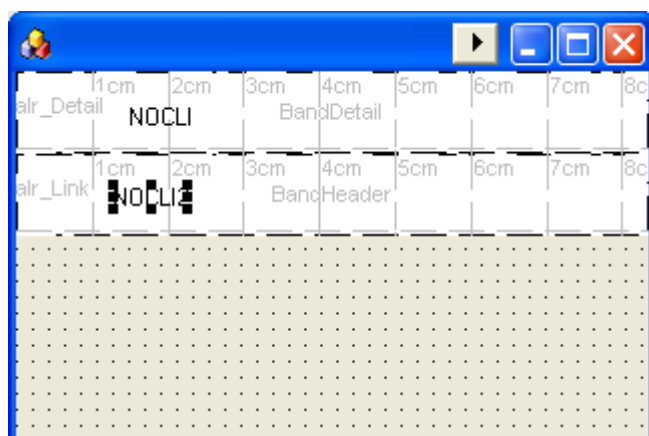


Figure 65

Ajoutez un élément dans la collection Groups de "BandDetail".

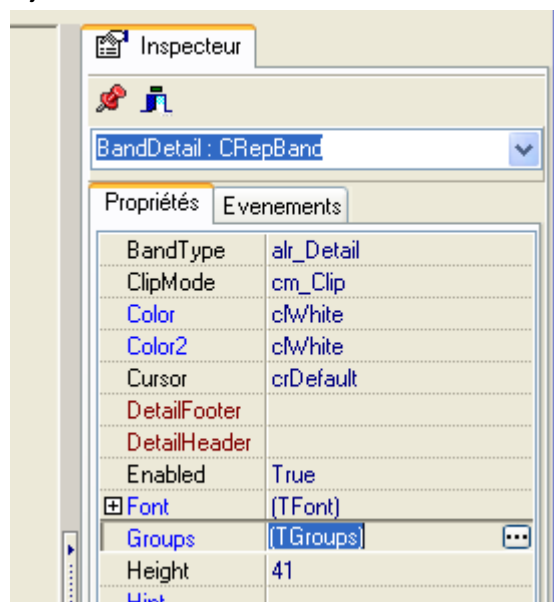


Figure 66

Modifiez les propriétés DataSource et GroupHeader de cet élément.

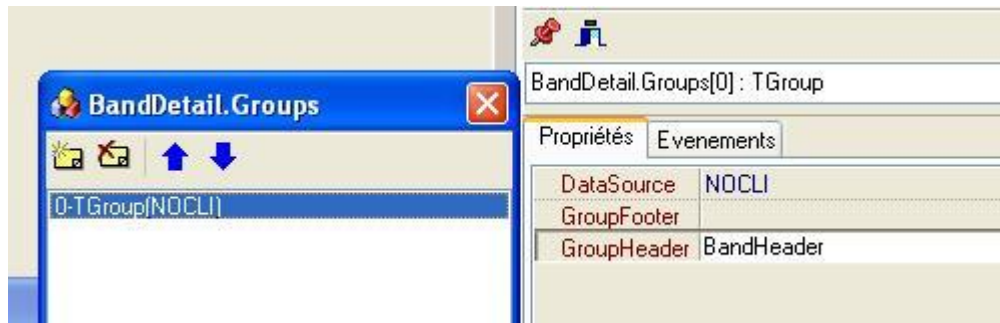


Figure 67

Dans le programme, renseignez uniquement le champ NOCLI. (pas le champ NOCLI2) en utilisant la fonction `sdAddValue`.

C	<code>callp</code>	<code>sdAddValue (F1: 'NOCLI' : %char (NOCLI) )</code>
---	--------------------	--

A chaque modification de la valeur de NOCLI, la bande "BandHeader" sera imprimée comme entête de la rupture.

*Remarque :*

*Le nombre de ruptures peut être multiple. Pour voir un exemple de rupture double, voyez le programme de démonstration SDDMORDERS et l'écran SDDMORDERS.*

Exemple 2 ( correspond à silverdemo/sddmorders) :

Maquette :

Commande de livres																						
Date commande		Numéro livre		Titre livre		Quantité		Prix														
Departement :		DEP_																				
Numéro de client :		NOCLI_		Nom :		NOMCLI_																
Date		IDLIVRE		TITRE		Quantite		Prix		NOCLI		NOMCLI		TOTAL		DEP						
Total															Total							
Page n°															#Page#							

Figure 68

Résultat :

**Aperçu avant impression**

1/9 Fermer Pages 1 9

**Commande de livres**

Date commande	Numéro livre	Titre livre	Quantité	Prix
Département : Bouches du Rhône				
Numéro de client : 125 Nom : Dalles				
2005/01/08	2	La ligne verte	1	6.00
2005/02/02	30	Manifeste du parti communiste	1	7.07
2005/02/26	2	La ligne verte	1	6.00
2005/04/09	67	J'accuse	2	14.00
2005/04/10	61	Le petit prince	2	4.00
2005/04/22	2	La ligne verte	1	6.00
2005/04/30	4	Misery	3	18.00
2005/05/02	2	La ligne verte	1	6.00
2005/05/16	87	JSP Professionnel	3	156.00
2005/05/29	23	La Cause des adolescents	1	21.00
2005/06/05	2	La ligne verte	2	12.00
2005/06/14	2	La ligne verte	2	12.00
2005/07/01	2	La ligne verte	3	18.00
2005/07/12	54	Voyez comme on danse	1	20.97
2005/07/19	16	Visual Basic 6	3	105.00
2005/09/01	68	Germinal	1	6.00
2005/09/20	2	La ligne verte	5	30.00
2005/10/07	2	La ligne verte	3	18.00
2005/10/13	48	L'amant	3	21.10
2005/10/24	40	La peau de chagrin	2	8.00
2005/10/24	50	La vie devant soi	1	4.70
2005/11/07	2	La ligne verte	3	18.00
2005/11/27	53	Pour qui sonne le glas	3	20.80
2005/11/28	1	Shining	1	7.00
Total				545.77
Numéro de client : 8942 Nom : Fortin				
2005/03/09	53	Pour qui sonne le glas	2	13.90
2005/03/30	83	Sac d'os	3	22.80

Figure 69

## Rupture sans détails

Voir **Erreur ! Source du renvoi introuvable.**

## Dessiner un tableau

Pour dessiner un tableau, il n'y a pas de composant CRepGrid.

Utilisez différents composants CRepShape les uns à côté des autres.

Dans l'exemple ci-dessous, le carré 2 commence exactement là où finit le carré 1,

Et la propriété sides du carré 2 ne comprend pas siLeft sinon, il y aurait un trait plus gros là où les deux carrés se touchent.

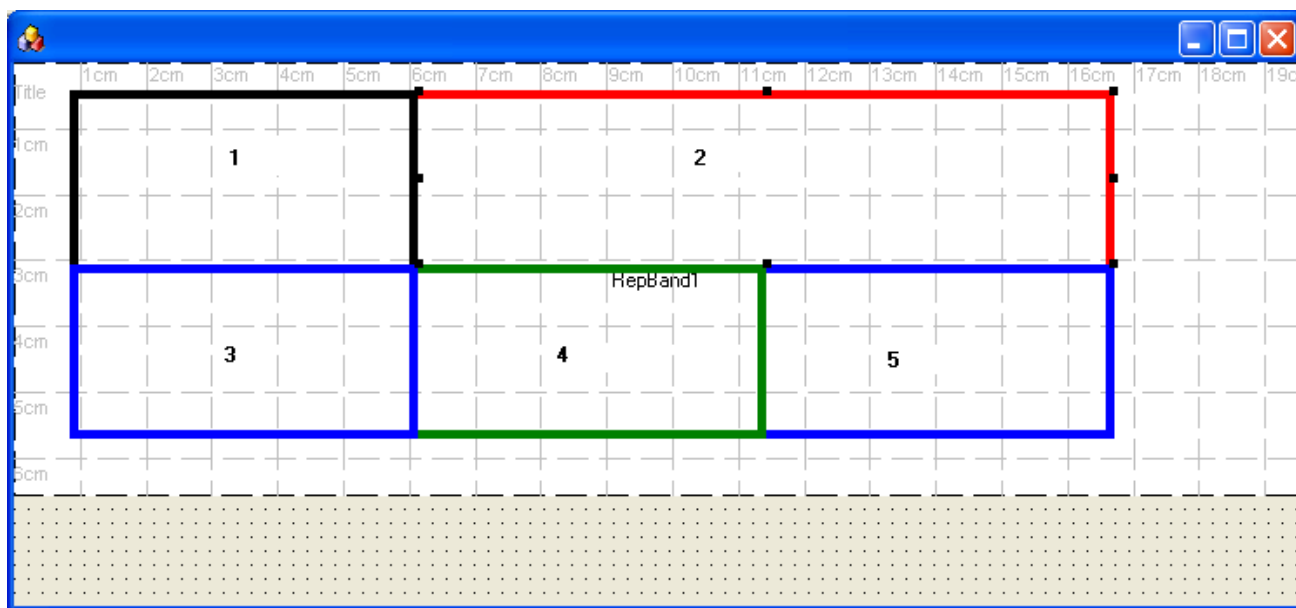


Figure 70

Remarque :

*Vous serez peut-être tenté de faire se chevaucher des carrés afin de n'avoir qu'un seul bord visible. Si le résultat obtenu est satisfaisant dans le designer, il n'en sera peut-être pas de même à l'impression. Les deux bords sensés se chevaucher pourraient former un bord plus épais voir même deux lignes distinctes. Préférez donc la méthode expliquée ci-dessus.*

**Reports de valeurs:**

Si vous souhaitez afficher un report en entête de page, utilisez une bande de type `alr_PageHeader`, posez un composant `CRepLabel`. Nommez le composant `CRepLabel` avec le même nom que celui se trouvant dans la bande détail avec le caractère « `_` » à la fin du champ.

Modifiez la propriété `Previous` à `true` pour la précédente valeur de la bande détail.

