



SharpWComp 3.2

(CLIFFJUMPER)



<http://www.wcomp.fr/>

SharpWComp fait partie du projet WComp. Celui-ci est subdivisé en plusieurs modules. Ce document présente les étapes nécessaires à l'installation des modules WComp qui ont été conçus comme une extension à SharpDevelop. Ce document détaille les différentes étapes d'installation nécessaires pour l'utilisation de SharpWComp (installation de l'environnement .NET, installation des outils pour UPnP, installation de SharpDevelop, installation du module complémentaire SharpWComp, etc.). Un tutorial sur l'utilisation de SharpWComp et sur les différents modules qui le composent complète ce document.

En vous souhaitant une bonne lecture de ce document et surtout une bonne utilisation de SharpWComp.

De nombreux contributeurs ont rendu cette réalisation possible. Voici la liste (par ordre alphabétique) des enseignants-chercheurs, ingénieurs et étudiants ayant contribué :

*Andral Christophe (étudiant)
Bourcet Alexandre (étudiant)
Carillo Oscar (étudiant)
Cheung-Foo-Wo Daniel (étudiant et doctorant)
Chungue Ludovic (étudiant)
Emsellem David (ingénieur expert)
Ferry Nicolas (étudiant et doctorant)
Fuchet Jérôme (étudiant)
Hourdin Vincent (étudiant et doctorant)
Joulie Gabriel (étudiant)
Lavirotte Stéphane (maître de conférences)
Rey Gaëtan (maître de conférences)
Tigli Jean-Yves (maître de conférences)
Weibel Samuel (étudiant)*

*Stéphane Lavirotte¹, Gaëtan Rey² et Jean-Yves Tigli³
Enseignants-Chercheurs
Membres de l'équipe de recherche Rainbow
dirigée par Michel Riveill, laboratoire I3S
<http://rainbow.i3s.unice.fr/>*

¹ Université de Nice – Sophia Antipolis / IUFM Célestin Freinet – Académie de Nice

² Université de Nice – Sophia Antipolis / IUT

³ Université de Nice – Sophia Antipolis / Polytech'Nice Sophia

Table des matières

Avant propos.....	1
1 Installation de l'environnement .NET.....	3
1.1 Microsoft .NET Framework 2.0.....	3
1.2 Microsoft .NET Framework 3.5.....	3
2 Installation des outils UPnP.....	4
3 Installation de SharpDevelop.....	5
4 Installation de SharpWComp (Addin SharpDevelop).....	6
5 Utilisation de SharpWComp, Addin SharpDevelop.....	8
5.1 Création d'un nouveau fichier SharpWComp.....	8
5.2 Présentation du concepteur graphique SharpWComp.....	9
5.3 Création d'assemblages dans SharpWComp.....	10
5.3.1 Instanciation de composants.....	10
5.3.2 Création d'un lien entre deux composants.....	10
5.3.3 Création d'un lien incompatible entre deux composants.....	11
5.4 Sauvegarde et Chargement d'un assemblage.....	11
5.4.1 Import d'un assemblage.....	11
5.4.2 Export d'un assemblage.....	13
5.4.3 Note importante.....	13
6 Création de composants SharpWComp.....	14
6.1 Création d'une solution pour votre (ou vos) composant(s).....	14
6.2 Code d'un composant à partir d'un modèle.....	14
6.3 Rendre ce composant utilisable dans SharpWComp.....	17
6.4 Icône d'un composant dans l'assemblage.....	17
6.5 Mise au point du code de votre composant.....	18
6.5.1 Développement d'un composant et tests unitaires.....	18
6.5.2 Mise au point à l'aide du débogueur.....	18
7 UPnP2Bean.....	19
7.1 UPnP et SharpWComp.....	19
7.2 Installation du module UPnP2Bean.....	19
7.3 Utilisation de UPnP2Bean.....	19
8 WSDL2Bean.....	21
8.1 WebServices et Wcomp.NET.....	21
8.2 Installation du module WSDL2Bean.....	21
8.3 Utilisation de WSDL2Bean.....	21

Table des illustrations

Figure 1: <i>Installation des Outils UPnP</i>	4
Figure 2: <i>Choix de l'emplacement d'installation</i>	4
Figure 3: <i>Installation de SharpDevelop 3.2.1.6466</i>	5
Figure 4: <i>Accord de Licence</i>	5
Figure 5: <i>Choix de l'emplacement d'installation</i>	5
Figure 6: <i>Installation des fichiers</i>	5
Figure 7: <i>Installation de SharpWComp, Addin SharpDevelop</i>	6
Figure 8: <i>Accord de licence</i>	6
Figure 9: <i>Sélection des composants à installer</i>	6
Figure 10: <i>Validation des paramètres d'installation</i>	6
Figure 11: <i>Menu Nouveau fichier</i>	8
Figure 12: <i> Icône nouveau fichier</i>	8
Figure 13: <i>Création d'un C# Container pour activer SharpWComp</i>	8
Figure 14: <i>Accès aux différents concepteurs (Source, SharpWComp, Design)</i>	9
Figure 15: <i>Concepteur SharpWComp</i>	9
Figure 16: <i>Boîte de dialogue création de lien</i>	11
Figure 17: <i>Sélection de la source</i>	11
Figure 18: <i>Sélection de la destination</i>	11
Figure 19: <i>Sélection d'un lien incompatible</i>	11
Figure 20: <i>Choix de la méthode de conversion</i>	11
Figure 21: <i>Importation d'un schéma d'assemblage</i>	12
Figure 22: <i>Sélection d'un fichier d'assemblage (wcc)</i>	12
Figure 23: <i>Concepteur Source après importation d'un assemblage (binary-counter-3.wcc)</i>	12
Figure 24: <i>Concepteur SharpWComp après chargement d'un assemblage (binary-counter-3.wcc)</i>	13
Figure 25: <i>Concepteur Design après chargement d'un assemblage (binary-counter-3.wcc)</i>	13
Figure 26: <i>Création d'une solution pour la création d'un composant</i>	14
Figure 27: <i>Création d'un composant SharpWComp</i>	15
Figure 28: <i>Lancement du Wizard UPnP2Bean</i>	19
Figure 29: <i>Sélection des dispositifs UPnP</i>	20
Figure 30: <i>Configuration de la Génération</i>	20
Figure 31: <i>Lancement du Wizard WSDL2Bean</i>	21
Figure 32 : <i>Sélection du Webservice</i>	22
Figure 33 : <i>Affichage sous forme d'arbre</i>	22
Figure 34: <i>Affichage sous forme de tableau</i>	22
Figure 35: <i>Code source d'un composant Webservice généré</i>	23

AVANT PROPOS

Le projet WComp est parti d'une idée originale d'enseignants-chercheurs et d'étudiants de l'Université de Nice - Sophia Antipolis (UNS). Son développement a été, à l'origine, soutenu par l'Ecole Supérieure en Sciences Informatiques (ESSI) maintenant département Sciences Informatiques de l'Ecole Polytechnique de l'Université de Nice - Sophia Antipolis (Polytech'Nice Sophia). Cette plate-forme est un des logiciels développés au sein de l'équipe de recherche Rainbow du Laboratoire I3S (CNRS / Université de Nice - Sophia Antipolis).

Il est ainsi un parfait exemple d'un transfert permanent entre recherche et pédagogie.

Les concepteurs de WComp enrichissent continuellement le projet WComp avec de nouveaux outils logiciels pour soutenir leurs propres recherches. Confrontés à la robustesse du projet, ils contribuent régulièrement au support de cet environnement logiciel, sans engagement, ni obligation.

Les différentes parties téléchargeables du projet qui ont été déposées à l'APP, sont soumises à des licences de différents types, sans redevance pour un usage non commercial, pour une phase d'évaluation des outils par exemple. Nous vous encourageons à lire ces licences afin d'en respecter toutes les spécificités.

Enfin, nous vous transmettons régulièrement une référence bibliographique actualisée⁴ qui vous permettra comme il est d'usage de citer notre équipe si vous utilisez WComp dans vos propres travaux, et si vous le souhaitez.

L'équipe du projet WComp vous remercie pour l'intérêt que vous portez à leurs travaux. N'hésitez pas à les contacter si vous avez des questions et pour les tenir informés de l'utilisation que vous avez de WComp. Merci d'adresser votre courrier avec les paramètres suivants pour un traitement plus rapide :

To : Stephane.Lavirotte@unice.fr, Jean-Yves.Tigli@unice.fr

Subject : [WComp User] ...

Si vous souhaitez nous faire connaître un bug dans SharpWComp ou dans un outil que nous distribuons, vous pouvez aussi le signaler grâce à l'interface Mantis qui est accessible à l'adresse suivante :

<https://trolen.polytech.unice.fr/mantis/>

Vous pouvez créer un compte et venir nous faire part d'un problème ou d'une fonctionnalité non documentée ! ;)

Bonne utilisation de SharpWComp !

⁴ Jean-Yves Tigli, Stéphane Lavirotte, Gaëtan Rey, Vincent Hourdin et Michel Riveill. « *Lightweight Service Oriented Architecture for Pervasive Computing* » in International Journal of Computer Science Issues (IJCSI), vol 4, num 1, pages 1-9, sept. 2009.

