



# SharpWComp 2.4



*SharpWComp fait partie du projet WComp. Celui est subdivisé en plusieurs modules. Ce document présente les étapes nécessaires pour l'installation des modules WComp qui ont été conçu comme une extension à SharpDevelop. Ce document détaille les différentes étapes d'installations nécessaires pour l'utilisation de SharpWComp (installation de l'environnement .NET, installation de SharpDevelop, installation de modules complémentaires de SharpWComp, etc.). Un tutorial sur l'utilisation de SharpWComp et sur les différents modules qui le compose 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)  
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 Lavirotte<sup>1</sup>, Gaëtan Rey<sup>2</sup> et Jean-Yves Tigli<sup>3</sup>  
Enseignants-Chercheurs  
Membres de l'équipe de recherche Rainbow  
dirigée par Michel Riveill, laboratoire I3S  
<http://rainbow.i3s.unice.fr/>*

---

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

<sup>2</sup> Université de Nice – Sophia Antipolis / IUT

<sup>3</sup> Université de Nice – Sophia Antipolis / Polytech'Nice Sophia

# Table des matières

I. Avant propos .....	1
II. Installation de l'environnement .NET .....	2
III. Installation de SharpDevelop .....	3
IV. Installation de SharpWComp, Addin SharpDevelop .....	4
V. Utilisation de SharpWComp, Addin SharpDevelop.....	5
1. Création d'un nouveau fichier SharpWComp .....	5
2. Présentation du concepteur graphique SharpWComp.....	6
3. Création d'assemblages dans SharpWComp.....	7
3.1. Instanciation de composants.....	7
3.2. Création d'un lien entre deux composants .....	7
3.3. Création d'un lien incompatible entre deux composants .....	8
4. Sauvegarde et Chargement d'un assemblage .....	9
4.1. Import d'un assemblage .....	9
4.2. Export d'un assemblage .....	10
4.3. Note importante .....	10
VI. Création de composants SharpWComp .....	11
1. Code d'un composant à partir d'un modèle .....	11
2. Rendre ce composant utilisable dans SharpWComp.....	12
3. Icône d'un composant dans l'assemblage .....	13
VII. UPnP2Bean.....	14
1. UPnP et SharpWComp.....	14
2. Installation du module UPnP2Bean .....	14
3. Utilisation de UPnP2Bean.....	14
VIII. WSDL2Bean .....	16
1. WebServices et Wcomp.NET .....	16
2. Installation du module WSDL2Bean .....	16
3. Utilisation de WSDL2Bean.....	16