**Sauts de page**

Pour effectuer une impression sur deux pages, posez deux bandes de type `alr_title`. Sur la seconde bande, modifiez la propriété `NewPage` à `true`.

De même, dans une impression de type liste, il est possible de commencer une nouvelle page à chaque rupture en modifiant la propriété `NewPage` de la bande de type `GroupHeader`.

**Formatage au milieu d'une chaine**

Il n'existe pas l'équivalent du composant RichEdit pour les impressions.

Pour afficher un mot avec une police différente au milieu d'une phrase, vous pouvez utiliser deux composants CrepLabel, et déplacer le second de la manière suivante :

Dlabel1	s	50	varying
C	eval	label1 = sdGetFormName(F1)+'.replabel1'	
C	callp	sdSetString(f1:'replabel1':'caption':	
C		'ville :')	
C	callp	sdRefresh(f1:'replabel1')	
C	callp	sdSetString(f1:'replabel2':'caption':	
C		'Grenoble')	
C	callp	sdSet(F1:'replabel2':'left':	
C		label1+'.left' + '+' label1+'.width')	
C	callp	sdSet(F1:'replabel2':'top':	
C		label1+'.top' )	
C	callp	sdpreview(F1:'report1')	

## Impression en pdf

Au lieu d'appeler la fonction sdPrint, appelez la fonction sdSavePdf pour enregistrer l'état dans un fichier pdf.

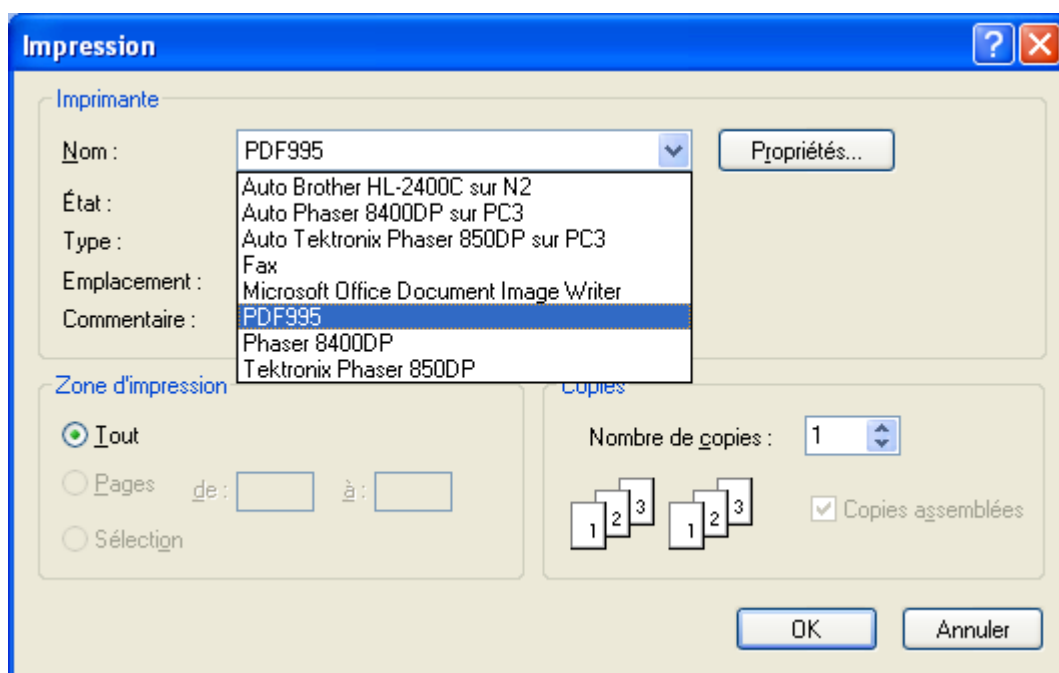
Le prototype de la fonction sdSavePdf est le suivant :

d	sdSavePdf	pr	1000	varying
d	pform		5u 0	const
d	Component		30	varying value
d	Path		1000	varying value options(*NOPASS)
d	Erase		N	value options(*NOPASS)
d	Code		10i 0	options(*NOPASS)

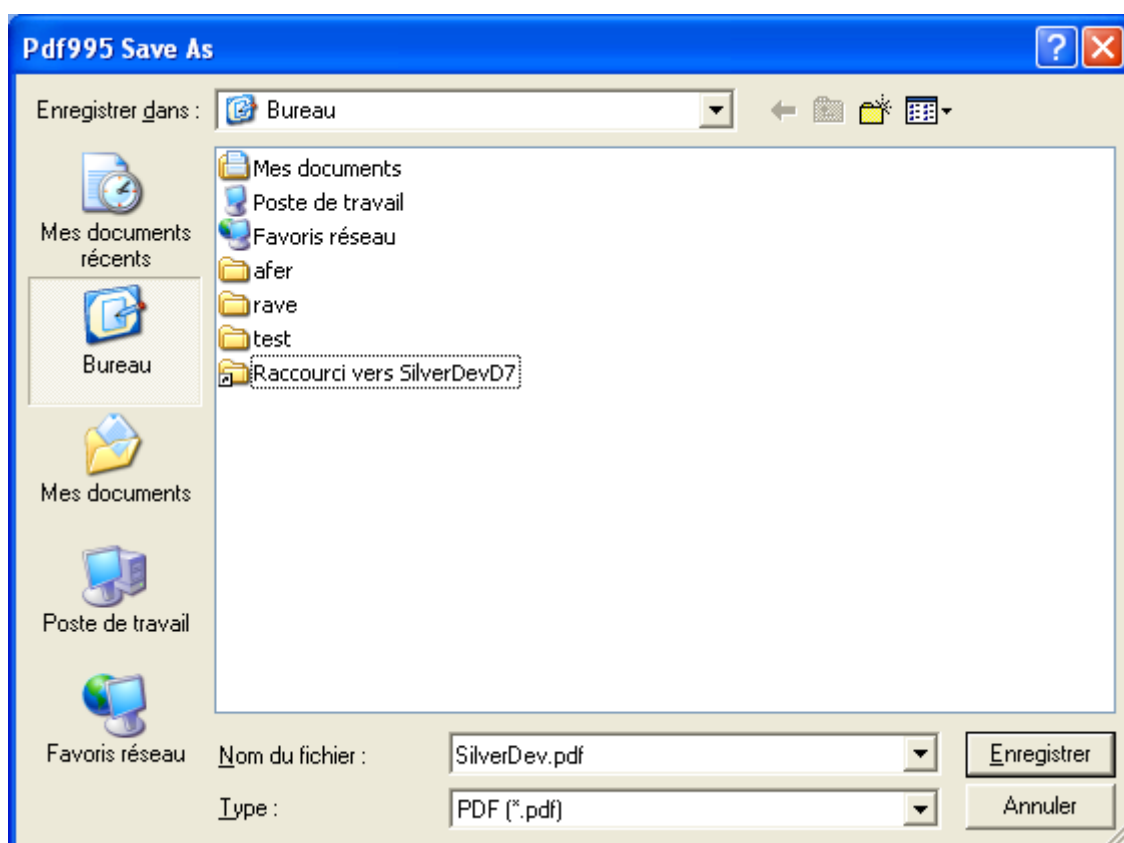
Si le paramètre path a la valeur '\*SELECT', une boîte de dialogue demande à l'utilisateur l'emplacement du fichier.

Une autre solution pour générer un document pdf est d'utiliser une imprimante virtuelle pdf.



**Figure 71**

Cette imprimante vous demandera un nom de fichier pour enregistrer le pdf.

**Figure 72**

## Choix d'imprimante

Si vous souhaitez laisser la possibilité à l'utilisateur de choisir l'imprimante de destination, utilisez la fonction sdDialog avec un composant CPrintDialog

```

C          if          sdDialog(F1: 'PrintDlg1')
C          callp      Load
C          callp      sdPrint(F2: 'Rep1')
C          endif

```

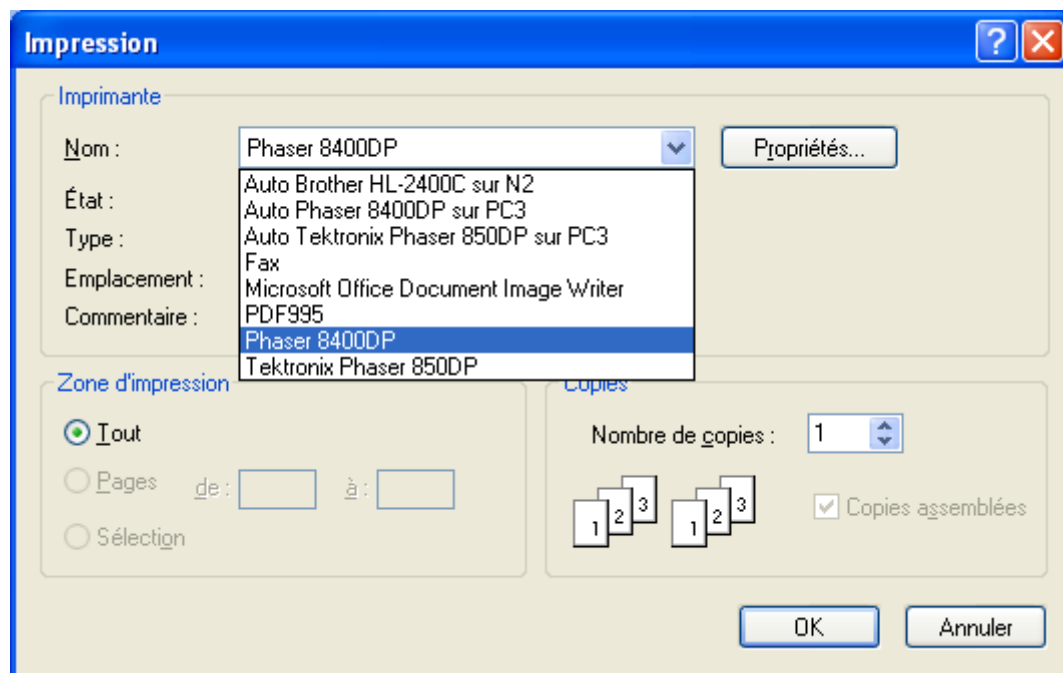


Figure 73

## Imposer une imprimante

Pour imposer une imprimante, utilisez le composant CPrinter qui vous permet de manipuler l'imprimante.

```

Dlist      s          *
D i        s          10u 0
C          eval      list = sdGetlist(F1: 'printer1.printers')
C          for       i = 0 to sdGetListCount(List)-1
C          if        %trim(sdGetListLabel(List:i))= 'PDF995'
C          callp     sdSetInt(F1: 'Printer1': 'PrinterIndex':i)
C          leave
C          endif
C          endfor
C          callp     sdPrint(F1: 'sf11')

```

## FromPage ToPage

Pour imprimer une partie des pages, utilisez les propriétés FromPage et ToPage.  
Ces propriétés existent sur le composant CReport et sur le composant CPrintDialog, mais c'est à vous de transférer les valeurs de l'un à l'autre.