Installation de SharpWComp 3.2



1 INSTALLATION DE L'ENVIRONNEMENT .NET

Pour procéder à l'installation de l'environnement nécessaire à SharpWComp, il est nécessaire de disposer de l'environnement de programmation Microsoft .NET Framework 3.5. Si vous souhaitez avec SharpDevelop réaliser des projets pour l'environnement .NET 2.0, vous devez aussi installer .NET 2.0.



Ces environnements de développement d'application .NET sont les environnements standards. Si vous avez déjà ces environnements installés sur votre machine, il n'est pas nécessaire de les réinstaller.

1.1 MICROSOFT .NET FRAMEWORK 2.0

- | | | |
|----|--|---|
| 1. | Télécharger l'environnement Microsoft .NET Framework 2.0 | http://www.microsoft.com/downloads/en/details.aspx?FamilyID=0856eacb-4362-4b0d-8edd-aab15c5e04f5 |
| 2. | Lancer l'installation du fichier <code>dotnetfx.exe</code> pour disposer de l'environnement .NET 2.0 | |

1.2 MICROSOFT .NET FRAMEWORK 3.5

- | | | |
|----|--|---|
| 1. | Télécharger l'environnement Microsoft .NET Framework 3.5 | http://www.microsoft.com/downloads/en/details.aspx?FamilyID=333325FD-AE52-4E35-B531-508D977D32A6 |
| 2. | Lancer l'installation du fichier <code>dotNetFx35setup.exe</code> pour disposer de l'environnement .NET 3.5 | |
| 3. | Télécharger la mise à jour de l'environnement Microsoft .NET 3.5 Service Pack 1 | http://www.microsoft.com/downloads/en/details.aspx?FamilyID=AB99342F-5D1A-413D-8319-81DA479AB0D7 |
| 4. | Lancer l'installation du fichier <code>dotnetfx35setup.exe</code> pour disposer de l'environnement .NET 3.5 mis à jour | |

Vous disposez maintenant de l'environnement .NET nécessaire au bon fonctionnement de SharpWComp et des outils liés à SharpWComp.

2 INSTALLATION DES OUTILS UPnP

Afin d'utiliser la technologie UPnP et pouvoir la mettre en œuvre au sein de SharpWComp, il est nécessaire d'installer les outils suivants pour son utilisation. Ce programme vous fournira un ensemble d'outils de développement et de référence pour la création de logiciels qui sont compatibles avec les spécifications UPnP. Vous pourrez utiliser ceux-ci avec SharpWComp.



1. Télécharger les outils UPnP mis à disposition par OpenTools (Intel) : <http://opentools.homeip.net/dev-tools-for-upnp>
2. Lancer l'installation du fichier `DeveloperToolsForUPnPTechnologies.msi` et suivez les étapes pour disposer des outils UPnP (voir Figure 1 et Figure 2).



Figure 1: Installation des Outils UPnP

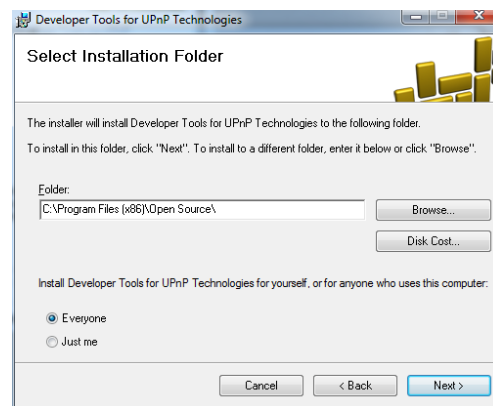


Figure 2: Choix de l'emplacement d'installation

3 INSTALLATION DE SHARPEVELOP

SharpDevelop est un environnement de développement intégré (IDE) libre et gratuit et équivalent à Visual Studio pour l'environnement .NET. SharpWComp a été conçu comme une extension (plugin ou addin) à SharpDevelop. Ceci nous permet de bénéficier de l'ensemble des fonctionnalités de SharpDevelop (génération de code correspondant à l'assemblage WComp, manipulation de la représentation graphique de l'application, génération du code exécutable correspondant à l'application, ...).

L'extension SharpWComp ayant été compilée avec la version 3.2.1.6466 de SharpDevelop, il est impératif d'utiliser cette version, sous peine de ne pouvoir faire fonctionner SharpWComp.



1. Télécharger
http://sourceforge.net/projects/sharpdevelop/files/SharpDevelop%203.x/3.2/SharpDevelop_3.2.1.6466_Setup.msi/download
2. Double cliquer sur l'exécutable téléchargé pour lancer l'installation.
3. Suivre les 5 étapes d'installation : Bienvenue (Figure 3), Accord de licence (Figure 4), Choix de l'emplacement d'installation, Copie des fichiers (Figure 5), Installation des fichiers (Figure 6), Fin de l'installation.
4. SharpDevelop est maintenant installé et vous pouvez utiliser cette IDE sans l'AddIn SharpWComp.

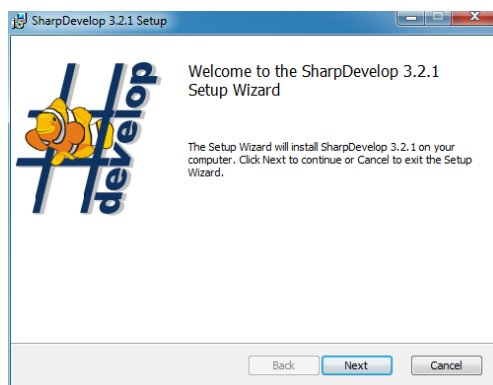


Figure 3: Installation de SharpDevelop 3.2.1.6466

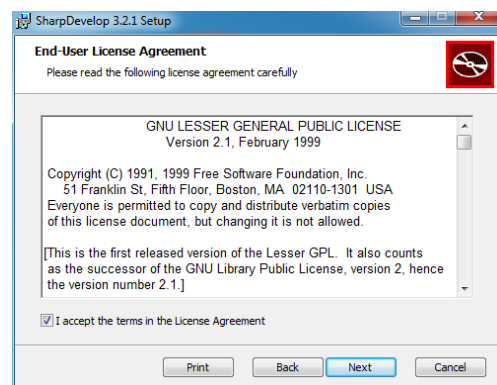


Figure 4: Accord de Licence

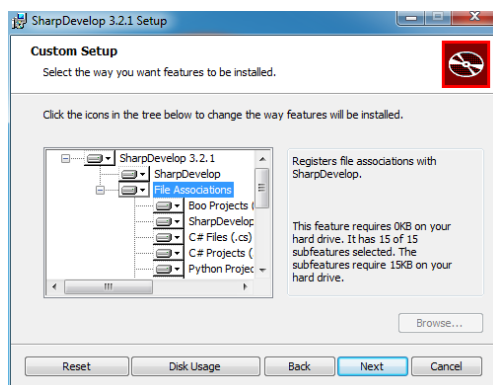


Figure 5: Choix de l'emplacement d'installation

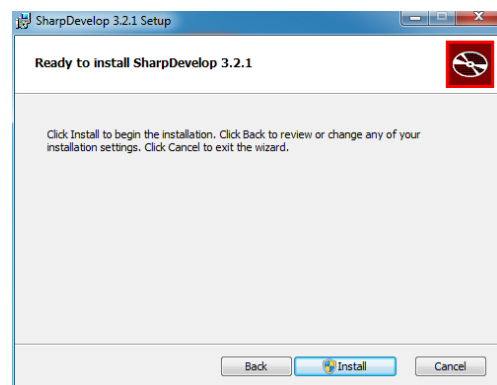


Figure 6: Installation des fichiers

4 INSTALLATION DE SHARPWCOMP (ADDIN SHARPDEVELOP)

SharpWComp est un AddIn (ou plugin) qui s'ajoute à l'environnement SharpDevelop. Il vous permettra de développer vos applications grâce à des briques logicielles que vous pourrez assembler grâce à un éditeur graphique. L'installation de SharpWComp s'effectue grâce à une application auto-extractible suivante. Il suffit de double cliquer sur l'icône de l'application téléchargée afin de procéder à l'installation de l'extension SharpWComp pour SharpDevelop.



1. Télécharger la dernière version de SharpWComp: <http://www.wcomp.fr/download>
2. Lancer l'exécution de l'installateur de SharpWComp
3. Valider l'écran de bienvenue (voir Figure 7).
4. Valider la licence de SharpWComp, Addin SharpDevelop (voir Figure 8).
5. Sélectionner les composants que vous souhaitez installer (voir Figure 9). Nous vous conseillons de tout installer si vous souhaitez utiliser toutes les fonctionnalités de SharpWComp.
6. Valider afin de procéder à l'installation des fichiers (voir Figure 10).