# Table des illustrations

Figure 1: <i>Installation de SharpDevelop 1.0.2a</i> .....	3
Figure 2: <i>Accord de Licence</i> .....	3
Figure 3: <i>Choix de l'emplacement d'installation</i> .....	3
Figure 4: <i>Copie des fichiers</i> .....	3
Figure 5: <i>Fin d'installation et lancement de l'application</i> .....	<b>Erreur ! Signet non défini.</b>
Figure 6: <i>Complétion de code : création d'une nouvelle base</i> .....	<b>Erreur ! Signet non défini.</b>
Figure 7: <i>Complétion de code: choix de l'emplacement</i> .....	<b>Erreur ! Signet non défini.</b>
Figure 8: <i>Complétion de code: création de la base</i> .....	<b>Erreur ! Signet non défini.</b>
Figure 9: <i>Complétion de code: terminer</i> .....	<b>Erreur ! Signet non défini.</b>
Figure 10: <i>Installation de SharpWComp, Addin SharpDevelop</i> .....	4
Figure 11: <i>Accord de licence</i> .....	4
Figure 12: <i>Détection automatique de l'installation de SharpDevelop</i> .....	4
Figure 13: <i>Validation des paramètres d'installation</i> .....	4
Figure 14: <i>Menu Nouveau fichier</i> .....	5
Figure 15: <i>Icône nouveau fichier</i> .....	5
Figure 16: <i>Création d'un C# Container pour activer SharpWComp</i> .....	5
Figure 17: <i>Accès aux différents concepteurs (Source, SharpWComp, Design)</i> .....	6
Figure 18: <i>Concepteur SharpWComp</i> .....	6
Figure 19: <i>Boîte de dialogue création de lien</i> .....	8
Figure 20: <i>Sélection de la source</i> .....	8
Figure 21: <i>Sélection de la destination</i> .....	8
Figure 22: <i>Sélection d'un lien incompatible</i> .....	8
Figure 23: <i>Choix de la méthode de conversion</i> .....	8
Figure 24: <i>Importation d'un schéma d'assemblage</i> .....	9
Figure 25: <i>Sélection d'un fichier d'assemblage (wcc)</i> .....	9
Figure 26: <i>Concepteur Source après importation d'un assemblage (binary-counter-3.wcc)</i> ....	9
Figure 27: <i>Concepteur SharpWComp après chargement d'un assemblage (binary-counter-3.wcc)</i> .....	10
Figure 28: <i>Concepteur Design après chargement d'un assemblage (binary-counter-3.wcc)</i> ..	10
Figure 29: <i>Création d'un composant SharpWComp</i> .....	11
Figure 30: <i>Lancement du Wizard UPnP2Bean</i> .....	14
Figure 31: <i>Sélection des dispositifs UPnP</i> .....	15
Figure 32: <i>Configuration de la Génération</i> .....	15
Figure 33: <i>Lancement du Wizard WSDL2Bean</i> .....	16
Figure 34 : <i>Sélection du Webservice</i> .....	17
Figure 35 : <i>Affichage sous forme d'arbre</i> .....	17
Figure 36: <i>Affichage sous forme de tableau</i> .....	17
Figure 37: <i>Code source d'un composant Webservice généré</i> .....	18

## I. Avant propos

Le projet WComp est parti d'une idée originale d'étudiants et d'enseignants chercheurs 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 recherche-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 transmettrons régulièrement une référence bibliographique actualisée<sup>4</sup> 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, Gaëtan.Rey@unice.fr, Jean-Yves.Tigli@unice.fr

**Subject :** [WComp User] ...

Bonne utilisation !

---

<sup>4</sup> **D. Cheung, J-Y. Tigli, S. Lavirotte and M. Riveill**, « WComp: a Multi-Design Approach for Prototyping Applications using Heterogeneous Resources », in the *Proceedings of 17th IEEE International Workshop on Rapid System Prototyping (RSP)*, June 14-16, 2006, Chania, Crete.

## II. Installation de l'environnement .NET

Pour procéder à l'installation de SharpDevelop, il est nécessaire de disposer de l'environnement de programmation .NET. La version courante utilisée pour SharpDevelop 2.2.1.2648 est l'environnement .NET 2.0.

1. Télécharger  
<http://www.microsoft.com/downloads/en/details.aspx?FamilyID=0856eacb-4362-4b0d-8edd-aab15c5e04f5&displaylang=en>
2. Lancer l'installation du fichier dotnetfx.exe pour disposer de l'environnement .NET 2.0
3. Vous disposez maintenant de l'environnement .NET nécessaire à l'étape suivante : l'installation de SharpDevelop.

### III. Installation de SharpDevelop

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 2.2.1.2648 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%202.x/2.2.1/SharpDevelop 2.2.1.2648 Setup.msi/download](http://sourceforge.net/projects/sharpdevelop/files/SharpDevelop%202.x/2.2.1/SharpDevelop%202.2.1.2648%20Setup.msi/download)
2. Double cliquer sur l'exécutable téléchargé pour lancer l'installation.
3. Suivre les 5 étapes d'installation : Bienvenue, Accord de licence, Choix de l'emplacement d'installation, Copie des fichiers, Fin d'installation et lancement de l'application (voir figures ci-dessous).