### Exemple :

```

DFromPage      s      10i 0
DToPage        s      10i 0
DprintRange    s      10i 0
C              if      sdDialog(F1: 'printDialog1')
C              eval     printRange=sdGetInt(F1: 'PrintDialog1':
C                        'PrintRange')
C              if      PrintRange=2
C              eval     FromPage = sdGetInt(F1: 'PrintDialog1':
C                        'FromPage')
C              eval     ToPage = sdGetInt(F1: 'PrintDialog1':
C                        'ToPage')
C              else
C              eval     FromPage = 0
C              eval     ToPage = 0
C              endif
...
C              callp    sdPrint(F1: 'Rep1')
C              endif

```

## Conversions mm/Pixels

Un peu de mathématiques.

Soit D la distance en mm sur la page imprimée.

Soit M la marge de l'imprimante en mm (toutes les imprimantes ont une marge)

La propriété left d'un composant est en pixels.

La valeur de pixels par inch d'un écran (PPP) par défaut est 96 pixels/inch.

Un inch correspond à 25,4mm.

Pour que un composant soit imprimée à la distance D mm sur une page, ce composant doit donc être placé avec la propriété left à :

$$(D-M)/25,4 * 96$$

# Chapitre 15. Evenements

## Gestionnaire d'évènement

Les composants possèdent des évènement. C'est lors de ces évènements que votre code sera exécuté sur l'as400. La partie cliente signale qu'un évènement a eu lieu, et le programme sur l'as400 exécute la procédure associée à cet évènement.

L'activation des évènements se fait à la conception en cochant l'évènement dans l'inspecteur de propriété. Nous verrons plus loin qu'il est possible d'activer/désactiver des évènements à l'exécution.



Figure 74

Dans le programme RPG, vous associez une procédure à cette évènement avec la fonction `sdSetCallBack`.

```
c          callp      sdSetCallBack (F1
c                               : 'BtnUp.OnClick'
c                               : %paddr (
c                               'BTNUP_ONCLICK' ) )
```

## Paramètres d'évènement

La procédure associée à l'évènement doit avoir au minimum le prototype suivant :

```
pBtnUp_OnClick      B
d                    PI
d PevtInf            *   const options(*nopass)
```

En consultant l'aide, vous constaterez que certains évènements possèdent des paramètres.

Ces paramètres sont en fait des valeurs passées par le programme client au programme RPG.

Toujours dans l'aide, vous pourrez voir que certains de ces paramètres sont modifiables. C'est à dire que la valeur pourra être modifiée dans le programme RPG et que la nouvelle valeur sera transmise au programme client à la fin du gestionnaire d'évènement.

Si vous utilisez le générateur, la liste des paramètres sera automatiquement ajoutée dans les bloc de type /\*EVENT



### Exemple :

Pour effectuer un drag and drop, nous devons utiliser ces paramètres.

Le drag and drop se déroule en 3 étapes :

\_L'utilisateur commence le glisser à partir d'un composant source. Pour cela modifiez la propriété dragMode du composant source en dmAutomatic. (Il est aussi possible de laisser la valeur dmManual, il faut alors appeler la fonction sdBeginDrag dans l'evt onMouseDown)

\_L'utilisateur survole le composant destination.

Vous pouvez alors décider si le drop est autorisé à l'aide du paramètre Drag qui est un paramètre modifiable. Le curseur sera modifié en conséquence.  ou 

```
pListBox2_OnDragOver...
p                    B
d                    PI
d PevtInf            *   const options(*nopass)
d parm              ds      based(pevtinf)
d Win                5u 0
d evt                48a
d drag                n
d name                100a  varying
d PosX                10i 0 overlay(PosXA)
d PosY                10i 0 overlay(PosYA)
c                    If      Name='ListBox1' and Win=F1
c                    eval    drag = *On
c                    Else
c                    eval    drag = *Off
c                    End
```

p

E

\_Si dans l'évt DragOver, vous avez autorisé le drop, alors l'évt OnDragDrop se produit au moment où l'utilisateur lâche le bouton de la souris.

```
pListBox2_OnDragDrop...
p                                B
d                                PI
d PevtInf                        *    const options(*nopass)
* -----
d List                          s        *    Inz(*Null)
d Index                        s        10  0
d Nbr                          s        10u 0
d Val                          s        20a  Varying
c                               Eval      List=SdGetList(F1:'ListBox1')
c                               If        List = *Null
c                               callp    sdShowMessage('pb')
c                               return
c                               endif
c                               Eval      Nbr=SdGetListCount(List)
c                               Callp    SdBeginUpd(F1:'ListBox2':'Items')
c                               For       Index=0 to Nbr-1
c                               If        SdGetListSelected(List:Index)
c                               Eval      Val= SdGetListLabel(List:Index)
c                               Callp    SdSlAdd(F1:'ListBox2':'Items':Val
c                               EndIf
c                               Callp    SdEndUpd(F1:'ListBox2':'Items')
c                               Callp    SdFreeList(List)
c                               EndFor
p                                E
```

Certains paramètres d'évènements sont de type ensemble. (Voir **Erreur ! Source du renvoi introuvable.** )

Pour savoir si une valeur est contenue dans cet ensemble, utilisez la fonction sdIsInSet.

### Exemple :

L'événement onMouseDown contient le paramètre Shift qui peut contenir les valeurs suivantes :

0	La touche Maj est enfoncée.
1	La touche Alt est enfoncée.
2	La touche Ctrl est enfoncée.

3	Le bouton gauche de la souris est enfoncé.
4	Le bouton droit de la souris est enfoncé.
5	Le bouton central de la souris est enfoncé.
6	On a double-cliqué sur la souris.

```
pButton1_OnMouseDown...
p          B
d          PI
d PevtInf          *    const options(*nopass)
DParams          ds          based(PevtInf)
D Win          5u 0
D Event          48
D Button          10i 0
D Shift          32
D PosX          10i 0
D PosY          10i 0
C          if          sdIsInSet(Shift:3)
C          ...
C          endif
p          E
```

Remarque :

Plusieurs valeurs peuvent être contenues dans un ensemble.

```
*/EVENT SFL1_OnChangeCellValue
d parm          ds          based(pevtinf)
d Win          5u 0
d evt          48a
  *Modifiable
d Allow          N
  *Non modifiable
d ColName          100a  varying
d Row          10i 0
d Value          100a  varying
```

Dans l'évènement onchangeCellValue, si vous interrogez la valeur de la cellule avec sdGetCell, vous n'obtiendrez pas la même valeur que celle passée dans le paramètre value.

En effet, cet évènement se produit avant la validation, ce qui permet de modifier le paramètre allow.

## Affectation dynamique d'évènements

Il est aussi possible d'activer un évènement à l'exécution à l'aide de la fonction sdEnableEvt.

**Prototype :**

```

d sdEnableEvt      pr
d pform            5u 0 const
d eventName        48a  varying value
d Enabled          N    value

```

**Remarque :**

A chaque évènement activé dans le designer ou à l'aide de la fonction `sdEnableEvt`, doit correspondre une procédure. Cette association se fait à l'aide de la fonction `sdSetCallBack`.

**Exemples d'utilisation :****Cas 1 :**

Lorsque vous modifiez la valeur de la propriété `checked` d'un composant `CCheckBox` par code, l'evt `onclick` est déclenché. Si vous souhaitez que l'evt ne soit pas déclenché, utilisez la fonction `sdEnableEvt` pour désactiver temporairement l'evt `on click`

```

C      callp      sdEnableEvt (F1: 'CheckBox1.onclick': *OFF)
C      callp      sdSetBool (F1: 'CheckBox1': 'Checked': *ON)
C      callp      sdEnableEvt (F1: 'CheckBox1.onclick': *ON)

```

**Cas 2 :**

Il est possible de créer dynamiquement des composants. On peut alors demander le signalement d'événements de ces composants créés dynamiquement.

```

D i      s      10u 0
D j      s      10u 0
D name1   s      50  varying
D name2   s      50  varying
C      for      i=0 to 5
C      eval      Name1='Mnu'+%char(i)
C      callp     sdCreate (F1: 'TMenuItem': Name1)
C      callp     sdAddMnuItem (F1: 'menu1.items': F1: Name1)
C      callp     sdSetString (F1: Name1: 'caption':
C      'Elt'+%char(i))
C      for      J=0 to 5
C      eval      Name2='Sub'+%char(i)+'_'+%char(j)
C      callp     sdCreate (F1: 'TMenuItem': Name2)
C      callp     sdAddMnuItem (F1: Name1: F1: Name2)
C      callp     sdSetString (F1: Name2: 'caption':
C      'Sous Elt'+%char(i))

```



```
C      callp      sdEnableEvt(F1: Name2+'.OnClick':*ON)
C      callp      sdSetCallBack(F1:Name2+'.OnClick':
C                  %PADDR('EVT1'))
C      EndFor
C      EndFor
```

---

# Chapitre 16. Fiches modales

---

## Introduction

Certaines fenêtres peuvent être affichées en modal. L'affichage de type modal est bloquant. Cela signifie que, au sein de la même application, seule cette fiche est accessible par l'utilisateur.