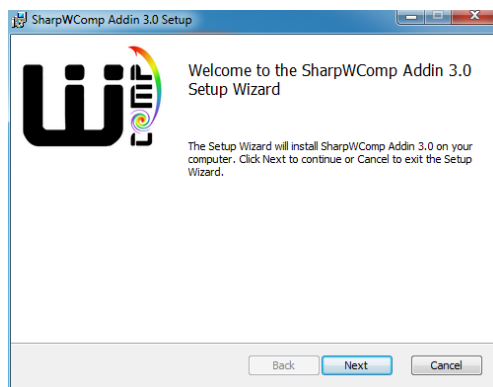


Figure 7: Installation de SharpWComp, Addin SharpDevelop

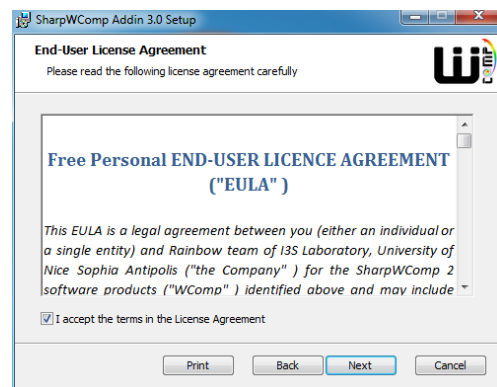


Figure 8: Accord de licence

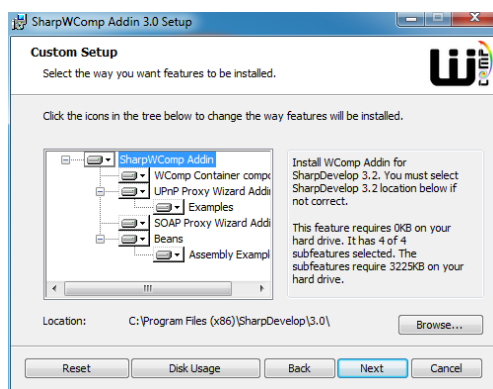


Figure 9: Sélection des composants à installer

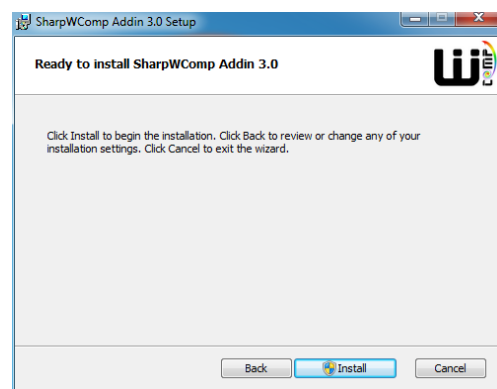


Figure 10: Validation des paramètres d'installation

Utilisation de SharpWComp 3.2



5 UTILISATION DE SHARPWCOMP, ADDIN SHARPDEVELOP

5.1 CREATION D'UN NOUVEAU FICHIER SHARPWCOMP

Pour utiliser SharpWComp dans SharpDevelop, il suffit de créer un fichier SharpWComp. L'AddIn que vous venez d'installer a ajouté un modèle SharpWComp a SharpDevelop. Il suffit de le sélectionner pour créer un nouveau fichier C# au format SharpWComp et ainsi avoir accès aux fonctionnalités.

1. Dans le menu **Fichier**, sélectionner **Nouveau** et cliquer sur **Fichier...** (ou **Ctrl+N**) (voir Figure 11), ou Cliquer sur l'icône **Nouveau Fichier** (voir Figure 12)

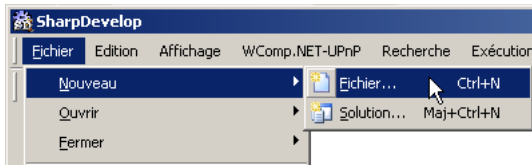


Figure 11: Menu Nouveau fichier



Figure 12: Icône nouveau fichier

2. Sélectionner la catégorie **SharpWComp** dans la boîte de dialogue qui s'est ouverte.
3. Cliquer sur le modèle **C# Container** (voir Figure 13).

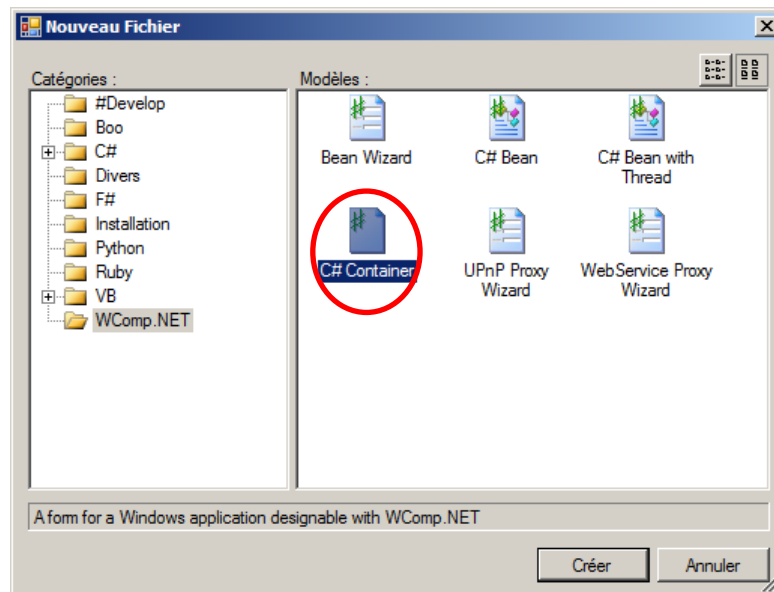


Figure 13: Création d'un C# Container pour activer SharpWComp

Vous avez maintenant accès à plusieurs modes pour l'édition de votre application. La première vue qui apparaît est l'éditeur de texte du code source de l'application : le concepteur **Source**. Vous disposez aussi du concepteur graphique **SharpWComp** et du concepteur **Design** permettant le placement des composants graphiques (widgets) de votre application. Les onglets en bas de la vue principale doivent être utilisés pour passer d'un concepteur à l'autre (voir Figure 14).

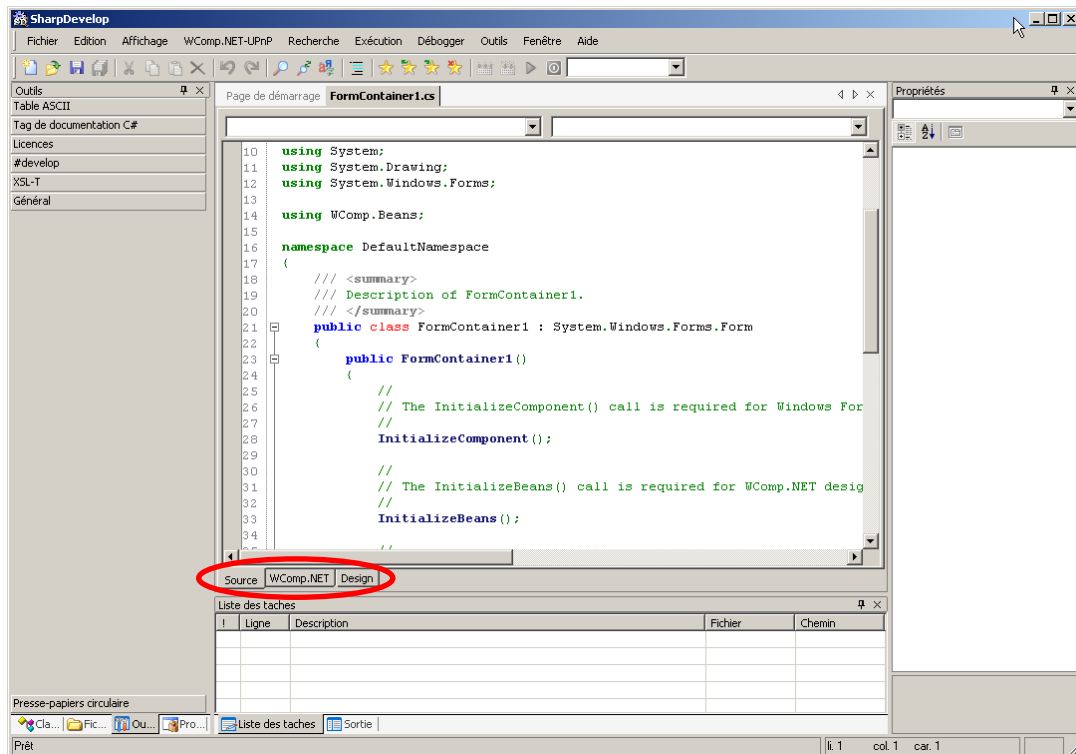


Figure 14: Accès aux différents concepteurs (Source, SharpWComp, Design)

5.2 PRESENTATION DU CONCEPTEUR GRAPHIQUE SHARPCOMP

Pour travailler avec le concepteur SharpWComp, vous devez sélectionner l'onglet SharpWComp en bas de la vue principale (voir Figure 14). Vous obtiendrez alors la vue ci-dessous (voir Figure 15).