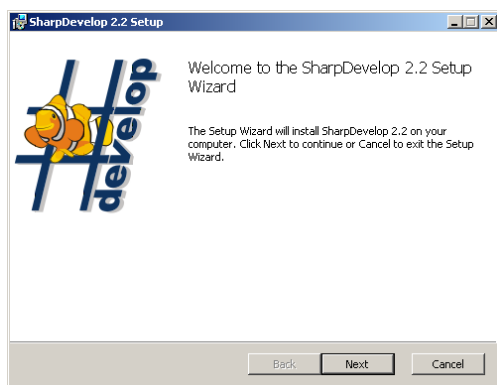


Figure 1: Installation de SharpDevelop 2.2.1.2648

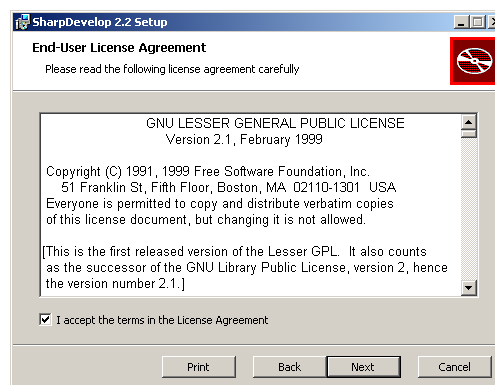


Figure 2: Accord de Licence

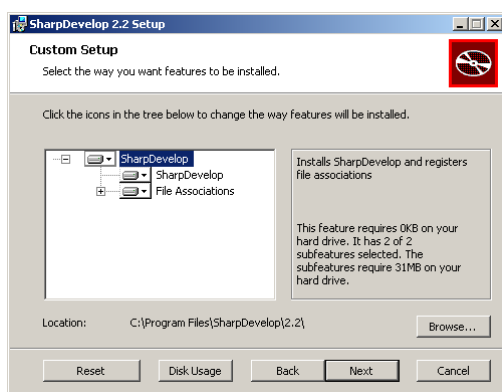


Figure 3: Choix de l'emplacement d'installation

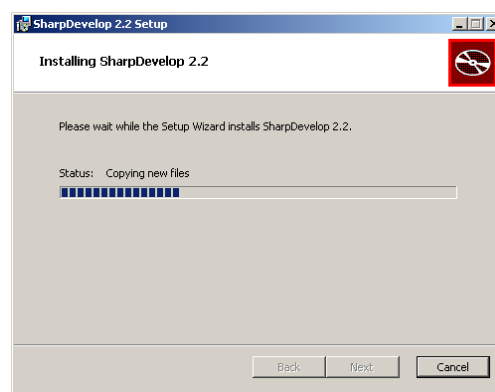


Figure 4: Copie des fichiers

SharpDevelop est maintenant installé sur votre ordinateur.

## IV. Installation de SharpWComp, Addin SharpDevelop

L'installation de SharpWComp s'effectue grâce à une application auto-extractible. 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. Valider l'écran de bienvenue (voir Figure 5).
2. Valider la licence de SharpWComp, Addin SharpDevelop (voir Figure 6).
3. Vérifier que le chemin d'installation de SharpWComp, Addin SharpDevelop est bien le dossier dans lequel SharpDevelop a été installé. La détection de l'installation de SharpDevelop est automatique (voir Figure 7).
4. Valider les paramètres pour que la copie des fichiers puisse s'effectuer (voir Figure 8).
5. Valider la fin d'installation.