---

## ModalResult

Pour connaître l'action de l'utilisateur, interrogez la propriété `ModalResult` de la fiche ou la valeur de retour de la fonction `sdShowModal`.

### Attention :

Il ne faut pas confondre la propriété `modalresult` d'un composant `CButton` et celle de la fiche.

Ces deux propriétés sont liées, mais elles n'ont pas la même signification.

Pour la fenêtre, la modification de la propriété `modalResult` à une autre valeur que `mrNone`, ferme la fenêtre.

Pour un bouton, la propriété `modalResult` signifie que le bouton transmettra sa valeur à la propriété `modalResult` de sa fenêtre lorsque l'utilisateur cliquera sur le bouton.

---

## Exemple

Dans un premier programme, il sera possible d'appeler un second programme dont la fenêtre est modale à partir d'un click sur un bouton :



**Figure 75**



Figure 76

Programme appelé : (modal)

C	*entry	plist		
C		parm	\$F1ModRes	
C		parm	PID	2 0

~/EVENT BtnOk_OnClick				
, * ----- *				
, * Description :				
, * ----- *				
C	Eval	PID	=sdGetNum(F1: 'Liste': 'KeySelected')	
C	callp		sdSetInt(F1: '*FORM': 'ModalResult': Mrok)	

Programme appelant :

~/EVENT BtnAuthors_OnClick				
, * ----- *				
, * Description :				
, * ----- *				
DModRes	s	10u 0		
C	call	'SDDMBKS3'		
C	parm		ModRes	
C	parm		PID	2 0

```

C          if      ModRes=MrOk
C          callp    sdSetNum(F1: 'IDAUT': 'value':PID)
C          endif

```

Remarque :

La fenêtre en modale pourra avoir les propriétés suivantes :

Position = poScreenCenter afin qu'elle apparaissent au milieu de l'écran

BorderIcons = seulement biSystemMenu d'activé

Deux boutons :

Un bouton marqué « annuler » dont la propriété modalResult est à mrCancel et la propriété Cancel à true. Ainsi lorsque l'utilisateur fait « Echap » la fenêtre se ferme.

Un bouton marqué « ok » dont la propriété default sera à true. La propriété modalResult du bouton ok pourra être à mrOk, mais il est plus souvent utile de le mettre à mrNone et de déclencher l'événement onclick du bouton, effectuer des tests , puis modifier la propriété modalResult de la fenêtre par code.

## Execution seule

Une fenêtre ne peut être affichée en modale que si elle n'est pas visible.

Si vous essayez d'afficher en modal une fenêtre visible, vous aurez le message suivant :

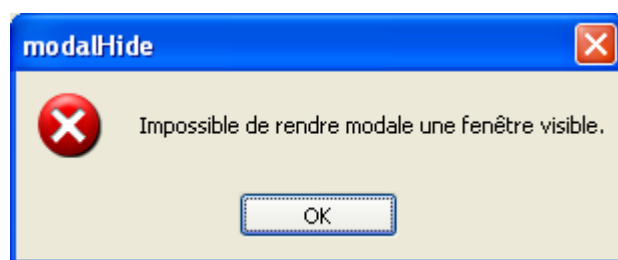


Figure 77

La première fenêtre dans une application silverdev est automatiquement rendue visible lors de sa création.

Dans certains cas, vous créerez une application avec une fenêtre modale qui aura pour vocation d'être appelée depuis une autre fenêtre, mais vous souhaitez la lancer directement depuis myDesk.

Vous aurez alors le message ci dessus qui apparaîtra lorsque vous lancerez l'application seule. Pour éviter d'avoir ce message, ajoutez l'instruction dans le moment initstart

```

*/BLOCK RPGINITSTART
* ----- RPGINITSTART : Begining of Initializations procedure
C          callp    sdHideMainForm

```

et l'instruction :

```
*/BLOCK RPGAFTERSHOW  
* ----- RPGAFTERSHOW : After show(F1)  
C          if          F1=1  
C          callp       sdEnd  
C          endif
```

Dans le moment AfterShow.

---

# Chapitre 17. Menu Outils

## Commande

Utilisez le menu outils/commandes pour exécuter une commande as400

La liste des paramètres est constituée dynamiquement.

Utilisez les touches page down et page up pour faire défiler les paramètres.

Les commandes interactives ne sont pas admises dans le programme Designer.

Exemple avec la commande CRTPGM :

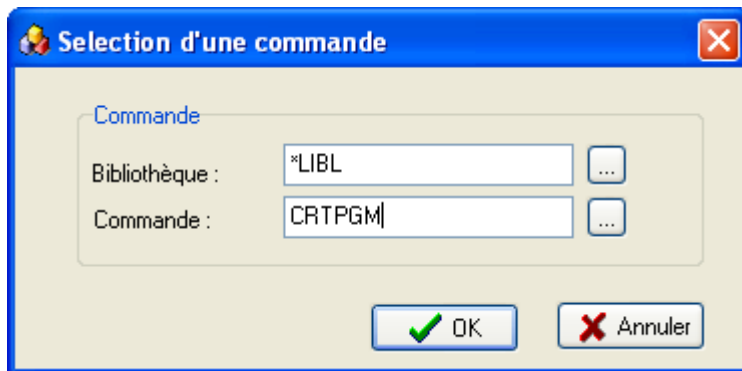


Figure 78

**CRTPGM Créer un programme**

Options

**PGM**

Programme  Nom

Bibliothèque  Nom, \*CURLIB

**MODULE**

Module  Nom, générique\*, \*ALL

Bibliothèque  Nom, \*LIBL, \*CURLIB, \*USRLIBL

**TEXT**

Texte 'descriptif'  Valeur alpha, \*ENTMODTXT...

**ENTMOD**

Module procédure entrée pgm  Nom

Bibliothèque  Nom, \*LIBL, \*CURLIB, \*USRLIBL

**BNDSRVPGM**

Lier programme de service  Nom, générique\*, \*ALL

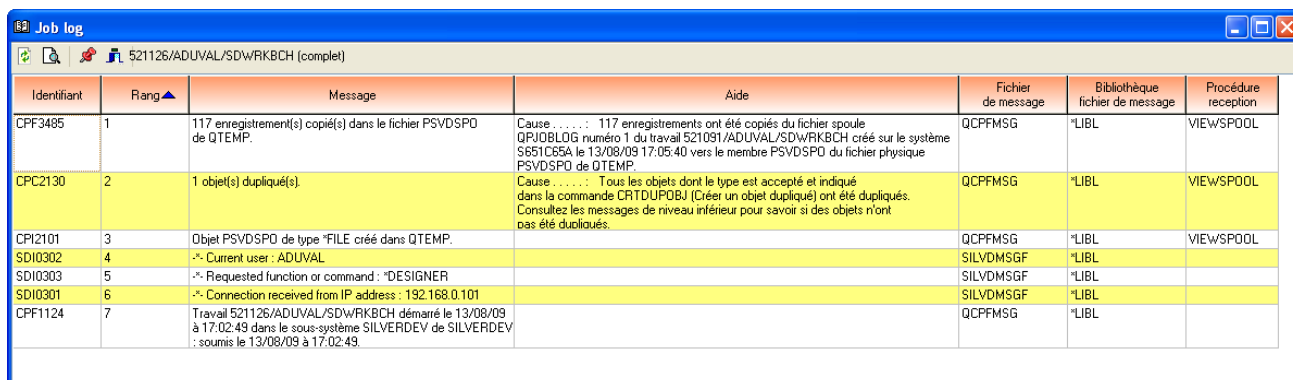
Bibliothèque  Nom, \*LIBL

Exécuter Annuler

Figure 79

## Job log

Utilisez le menu outils/job log pour afficher le job log



Identifiant	Rang	Message	Aide	Fichier de message	Bibliothèque fichier de message	Procédure reception
CPF3485	1	117 enregistrement(s) copié(s) dans le fichier PSVDSPD de QTEMP.	Cause . . . . . : 117 enregistrements ont été copiés du fichier spoule QPJOBLOG numéro 1 du travail 521091/ADUVAL/SDWRKBCH créé sur le système S651C65A le 13/08/09 17:05:40 vers le membre PSVDSPD du fichier physique PSVDSPD de QTEMP.	QCPFMSG	*LIBL	VIEWSPPOOL
CPC2130	2	1 objet(s) dupliqué(s).	Cause . . . . . : Tous les objets dont le type est accepté et indiqué dans la commande CRTDUPOBJ (Créer un objet dupliqué) ont été dupliqués. Consultez les messages de niveau inférieur pour savoir si des objets n'ont pas été dupliqués.	QCPFMSG	*LIBL	VIEWSPPOOL
CP12101	3	Objet PSVDSPD de type *FILE créé dans QTEMP.		QCPFMSG	*LIBL	VIEWSPPOOL
SDI0302	4	*- Current user : ADUVAL		SILVDMMSGF	*LIBL	
SDI0303	5	*- Requested function or command : *DESIGNER		SILVDMMSGF	*LIBL	
SDI0301	6	*- Connection received from IP address : 192.168.0.101		SILVDMMSGF	*LIBL	
CPF1124	7	Travail 521126/ADUVAL/SDWRKBCH démarré le 13/08/09 à 17:02:49 dans le sous-système SILVERDEV de SILVERDEV ; soumis le 13/08/09 à 17:02:49.		QCPFMSG	*LIBL	

Figure 80

## Liste des travaux

Utilisez le menu Outils/travaux pour afficher la liste des travaux.

Utilisez le bouton  pour modifier les critères de sélection.

Utilisez le click droit pour choisir une action sur un travail.