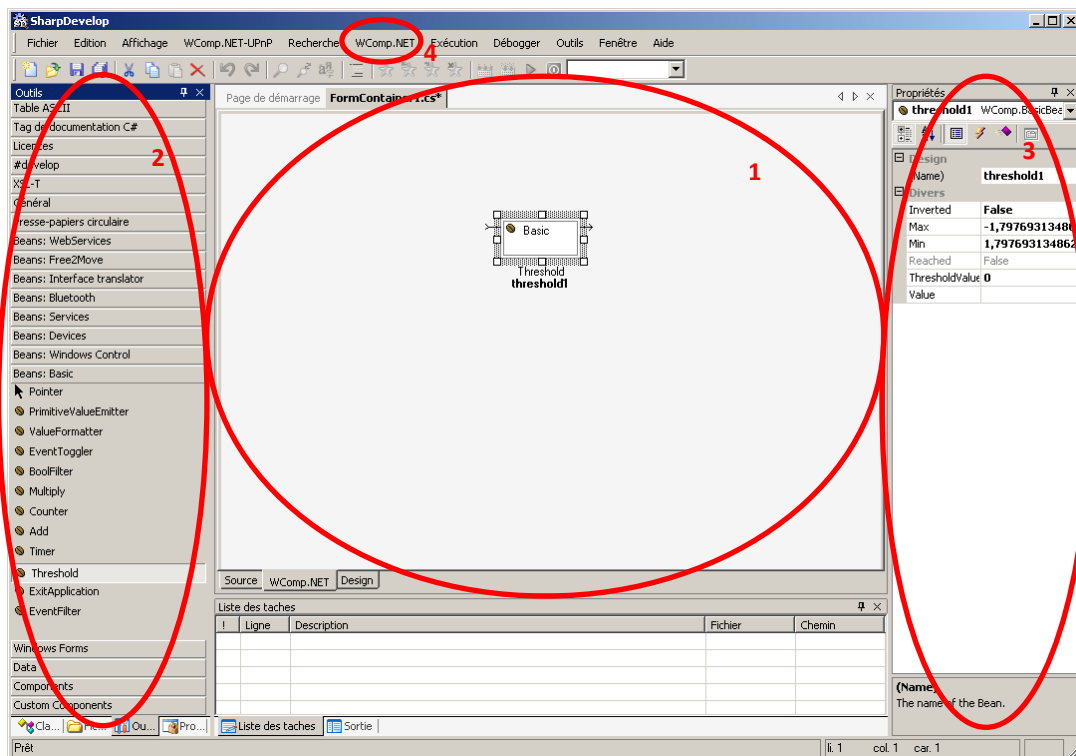


Figure 15: Concepteur SharpWComp

Sur cette vue, quatre parties sont à distinguer :

1. Le **concepteur graphique** : c'est la partie principale où est réalisé l'assemblage des composants constituant l'application.
2. La **liste des composants** mis à votre disposition : cette barre d'outils située sur la partie gauche de l'écran contient la liste des composants qui peuvent être utilisés dans le concepteur graphique. Si l'onglet n'est pas affiché par défaut, il vous faudra le sélectionner, son nom est 'Outils', ou 'Tools' en version anglaise.
3. Le panneau de paramétrage des **propriétés des composants et des liens** : situé dans la partie droite de l'écran, cette boîte à outils présente les propriétés des composants et permet de modifier leurs valeurs. Si besoin, tapez F4 pour l'afficher.
4. le **menu SharpWComp** : ce menu permet entre autres d'importer ou d'exporter un assemblage de composants qui a été réalisé (au format WCC qui permet de sauvegarder l'assemblage indépendamment du code généré).

5.3 CREATION D'ASSEMBLAGES DANS SHARPCOMP

La création d'assemblages dans SharpWComp peut se faire de diverses manières. Dans les sections suivantes, nous détaillerons les opérations manuelles à effectuer pour arriver à créer un assemblage de composants SharpWComp à l'aide du concepteur graphique.

5.3.1 INSTANCIATION DE COMPOSANTS

L'instanciation de composants dans le concepteur SharpWComp s'effectue en 3 temps :

1. Choisir le type de composant que l'on désire créer dans la barre d'outils sur la gauche du concepteur en cliquant sur le nom du composant que l'on souhaite créer.
2. Cliquer dans le concepteur pour créer un composant du type sélectionné.
3. Adapter les propriétés de ce composant en utilisant le panneau de paramétrage sur la droite, une fois l'objet sélectionné.

5.3.2 CREATION D'UN LIEN ENTRE DEUX COMPOSANTS

La création de liens entre composants permet d'effectuer un assemblage et de créer des communications sous forme d'événements entre les composants.

1. Cliquer sur un port de sortie du composant source (port de droite).
2. Relier celui-ci à un port d'entrée d'un composant destination (port de gauche).
3. Une boîte de dialogue apparaît permettant de sélectionner l'événement désiré (voir Figure 16). S'il existe un unique événement pour la source, celui-ci est automatiquement sélectionné.
4. La sélection de la source crée la liste des destinations possibles. Les méthodes compatibles (celles qui ont la même signature, c'est à dire qui reçoivent les mêmes paramètres que l'événement source) sont mises en gras, et les méthodes sans argument en police normale (voir Figure 17).
5. La sélection de la destination (voir Figure 18) autorise à valider cette boîte de dialogue pour créer le lien entre les deux composants.

A noter : pour annuler la création d'un lien en cours alors que la source a été sélectionnée et que la destination ne l'est pas encore, il suffit de faire un clic droit dans le concepteur SharpWComp, ou appuyer sur 'ESC'.

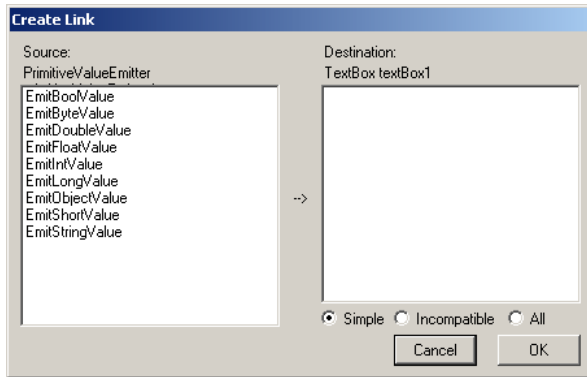


Figure 16: Boîte de dialogue création de lien

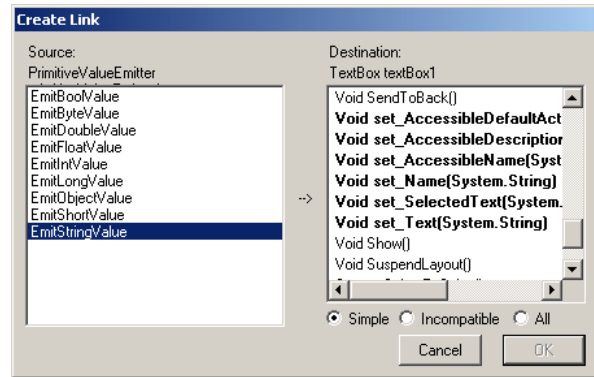


Figure 17: Sélection de la source

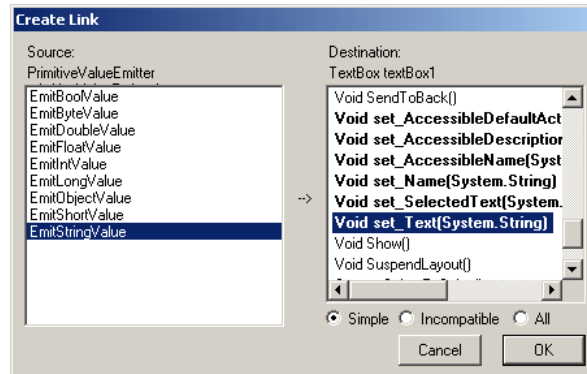


Figure 18: Sélection de la destination

5.3.3 CREATION D'UN LIEN INCOMPATIBLE ENTRE DEUX COMPOSANTS

1. Sélectionner **Incompatible** : les méthodes supplémentaires sont celles qui nécessitent un ou plusieurs arguments de type différents de ceux correspondants à l'événement source sélectionné (voir Figure 19).
2. Une liste des méthodes pour la transformation du type du paramètre est proposée. Sélectionner la méthode adéquate et valider pour effectuer la création du lien (voir Figure 20).

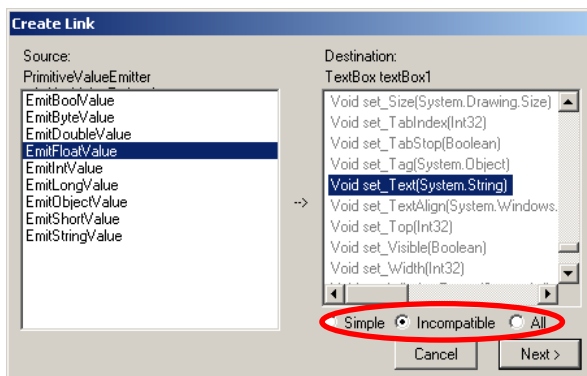


Figure 19: Sélection d'un lien incompatible

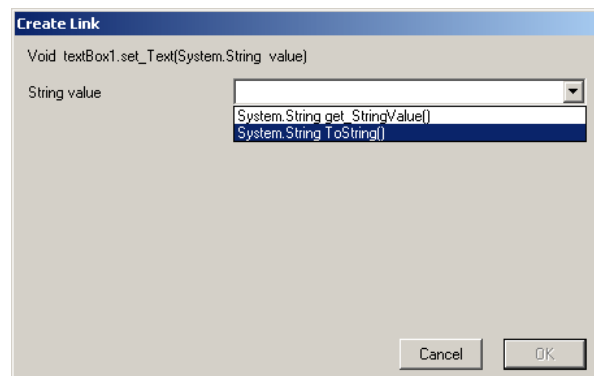


Figure 20: Choix de la méthode de conversion

5.4 SAUVEGARDE ET CHARGEMENT D'UN ASSEMBLAGE

5.4.1 IMPORT D'UN ASSEMBLAGE

L'importation d'un assemblage de composants permet la restauration de composants et des liens qui les unissent.

1. Dans le menu **SharpWComp** sélectionner **Import...** (voir Figure 21).
2. Sélectionner le fichier à charger dans le concepteur SharpWComp (voir Figure 22). Un dossier Exemples dans le répertoire d'installation de SharpDevelop a été ajouté lors de l'installation du module SharpWComp. Son chemin par défaut est C:\Program Files (x86)\SharpDevelop\3.0\Examples\. Celui-ci contient des exemples d'assemblages de composants.

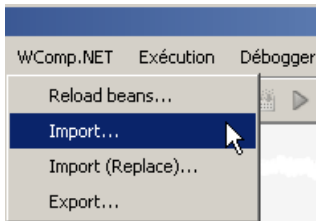


Figure 21: Importation d'un schéma d'assemblage

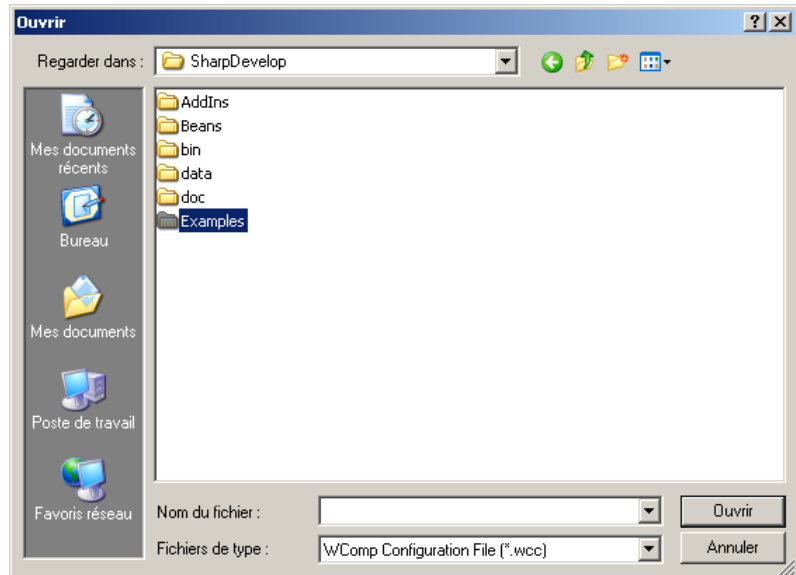


Figure 22: Sélection d'un fichier d'assemblage (wcc)

Après le chargement d'un assemblage, les différents concepteurs sont mis à jour en fonction des composants chargés et des liens qui les unissent (voir Figure 23, Figure 24 et Figure 25).

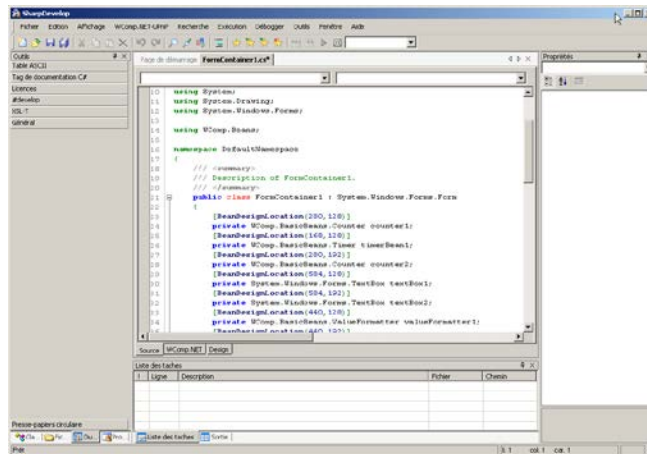


Figure 23: Concepteur Source après importation d'un assemblage (binary-counter-3.wcc)

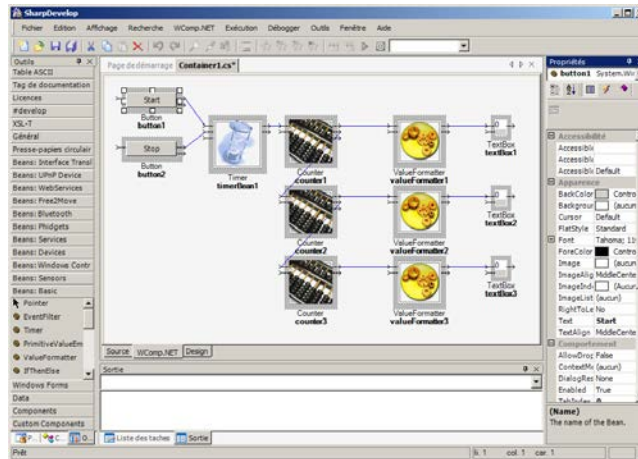


Figure 24: Concepteur SharpWComp après chargement d'un assemblage (binary-counter-3.wcc)

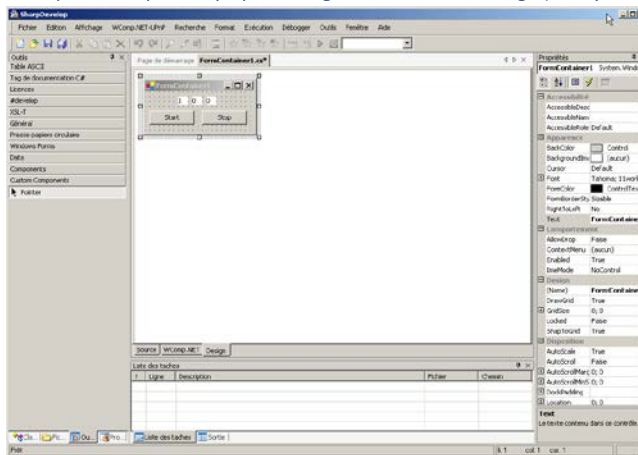


Figure 25: Concepteur Design après chargement d'un assemblage (binary-counter-3.wcc)

5.4.2 EXPORT D'UN ASSEMBLAGE

L'exportation d'un assemblage de composants permet la sauvegarde de composants et des liens qui les unissent. Les données seront sauvegardées dans un fichier au format XML avec l'extension .wcc.

5.4.3 NOTE IMPORTANTE

Attention ! Il est nécessaire d'utiliser l'importation et l'exportation pour sauvegarder les assemblages que vous avez pu construire. Si vous utilisez la fonctionnalité de sauvegarde de SharpDevelop (Fichier / Enregistrer...), vous sauvegarderez le code source C# généré.

SharpWComp travaille avec une transformation de modèle qui permet, à partir de l'assemblage de composants, de générer le code source C# correspondant. Toutefois, la transformation inverse (du code source C# vers l'assemblage de composants) ne garantit pas que toutes les données seront récupérées au rechargement (l'ensemble des données du code source ne sont pas projetables dans le modèle utilisé pour l'assemblage de composants). Il est donc conseillé d'utiliser la procédure d'import / export d'un assemblage dans un fichier WCC.

6 CREATION DE COMPOSANTS SHARPWCOMP

La plate-forme SharpWComp permet à tout utilisateur de créer ses propres composants.

6.1 CREATION D'UNE SOLUTION POUR VOTRE (OU VOS) COMPOSANT(S)

Tout d'abord, il est nécessaire de créer une solution qui va nous permettre de compiler le code de votre ou vos composants (il est possible de mettre le code de plusieurs composants dans la même librairie dll).

1. Dans le menu **Fichier**, sélectionner **Nouveau** et cliquer sur **Solution...** (ou **Ctrl+Shit+N**)
2. Donner un nom à la solution pour la création de votre composant
3. Dans la partie **Références** de la solution, il est nécessaire de supprimer la référence à Beans pour la remplacer par la librairie Beans de votre installation (C:\Program Files\SharpDevelop\3.0\Beans\Beans.dll)

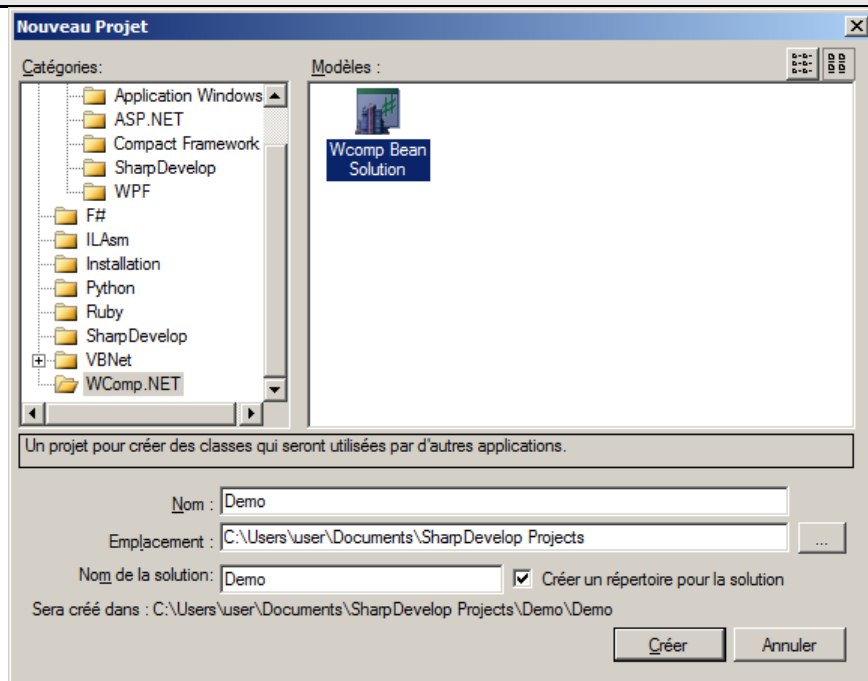


Figure 26: Création d'une solution pour la création d'un composant

6.2 CODE D'UN COMPOSANT A PARTIR D'UN MODELE

Il est nécessaire de respecter le modèle de composant de WComp. SharpDevelop autorisant la création de modèles de code source, nous fournissons un modèle de code pour la création de composants SharpWComp.