Travaux

Travail : SDWRKBCH/ADUVAL/\*ALL Type : Etat : \*ACTIVE

Rang	Nom travail	Utilisateur	Numéro travail	Etat	Etat actif	Fonction	Type fonction	File d'attente	Commande silverdev	Utilisateur	Adresse IP
4	SDWRKBCH-ADUVAL	521147	*ACTIVE	TIMW	SDWRKBCH	P	*DEV				
3	SDWRKBCH-ADUVAL	521142	*ACTIVE	TIMW	SDWRKBCH	P	*DEV				
2	SDWRKBCH-ADUVAL	521126	*ACTIVE	RUN	SDWRKBCH	P	*DEV	*DESIGNER	ADUVAL	192.168.0.101	
1	SDWRKBCH-ADUVAL	519033	*ACTIVE	TIMW	SDWRKBCH	P	*DEV				

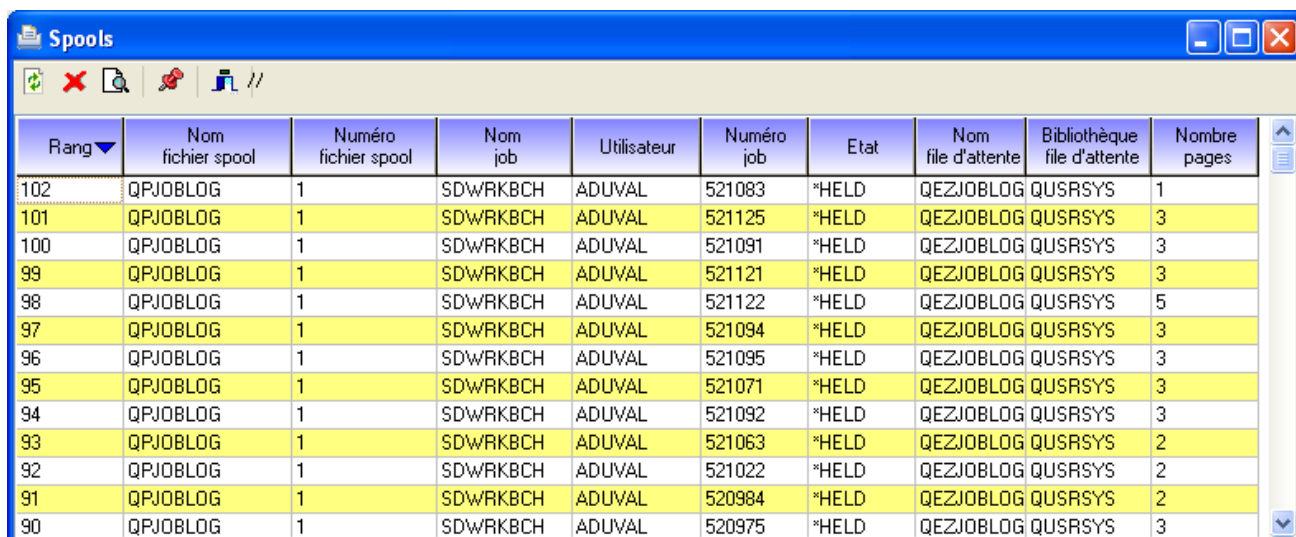
Figure 81

## Liste des spools

Utilisez le menu Outils/fichiers spool pour afficher la liste des fichiers spool.

Double cliquez sur une ligne pour afficher le spool.





Rang	Nom fichier spool	Numéro fichier spool	Nom job	Utilisateur	Numéro job	Etat	Nom file d'attente	Bibliothèque file d'attente	Nombre pages
102	QPJOBLOG	1	SDWRKBCH	ADUVAL	521083	*HELD	QEZJOBLOG	QUSRSYS	1
101	QPJOBLOG	1	SDWRKBCH	ADUVAL	521125	*HELD	QEZJOBLOG	QUSRSYS	3
100	QPJOBLOG	1	SDWRKBCH	ADUVAL	521091	*HELD	QEZJOBLOG	QUSRSYS	3
99	QPJOBLOG	1	SDWRKBCH	ADUVAL	521121	*HELD	QEZJOBLOG	QUSRSYS	3
98	QPJOBLOG	1	SDWRKBCH	ADUVAL	521122	*HELD	QEZJOBLOG	QUSRSYS	5
97	QPJOBLOG	1	SDWRKBCH	ADUVAL	521094	*HELD	QEZJOBLOG	QUSRSYS	3
96	QPJOBLOG	1	SDWRKBCH	ADUVAL	521095	*HELD	QEZJOBLOG	QUSRSYS	3
95	QPJOBLOG	1	SDWRKBCH	ADUVAL	521071	*HELD	QEZJOBLOG	QUSRSYS	3
94	QPJOBLOG	1	SDWRKBCH	ADUVAL	521092	*HELD	QEZJOBLOG	QUSRSYS	3
93	QPJOBLOG	1	SDWRKBCH	ADUVAL	521063	*HELD	QEZJOBLOG	QUSRSYS	2
92	QPJOBLOG	1	SDWRKBCH	ADUVAL	521022	*HELD	QEZJOBLOG	QUSRSYS	2
91	QPJOBLOG	1	SDWRKBCH	ADUVAL	520984	*HELD	QEZJOBLOG	QUSRSYS	2
90	QPJOBLOG	1	SDWRKBCH	ADUVAL	520975	*HELD	QEZJOBLOG	QUSRSYS	3

Figure 82

### Liste de bibliothèques

Modifiez la liste de bibliothèques du job associé au designer en utilisant le menu "Outils/Liste de bibliothèques". Cette liste de bibliothèques est nécessaire pour compiler les programmes, choisir un contexte de développement, etc...

Les bibliothèques systèmes sont affichées en rouge et ne peuvent être modifiées,

Les bibliothèques de production sont affichées en bleu et ne peuvent être modifiées.

La bibliothèque courante est affichée en vert et peut être modifiée en utilisant le bouton .


Les bibliothèques utilisateurs sont affichées en noir et peuvent être modifiées en utilisant les boutons .



Figure 83

*Remarque :*

*De nombreuses fonctionnalités nécessitent que Silverdev soit dans la liste de bibliothèque.*

### **Propriétés**

Le menu "Outils/Propriétés" permet de rendre visible l'inspecteur de propriétés.

### **Assistant**

Le menu "Outils/Assistant" permet d'afficher l'assistant base de données.

Cet outil permet d'afficher la description d'un fichier de données sur l'iSeries/400 et de créer des contrôles à partir des champs du fichier.

Les contrôles créés dépendent du type de champ et ses propriétés dépendent de certains attributs du champ (Nom, Longueur, description, type, valeurs possibles etc..)

Dans l'assistant, une arborescence affiche la liste des bibliothèques et des fichiers de données de l'iSeries/400.

Pour développer un nœud, utiliser le click droit pour faire apparaître un menu déroulant.

Double cliquez sur le nom d'un fichier pour voir sa description.

Vous pouvez imprimer cette description en utilisant un click droit.

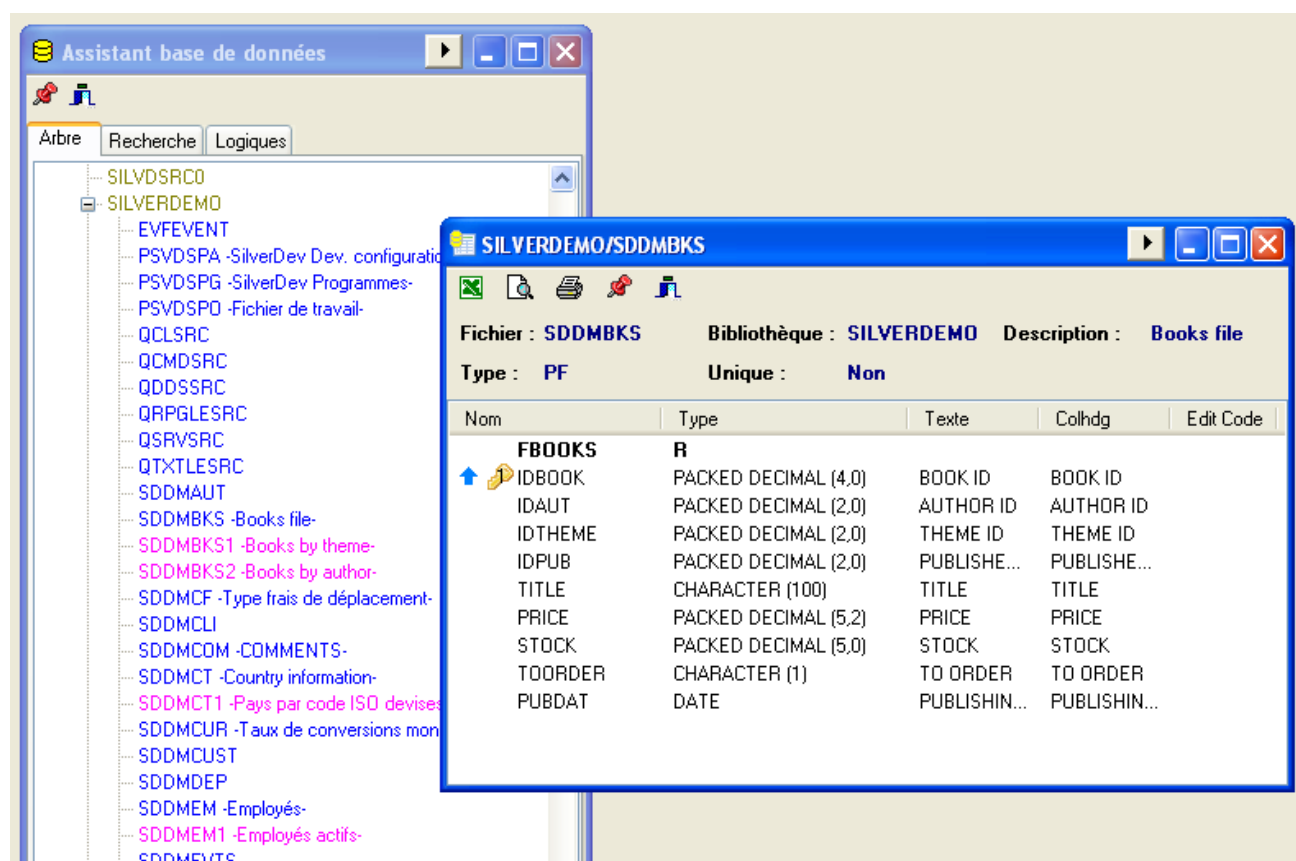


Figure 84

Le second onglet permet d'effectuer une recherche d'un fichier. Il est possible d'utiliser le caractère générique \*



Figure 85

En bleu, les fichiers physiques, et en violet, les fichiers logiques.

Affichez tous les fichiers logiques dépendants d'un physique en utilisant le click droit, menu « Logiques »

## Drag and drop

Depuis la description, il est possible de faire glisser les champs pour créer des composants aux propriétés correspondant à la définition des champs

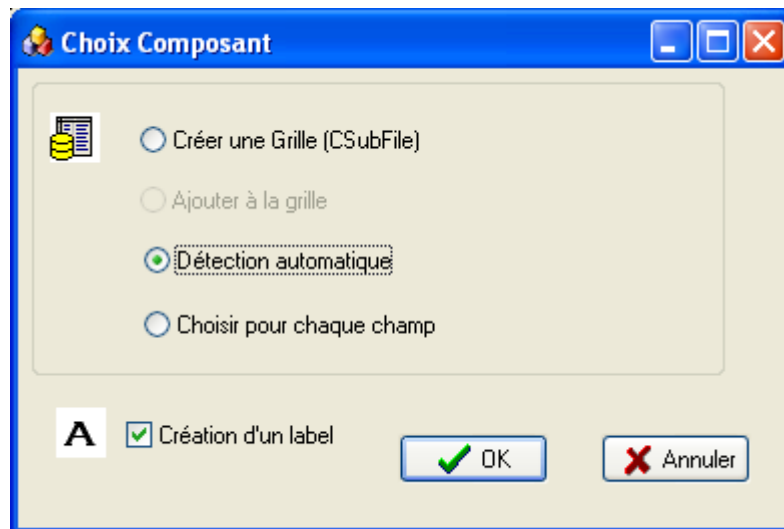


Figure 86

## Recherche

Dans la partie basse de la fenêtre, il est possible de saisir le nom d'un fichier.

La liste de tous les fichiers correspondant est affichée. (Il est possible de saisir une étoile à la fin du nom)

## Fichiers logiques

Depuis cette liste, vous pouvez avec un click gauche trouver la liste des fichiers logiques dépendant du fichier.



Figure 87

## Schémas base de données

Le menu schémas base de données permet de créer des schémas de base de données, affichant les relations entre les fichiers.

Vous pouvez faire glisser une table depuis l'assistant base de données. La liste des champs est affichée.

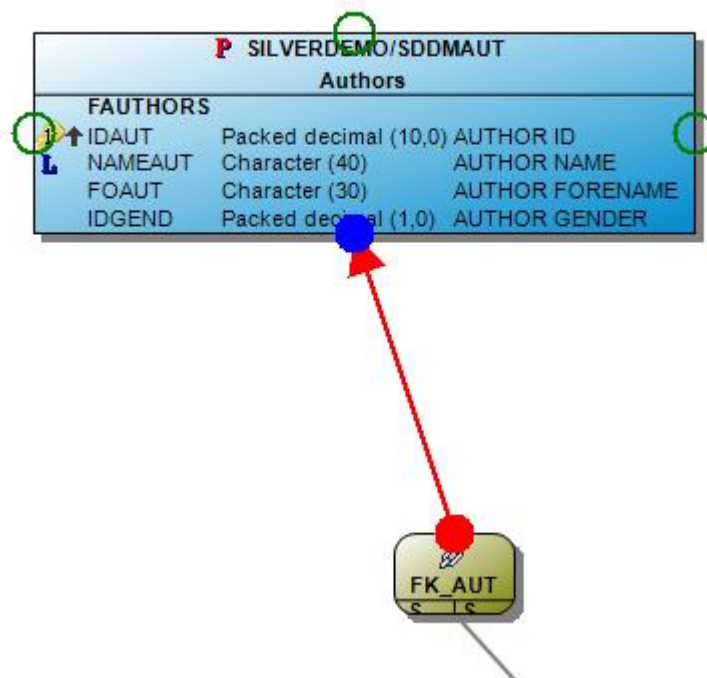
Les contraintes référentielles entre les fichiers sont affichées automatiquement, ainsi que les liens entre logiques et physiques.

Il est possible d'ajouter manuellement des relations entre les fichiers.

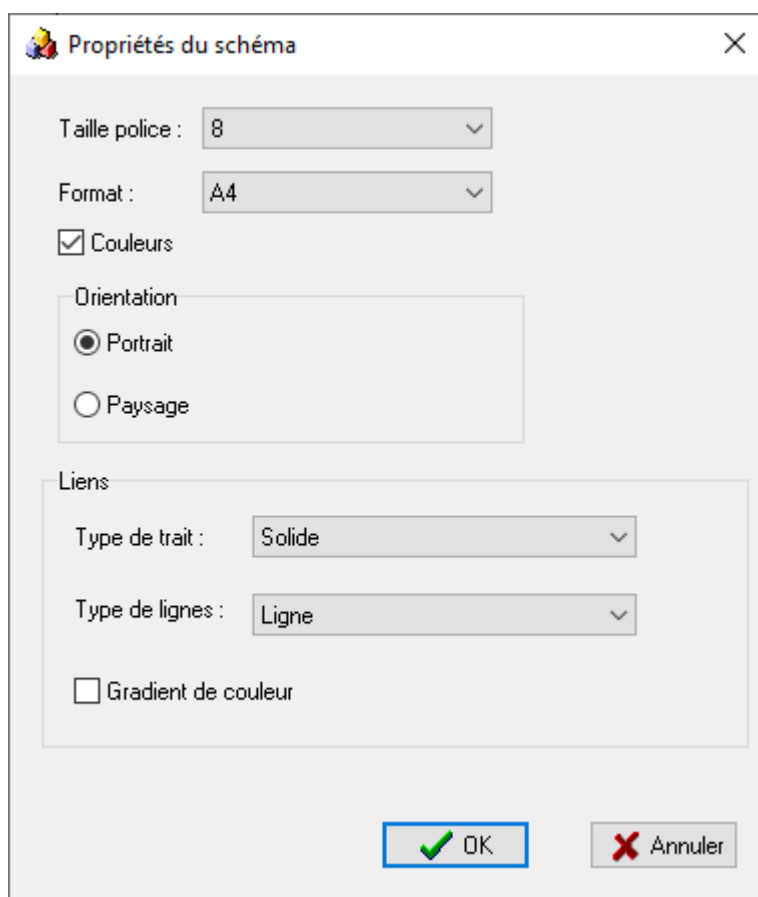
Le schéma permet d'avoir une représentation de la base de données.

A partir du schéma, il est aussi possible de créer des applications automatiquement. Pour cela, voir le document RAD chapitre wizard

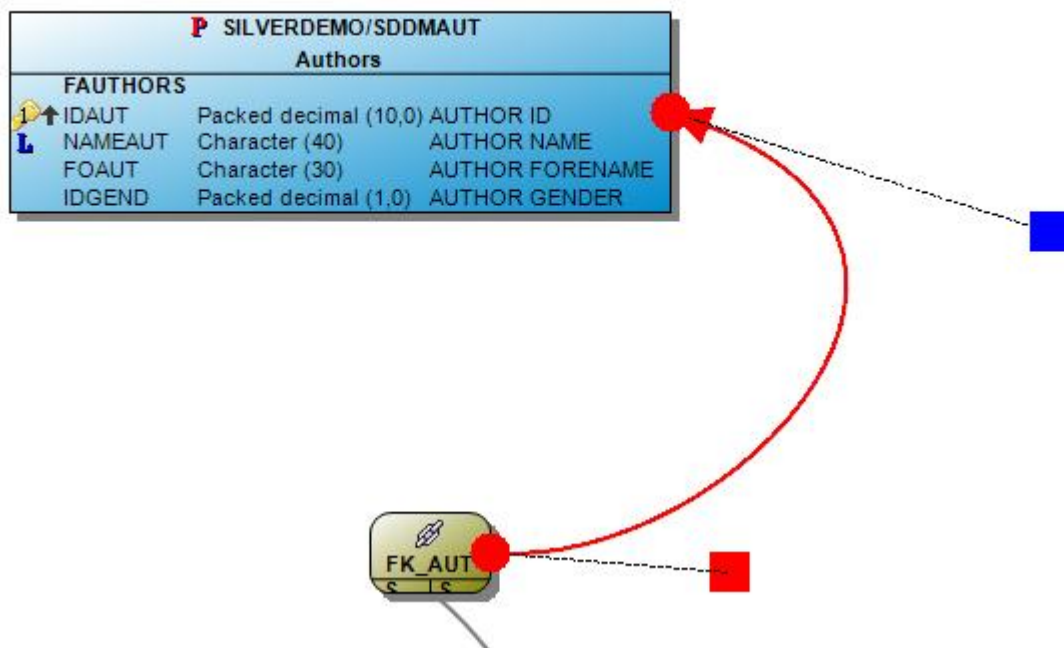
Pour modifier les points d'ancrage d'une relation, sélectionnez cette relation et faites glisser le point d'ancrage :



En effectuant un click droit sur le schéma, vous pouvez modifier les propriétés du schéma :

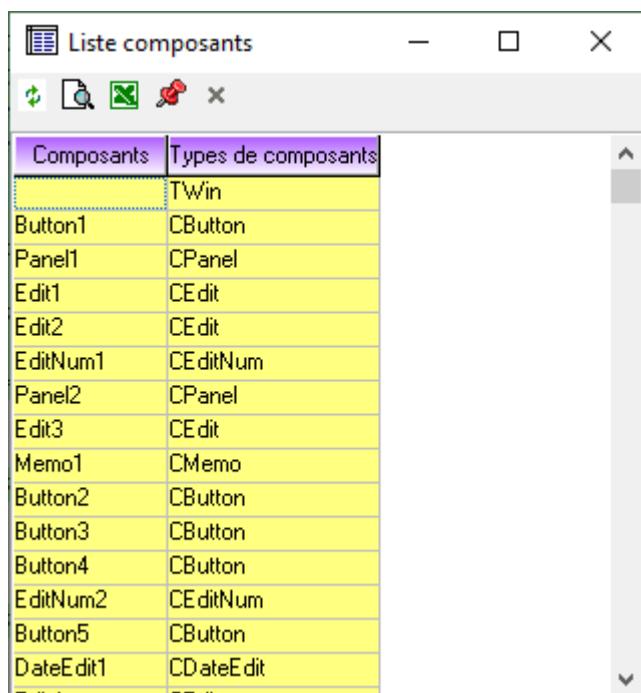


Si vous sélectionnez courbe pour le type de ligne, vous pouvez alors sélectionner les relations et déplacer les points de contrôle :