1. Dans le menu **Fichier**, sélectionner **Nouveau** et cliquer sur **Fichier...** (ou **Ctrl+N**) (voir Figure 11) ou cliquer sur l'icône **Nouveau Fichier** (voir Figure 12)
2. Sélectionner la catégorie **WComp.NET** dans la boîte de dialogue qui s'est ouverte.
3. Cliquer sur le modèle **C# Bean** (voir Figure 27).

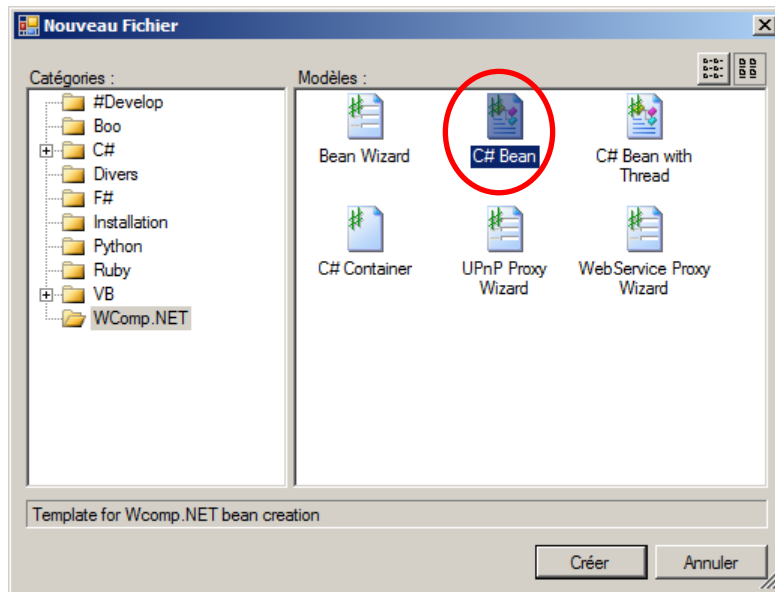


Figure 27: Création d'un composant SharpWComp

Le concepteur Source contiendra automatiquement un code source exemple pour la création d'un composant SharpWComp. Voici le modèle de code d'un composant SharpWComp.

```

using System;
using WComp.Beans;

namespace WComp.Beans
{
    /// <summary>
    /// This is a sample bean, which has an integer evented property and a method.
    ///
    /// Notes: for beans creating threads, the IThreadCreator interface should be
    /// implemented, providing a cleanup method should be implemented and named `Stop()`.
    /// For proxy beans, the IProxyBean interface should be implemented,
    /// providing the IsConnected property, allowing the connection status to be drawn in
    /// the AddIn's graphical designer.
    ///
    /// Several classes can be defined or used by a Bean, but only the class with the
    /// [Bean] attribute will be available in WComp. Its ports will be all public methods,
    /// events and properties defined in that class.
    /// </summary>
    [Bean(Category="MyCategory")]
    public class Bean1
    {
        /// <summary>
        /// Fill in private attributes here.
        /// </summary>
        private int property;

        /// <summary>
        /// This property will appear in bean's property panel and bean's input functions.
        /// </summary>
        public int MyProperty {
            get { return property; }
            set {
                property = value;
                FireIntEvent(property); // event will be fired for every property set.
            }
        }

        /// <summary>
        /// A method sending an event, which is here simply the argument + 1.
        /// Note that there is no return type to the method, because we use events to send
        /// information in WComp. Return values don't have to be used.
        /// </summary>
        public void MyMethod(int arg) {
            FireIntEvent(arg+1);
        }

        /// <summary>
        /// Here are the delegate and his event.
        /// A function checking nullity should be used to fire events (like FireIntEvent).
        /// </summary>
        public delegate void IntValueEventHandler(int val);
        /// <summary>
        /// the following declaration is the event by itself. Its name, here
        "PropertyChanged",
        /// is the name of the event as it will be displayed in the bean type's interface.
        /// </summary>
        public event IntValueEventHandler PropertyChanged;

        private void FireIntEvent(int i) {
            if (PropertyChanged != null)
                PropertyChanged(i);
        }
    }
}

```

Il est alors nécessaire au programmeur d'adapter ce code aux besoins et aux fonctionnalités souhaitées.

1. Compiler le code à l'aide d'une solution SharpDevelop générant un assembly (un fichier dll est généré correspondant à votre ou vos composant(s) créés).

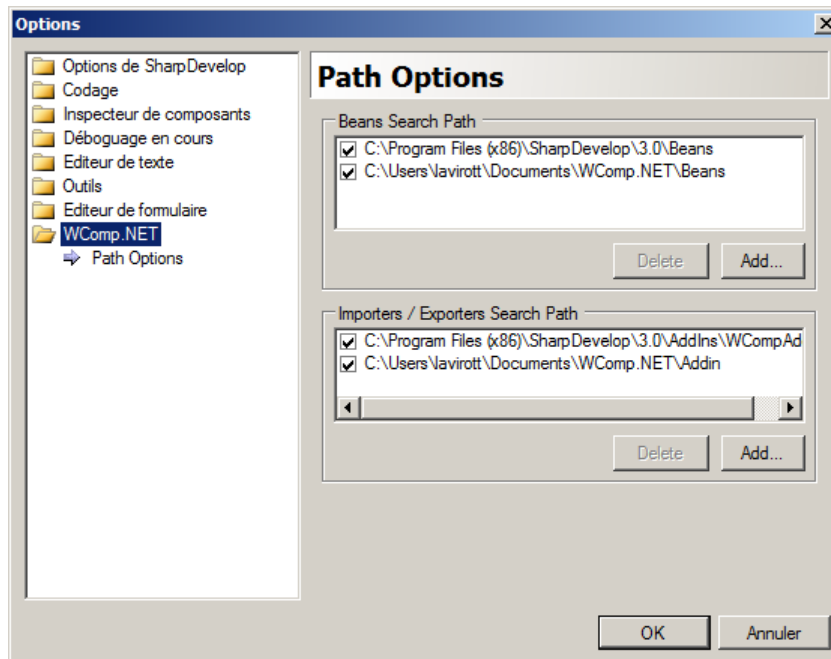
De plus amples informations sur la création de composants Bean peuvent être trouvées à l'adresse suivante :

<http://www.wcomp.fr/comPCODE>

6.3 RENDRE CE COMPOSANT UTILISABLE DANS SHARPCOMP

Depuis la version 3.2.1.1367, l'utilisateur peut ajouter ses propres composants dans un dossier par défaut de son répertoire personnel (qui est créé lors de l'installation de SharpWComp). Ce dossier par défaut est « Documents\WComp.NET\Beans ». Vous pouvez aussi modifier l'endroit où sont stockés les composants avec lesquels vous souhaitez travailler.

1. Dans le menu **Outils**, sélectionner **Options puis** sélectionner l'option WComp .NET.
2. Il est possible d'ajouter ou de retirer des chemins où trouver les composants de travail (partie supérieure de la boîte de dialogue, section **Beans Search Path**).



3. Pour utiliser un composant, il suffit de copier la dll générée dans un des dossiers Beans de SharpDevelop (ex : C:\Users\user\WComp.NET\Beans). Ce répertoire a été automatiquement ajouté à l'installation de SharpWComp.

Attention ! Il est impératif de ne pas avoir deux assemblies (donc deux dll) correspondant au même composant. Cela provoquera une erreur au chargement du concepteur SharpWComp.

6.4 ICONE D'UN COMPOSANT DANS L'ASSEMBLAGE

Il est possible de modifier la représentation graphique du composant dans l'assemblage et en particulier d'associer un icône au composant développé (où existant). Pour associer l'icône à un type de composants, il suffit de :

1. Créer un dossier icons à l'endroit où se trouve la dll contenant le Bean développé (ex : C:\Program Files (x86)\SharpDevelop\3.0\Beans\icons).
2. Ajouter un fichier image (par exemple png) avec : soit avec le nom de l'assembly (pour faire un affichage par défaut pour tout Bean de l'assembly), soit avec le nom de la classe préfixé par le namespace (ex. WComp.Beans.Bean1.png). La définition généralement utilisée est 128x128 pixels.

Il est aussi possible d'ajouter des icônes aux Beans qui ont été générés par le Wizard UPnP2Bean présenté à la section 7.

6.5 MISE AU POINT DU CODE DE VOTRE COMPOSANT

6.5.1 DEVELOPPEMENT D'UN COMPOSANT ET TESTS UNITAIRES

La mise au point d'un composant est essentielle au bon fonctionnement de votre assemblage de composant. Chaque composant doit donc être intensivement testé indépendamment des autres composants que vous pourriez utiliser dans un assemblage. La méthode conseillée est bien entendu la création de tests unitaires pour valider les différentes méthodes, propriétés et événements de votre composant. L'utilisation de NUnit (<http://www.nunit.org/>) est recommandée, cet outil s'intégrant à l'environnement SharpDevelop (<http://community.sharpdevelop.net/blogs/mattward/articles/FeatureTourUnitTesting.aspx>).