## Liste de composants

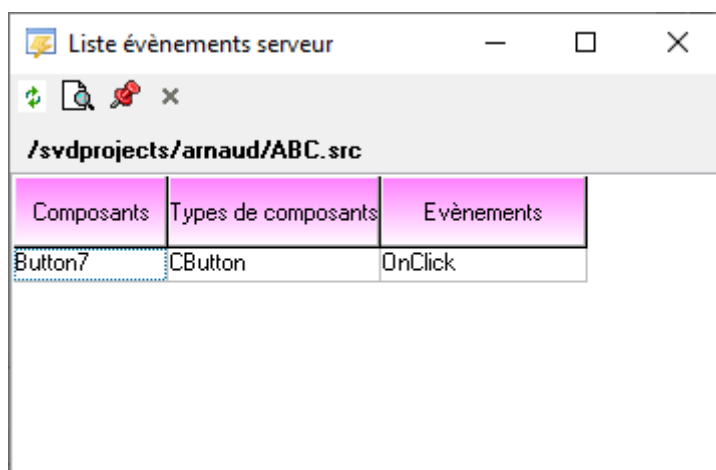
Le menu "Outils/Liste des composants" permet de voir la liste des composants de la fiche en cours.



Composants	Types de composants
	TWin
Button1	CButton
Panel1	CPanel
Edit1	CEdit
Edit2	CEdit
EditNum1	CEditNum
Panel2	CPanel
Edit3	CEdit
Memo1	CMemo
Button2	CButton
Button3	CButton
Button4	CButton
EditNum2	CEditNum
Button5	CButton
DateEdit1	CDateEdit

## Liste évènements serveur

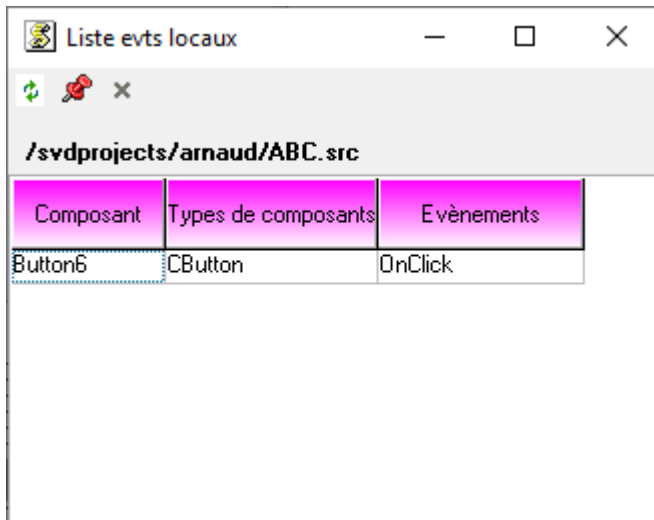
Le menu « outils/liste des évènements serveur » permet de voir la liste des événements serveur de la fenêtre.



Composants	Types de composants	Evènements
Button7	CButton	OnClick

## Liste des évènements locaux

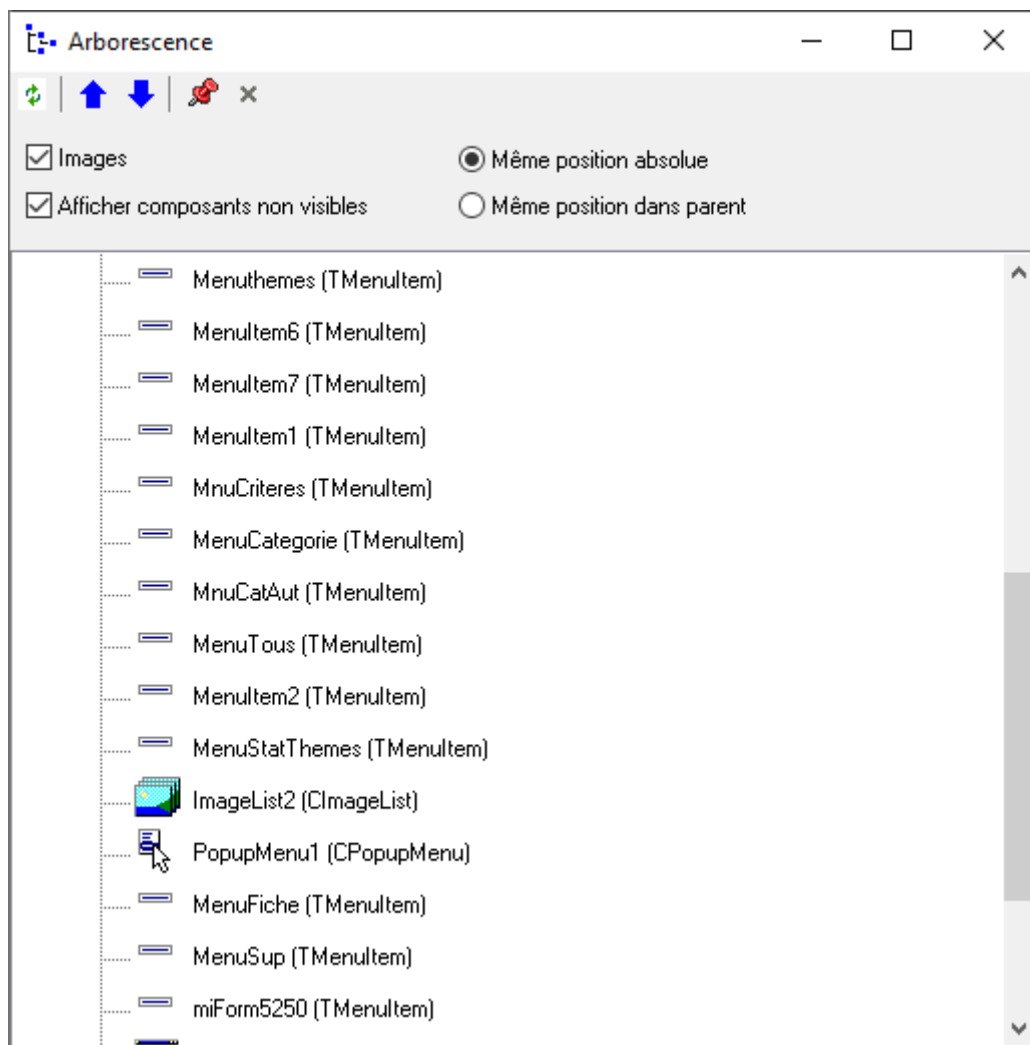
Le menu « outils/liste des évènements locaux » permet de voir la liste des événements locaux de la fenêtre.



### Arborescence

Le menu "Outils/Arborescence" permet de voir la fiche en cours sous la forme d'une arborescence. Il est alors possible de déplacer des contrôles d'un parent à un autre. Ceci est une alternative à la solution du couper-coller pour changer un contrôle de parent. Cette solution a l'avantage de ne pas perdre les événements.



**Figure 88**

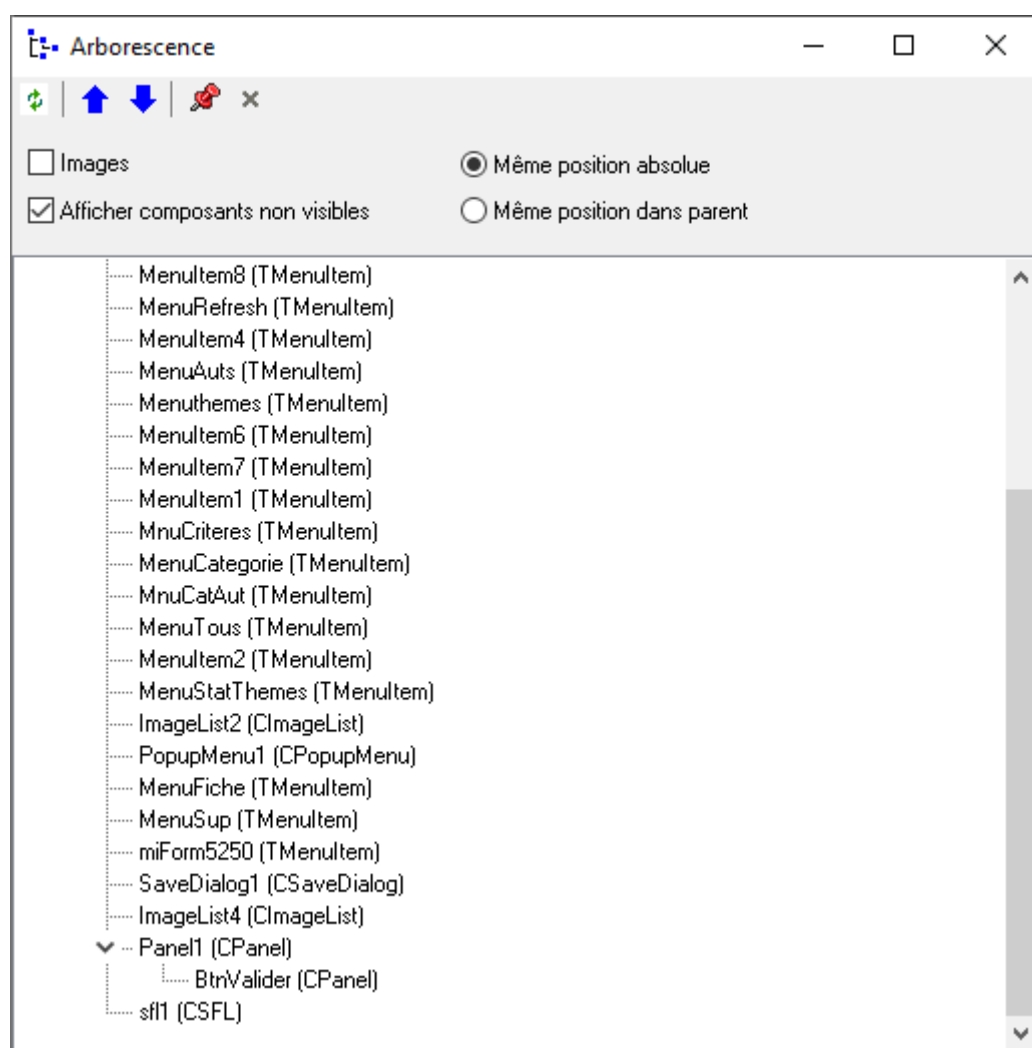


Figure 89

La case à cocher images permet d'afficher ou non l'icone representant le composant. Sans les images, plus de composants sont affichés dans la fenêtre.