6.5.2 MISE AU POINT A L'AIDE DU DEBOGUEUR

Une fois votre composant développé et testé, il est parfois nécessaire de finaliser sa mise au point lorsque celui-ci est intégré dans un assemblage de composants dans un container WComp. Pour réaliser cette étape, il faut procéder aux réglages suivant sur le projet du ou des composants (plusieurs composants pouvant se trouver dans un même assembly) que vous souhaitez mettre au point.

1. Aller dans les propriétés du projet de votre composant (par exemple le projet Demo comme expliqué dans les sections 6.1 et 6.2) en faisant **Clic droit / Propriétés** sur le nom du projet.
2. Dans l'onglet **Compiler**, régler la **Configuration** des propriétés pour le mode **Debug**
3. Modifier la propriété « Chemin de sortie » pour que celle-ci spécifie le dossier contenant les Beans (soit le dossier « C:\Program Files\SharpDevelop\3.0\Beans », soit votre dossier personnel contenant les Beans « Mes Documents\WComp.NET\Beans ». La dll sera ainsi directement générée dans le dossier où sont les Beans de WComp.
4. Dans l'onglet **Débugage**, modifier la propriété « Lancer programme externe » pour spécifier l'emplacement de l'application SharpDevelop correspondant au lancement de WComp : (C:\Program Files\SharpDevelop\3.0\bin\SharpDevelop.exe)
5. Dans le code source de votre composant, vous pouvez ajouter un Break Point pour le retour aux sources lors de l'exécution de votre composant dans le container.

7 UPNP2BEAN

UPnP2Bean est un Wizard permettant de générer automatiquement un composant pour l'environnement SharpWComp à partir d'un serveur UPnP. SharpWComp est alors considéré comme un Control Point dans la terminologie UPnP. Un assemblage pourra être constitué de composants logiciels que vous avez programmés et de composants générés à partir de dispositifs ou service UPnP ou bien encore de WebServices (voir section 8 « WSDL2Bean », page 21).

7.1 UPNP ET SHARPCOMP

UPnP signifie Universal Plug and Play. Cette technologie vise à fournir un protocole de communications inter-périphériques. Il est aussi parfois appelé : Web service pour dispositifs.

7.2 INSTALLATION DU MODULE UPNP2BEAN

L'installation du module UPnP2Bean s'effectue de manière identique à l'installation décrite dans la section 4, page 6. Vous pouvez vous reporter à cette section si vous rencontrez des problèmes d'installation de ce module.

7.3 UTILISATION DE UPNP2BEAN

1. Dans le menu **Fichier**, sélectionner **Nouveau** et cliquer sur **Fichier...** (ou **Ctrl+N**) (voir Figure 11) ou Cliquer sur l'icône **Nouveau Fichier** (voir Figure 12)
2. Sélectionner la catégorie **SharpWComp** dans la boîte de dialogue qui s'est ouverte.
3. Cliquer sur le modèle **UPnP Device WebService Proxy** (voir Figure 28).

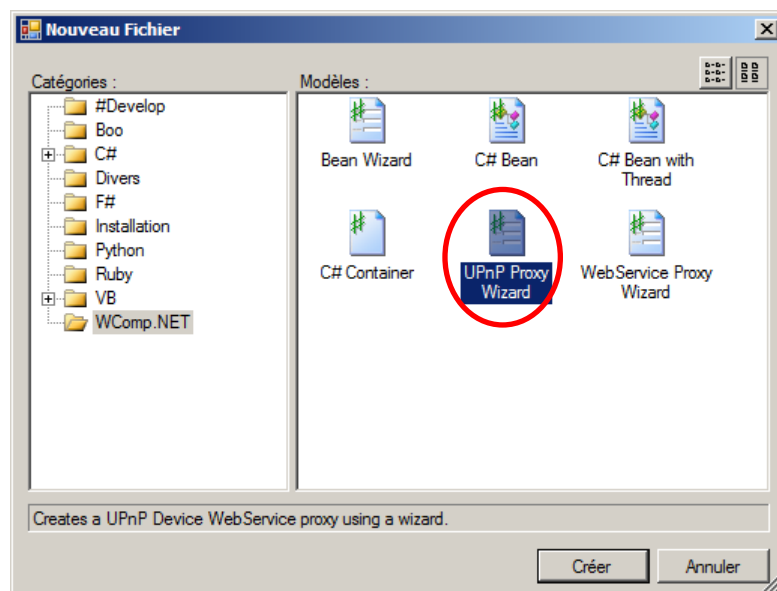


Figure 28: Lancement du Wizard UPnP2Bean

Nous voyons un premier panneau contenant l'arbre graphique de nos dispositifs UPnP présents sur le réseau local. Il est possible d'explorer un dispositif pour en connaître les services proposés. Une case à cocher permet de valider ou invalider la génération finale d'un événement, d'une méthode, de tout un service ou voir même tout un dispositif. Sur cet exemple le dispositif "Lumière (TWEEDY)" contient deux services "DimmingService.0001" et "SwitchPower.0001". Ce dernier donne accès à deux méthodes "GetStatus" et "SetTraget", et une variable d'état à évènement "Status".

4. Sélectionner le ou les dispositifs et la ou les méthodes ou variables pour lesquels vous souhaitez générer un composant (voir Figure 29).

Un second panneau affiche une liste sommaire des principales propriétés de l'élément sélectionné par l'utilisateur.

5. Configurer la génération de code : vous pouvez modifier le nom du dispositif (qui sera le nom du composant WComp), modifier le namespace, montrer le code source généré, convertir l'URL en une adresse locale pour une utilisation simplifiée en phase de tests locaux, et compiler et placer la dll générée dans les Beans SharpDevelop (voir Figure 30).
6. Dans le cas où l'option « *Compil and Copy Bean (dll) to Beans directory* », le code généré est automatiquement compilé et la dll est déplacée dans le dossier contenant les Beans WComp.

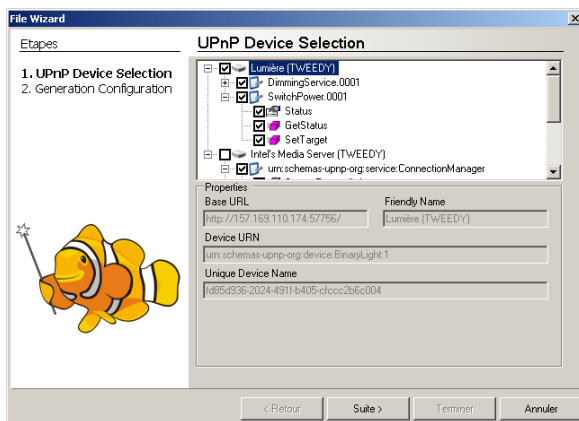


Figure 29: Sélection des dispositifs UPnP

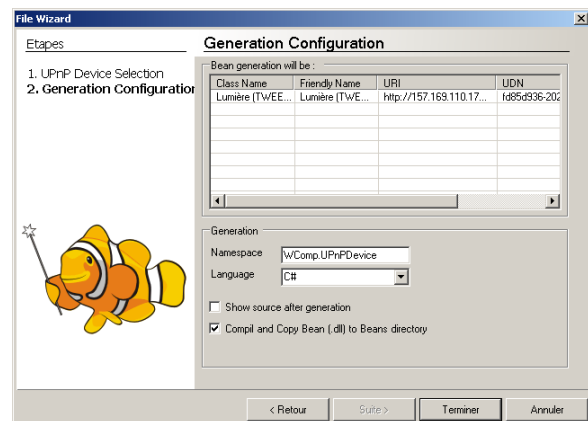


Figure 30: Configuration de la Génération

Les composants ainsi générés sont directement utilisables dans le designer graphique WComp et manipulables de la même manière.

Attention ! Pour utiliser un composant correspondant à un dispositif, il est nécessaire de vérifier dans les propriétés que l'adresse IP et le port soient corrects pour l'instance du dispositif (les valeurs par défaut stockées dans le composant sont celles qui étaient définies lors de la génération du composant).

8 WSDL2BEAN

WSDL2Bean est un Wizard permettant de générer automatiquement un composant pour l'environnement Wcomp.NET à partir d'un Webservice. Wcomp.NET permet alors d'envoyer des requêtes au Webservice et de récupérer les informations retournées par celui-ci.

8.1 WEBSERVICES ET WCOMP.NET

Un Web Service est un ensemble de fonctionnalités accessible via un réseau par différentes applications. Dans WComp, le Wizard permet de générer un composant proxy du Web Service.

8.2 INSTALLATION DU MODULE WSDL2BEAN

L'installation du module WSDL2Bean s'effectue de manière identique à l'installation décrite dans la section 4, page 6. Vous pouvez vous reporter à cette section si vous rencontrez des problèmes d'installation de ce module.

8.3 UTILISATION DE WSDL2BEAN

1. Dans le menu **Fichier**, sélectionner **Nouveau** et cliquer sur **Fichier...** (ou **Ctrl+N**) (voir Figure 11). ou cliquer sur l'icône **Nouveau Fichier** (voir Figure 12).
2. Sélectionner la catégorie **Wcomp.NET** dans la boîte de dialogue qui s'est ouverte.
3. Cliquer sur le modèle **WebService Proxy Wizard** (voir Figure 31).

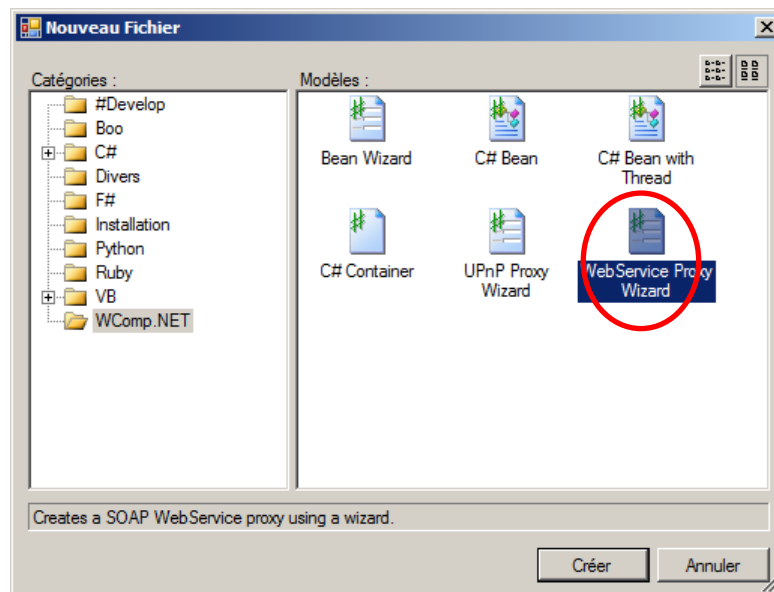


Figure 31: Lancement du Wizard WSDL2Bean

4. Fournir une URL ou un fichier local contenant la description du Webservice (fichier .wsdl). Remplir le champ namespace qui définira le nom du composant que vous allez créer (voir Figure 32).

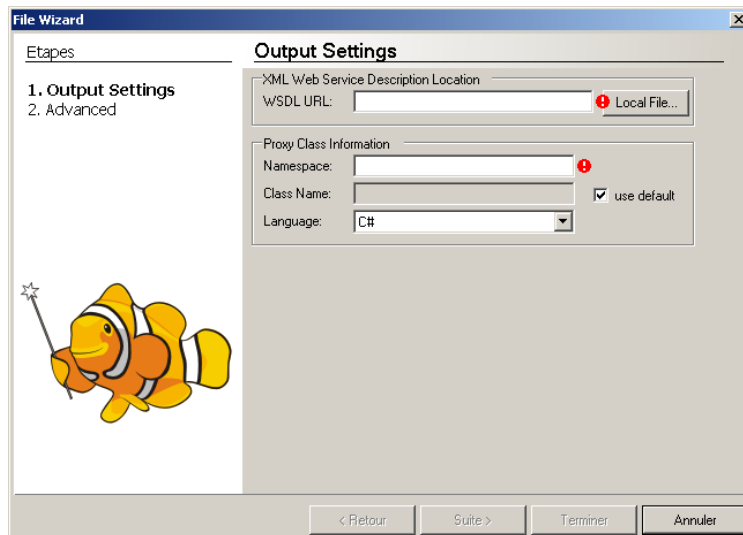


Figure 32 : Sélection du Webservice

5. Configurer la génération de code : vous pouvez sélectionner la ou les méthodes pour lesquels vous souhaitez générer un composant. Deux affichages sont proposés, la première est une vue des méthodes du Webservice sous forme d'arbre, et une seconde sous forme de tableau (voir Figure 33 et Figure 34).
6. Dans le cas où l'option « *Compile and Copy Bean (dll) to Beans directory* » est validée, le code généré est automatiquement compilé et la dll est déplacée dans le dossier contenant les Beans Wcomp.

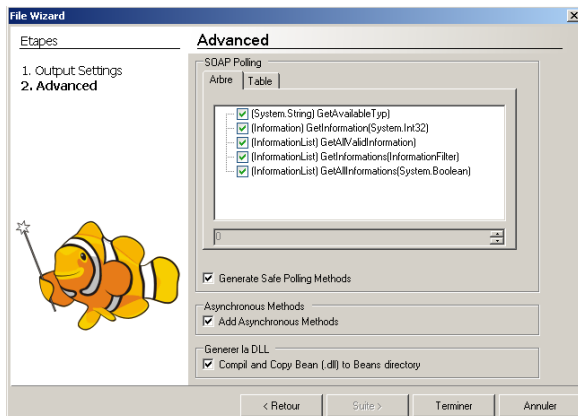


Figure 33 : Affichage sous forme d'arbre

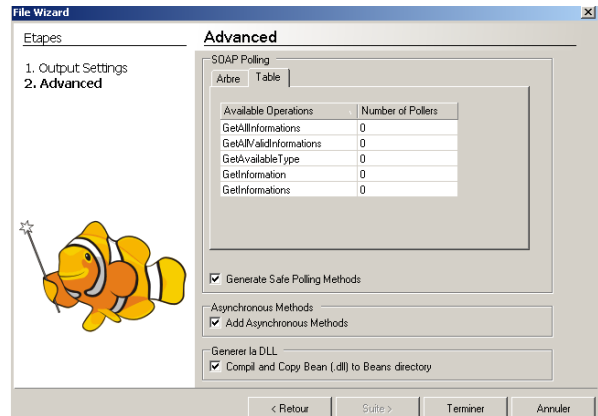


Figure 34: Affichage sous forme de tableau

7. Le code source correspondant à l'accès au Webservice est alors généré (voir Figure 35).
8. Dans le cas où l'option « *Compile and Copy Bean (dll) to Beans directory* » n'est pas validée, répéter l'opération d'écriture dans la section 6, page 16.

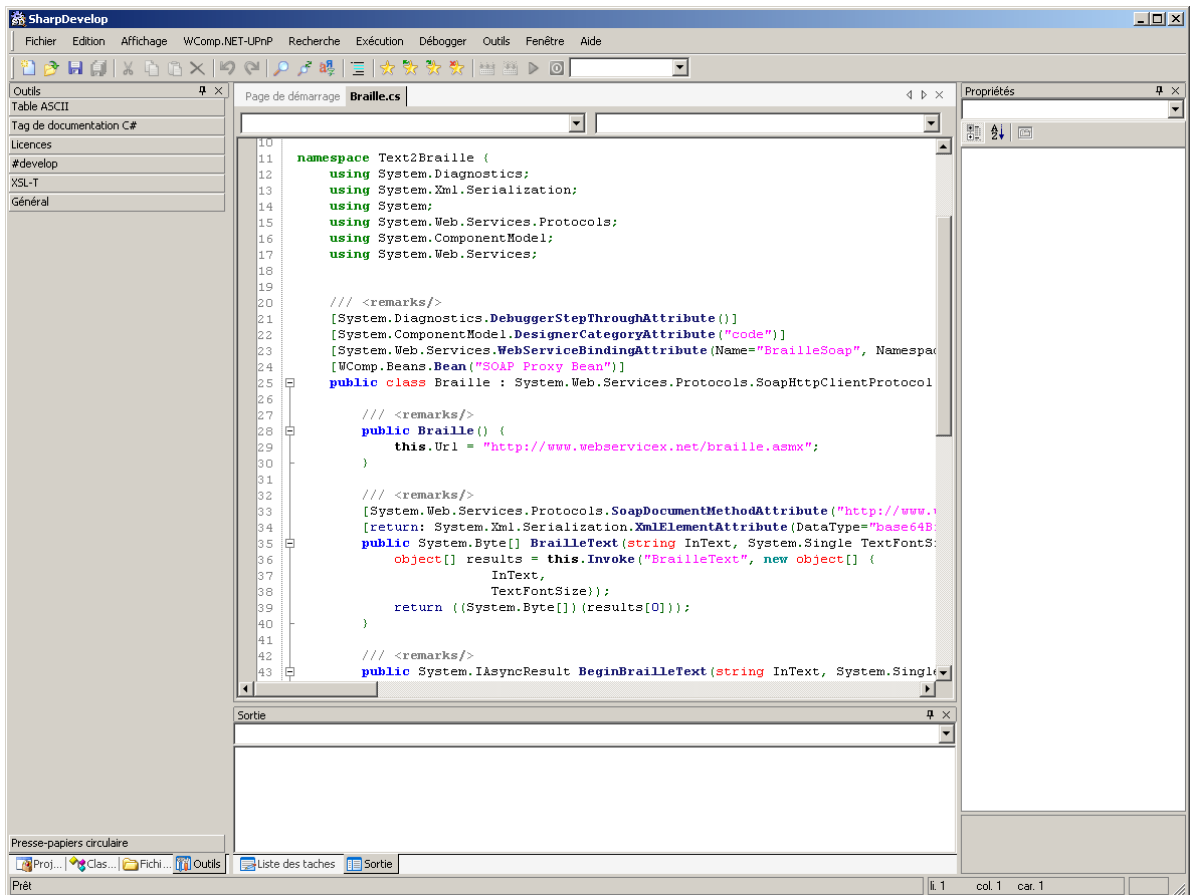


Figure 35: Code source d'un composant WebService généré