Les boutons radio permettent de modifier le comportement.

Si « Même position absolue » est coché, la position relative par rapport au bord haut gauche de la fenêtre ne changera pas.

Si « Même position dans parent » est choché, la position relative par rapport au bord haut gauche du nouveau parent sera la même que celle dans l'ancien parent.

## Décompilation

Si vous perdez un source mais que vous avez toujours l'objet écran, vous pouvez décompiler l'écran à partir du menu outils/décompiler.

## Explorateur

Utilisez l'explorateur pour ouvrir un membre.

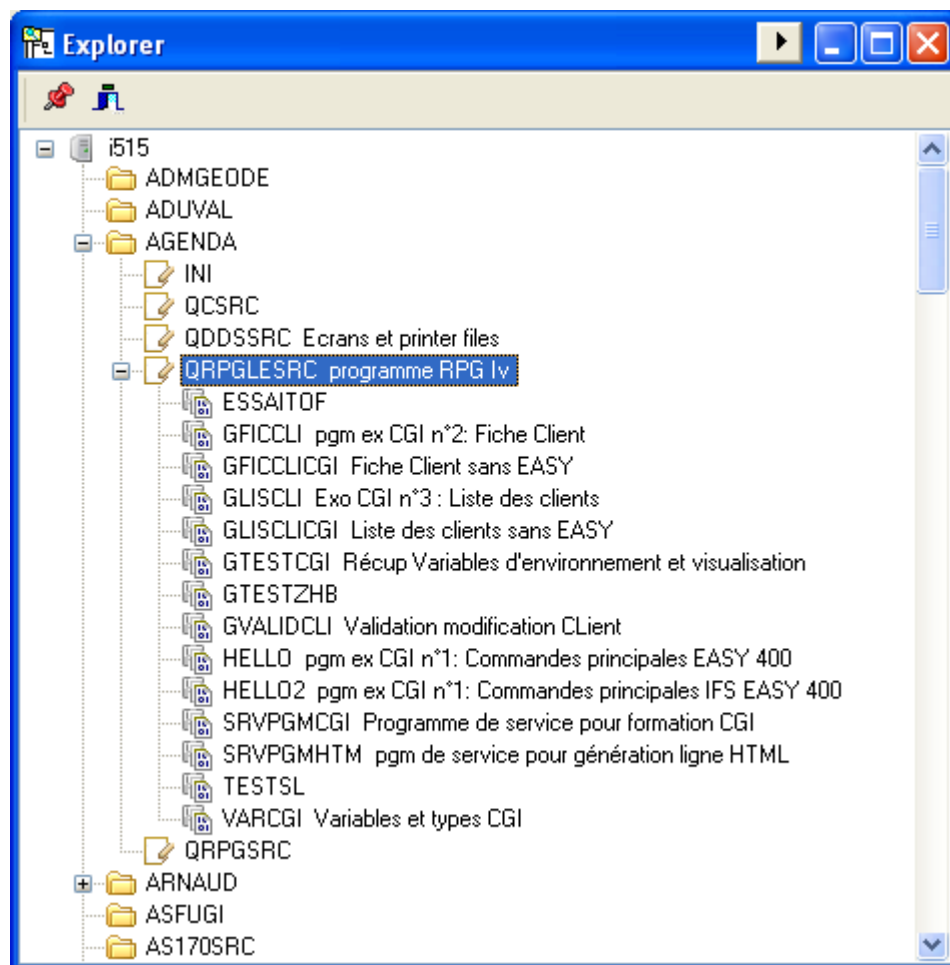


Figure 90

## Versionning

Voir le chapitre Versionning dans le document Administration

## Comparer des sources

L'outil de comparaison de sources permet de détecter les ajouts, suppressions et modifications d'une source à l'autre. Il est possible de comparer des sources provenant d'un membre, de l'ifs, ou de l'historique (versionning)

## Comparer des contenus

L'outil de comparaison des contenus permet de comparer deux bibliothèques , (objets ajoutés ou supprimés), des répertoires (fichiers et répertoires ajoutés ou supprimés), ou des fichiers sources (membres ajoutés ou supprimés)

## Chapitre 18. Erreurs fréquentes

Erreur	Symptôme
<p>Le nom de la fonction gestionnaire d'évènement est mal orthographié dans la fonction sdSetCallBack.</p> <p><u>Remarque :</u> Le nom doit être saisi en <b>majuscules</b>.</p>	<p>Le programme ne compile pas.</p> <p>Message du compilateur : "Des erreurs ont été détectées pendant le liage."</p>
Une variable locale à une procédure handler d'évènement est passée en paramètre à un second programme.	La variable ne contient pas les données attendues.
Le paramètre PForm a été déclaré sans le mot clef Value dans la fonction gestionnaire d'évènement.	Le programme se plante lors de l'appel d'un gestionnaire d'évènement avec le message: "Pointeur non défini pour position mémoire référencée"
Le nom d'un composant ou d'une propriété est mal saisi.	L'erreur n'est pas détectée à la compilation, mais une erreur survient à l'exécution. Pour retrouver ce type d'erreur, utilisez l'outil de Debug
Un appel à une fonction a été passé avec une fiche comme paramètre, mais la fiche n'a pas encore été créée par la fonction sdcreateForm.	Plantage de l'application avec le message "Form 0 doesn't exist"
Un programme Silverdev compilé dans le groupe d'activation *NEW	Plantage avec le message d'erreur MCH3402. Tentative d'accès à tout ou partie d'un objet qui n'existe plus.
La propriété « visible » de la fenêtre est à false. C'est la valeur par défaut. <u>Seule la première fenêtre</u> affichée est automatiquement modifiée en « visible=true » lors de sa création.	Une fenêtre « secondaire » ne s'affiche pas lors d'un appel à sdcreateForm. La première fenêtre s'affiche correctement.
<p>Perte du source d'un écran.</p> <p>Dans le programme designer, menu outils/décompiler, il est possible de décompiler un écran.</p>	
Excel fichiers corrompus.	<p>La fonction sdExport génère des fichiers excels au format de excel 2 afin de pouvoir les ouvrir avec n'importe quel programme excel. Depuis microsoft office 2010, un message "fichiers corrompus" peut apparaître.</p> <p>Allez dans "fichiers/options/centre de gestion de la confidentialité/paramètres du centre de gestion de la confidentialité/Paramètres de</p>

	blocage des fichiers. Décochez toutes les cases dans la colonne ouvrir.
--	---

---

# Chapitre 19. Solutions remplacées

L'évolution de silverdev implique que des fonctions, composants ou techniques sont remplacées par d'autres. Pour des raisons de compatibilité, les anciennes solutions sont conservées pour des raisons de compatibilité. Mais il est bon de les connaître.

---

## Composant CStringGrid

Le composant CStringGrid a été remplacé par le composant CSFL dont les possibilités sont beaucoup plus étendues.

---

## SdSelectColor, sdSelectFile, sdPrintDlg

La fonction sdSelectColor est avantageusement remplacée par l'utilisation du composant CColorDialog et la fonction sdDialog. L'utilisation du composant CColorDialog permet notamment de modifier l'apparence de la boîte de dialogue.(Couleurs personnalisée, boîte agrandie,etc..)

De même, les fonctions sdSelectFile et sdPrintDlg peuvent être remplacées par la fonction sdDialog en association avec les composants COpenDialog et CPrintdialog.

---

## SdSetSet,sdAddSet,sdDelSet

Ces fonctions sont avantageusement remplacées par les fonction sdGetSest et sdUpdSet qui manipulent une chaîne de 32 caractères.

---

## sdAddNode

La fonction sdAddNode est remplacée par la fonction sdAddNode2 qui pour le chargement de treeview avec un nombre important de noeuds est beacoup plus rapide.

De plus, la fonction sdAddNode2 est plus simple d'utilisation car elle ne necessite pas d'utiliser des compteurs.

---

# Chapitre 20. Technique avancées

Le document technique avancées regroupe les chapitres suivants :

**Fonctions de la catégorie List**

**Optimisation**

**SFL : Recopie des données en locales**

**Evenements locaux**

**Applications Multifenêtres**

**Applications MDI**

**Application multifenêtres avec des onglets**

**MultiOccurences d'une fenêtre**

**Multioccurrences , programme de service sdsrvlst**

**Exemples et astuces**

**Web services**

**Applications multilangues**

**Fonctions catégorie images**

**Fonction boîtes de dialogues**

**Fonction catégorie fichiers pc**

**Fonctions catégorie collections**

**Fonctions catégorie TStrings**

**Fonction catégorie ensembles**

**Fonctions diverses**



**Liste des fenêtres et des évènements**

**Couleurs**

**Ancrer des composants**

**Opérations de drag and drop.**

**Adaptation taille écran**

**ADO**

**Exit points**

**Interaction avec des programmes externes.**

**Programme de service sdsrvjson**

**WebBrowser**