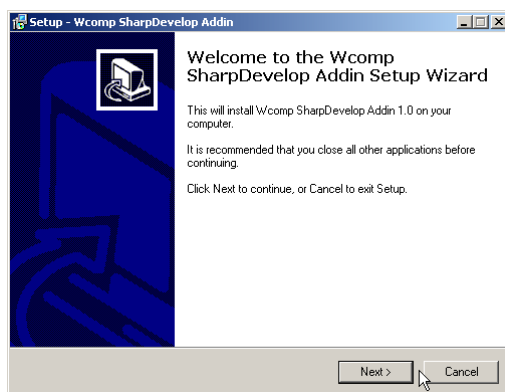


Figure 5: Installation de SharpWComp, Addin SharpDevelop

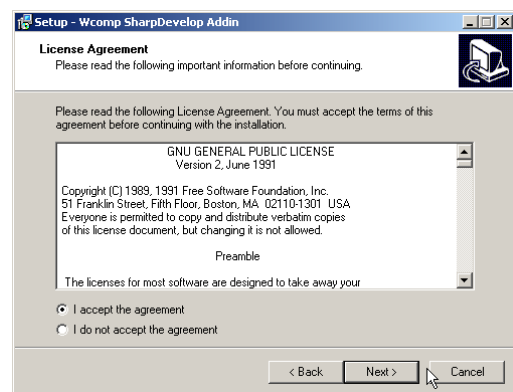


Figure 6: Accord de licence

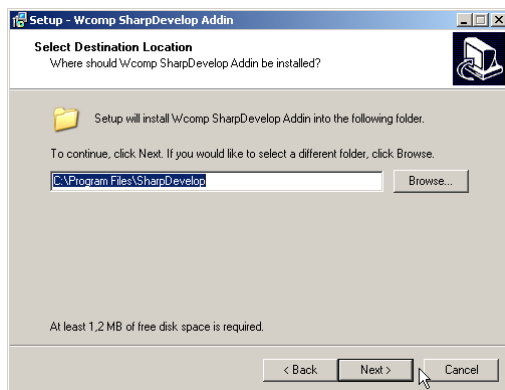


Figure 7: Détection automatique de l'installation de SharpDevelop

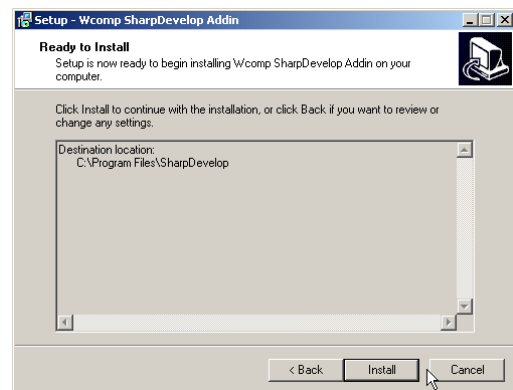


Figure 8: Validation des paramètres d'installation



## V. Utilisation de SharpWComp, Addin SharpDevelop

### 1. Création 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 à 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 9),
2. ou Cliquer sur l'icône **Nouveau Fichier** (voir Figure 10)

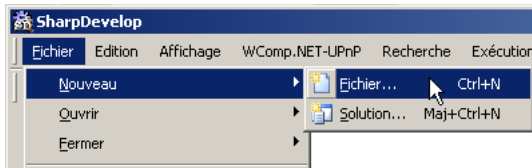


Figure 9: Menu Nouveau fichier



Figure 10: Icône nouveau fichier

3. Sélectionner la catégorie **SharpWComp** dans la boîte de dialogue qui s'est ouverte.
4. Cliquer sur le modèle **C# Container** (voir Figure 11).

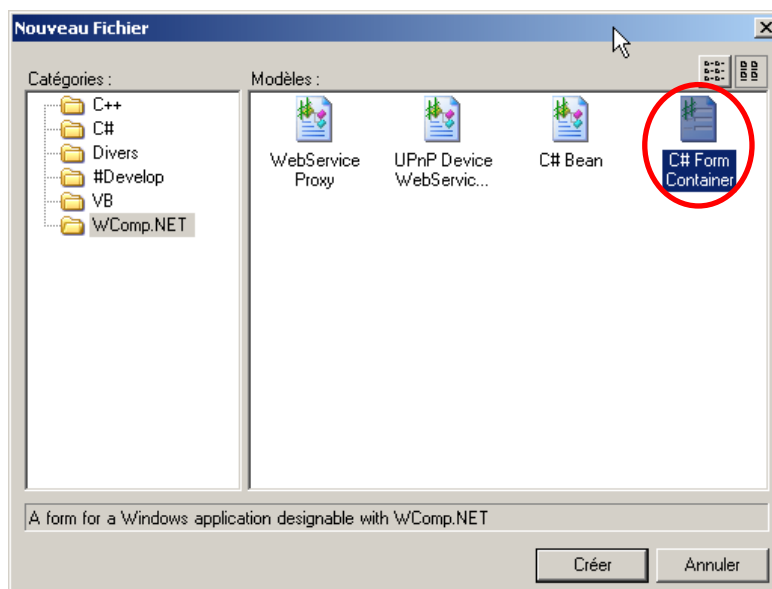


Figure 11: Création d'un C# Container pour activer SharpWComp

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 12)

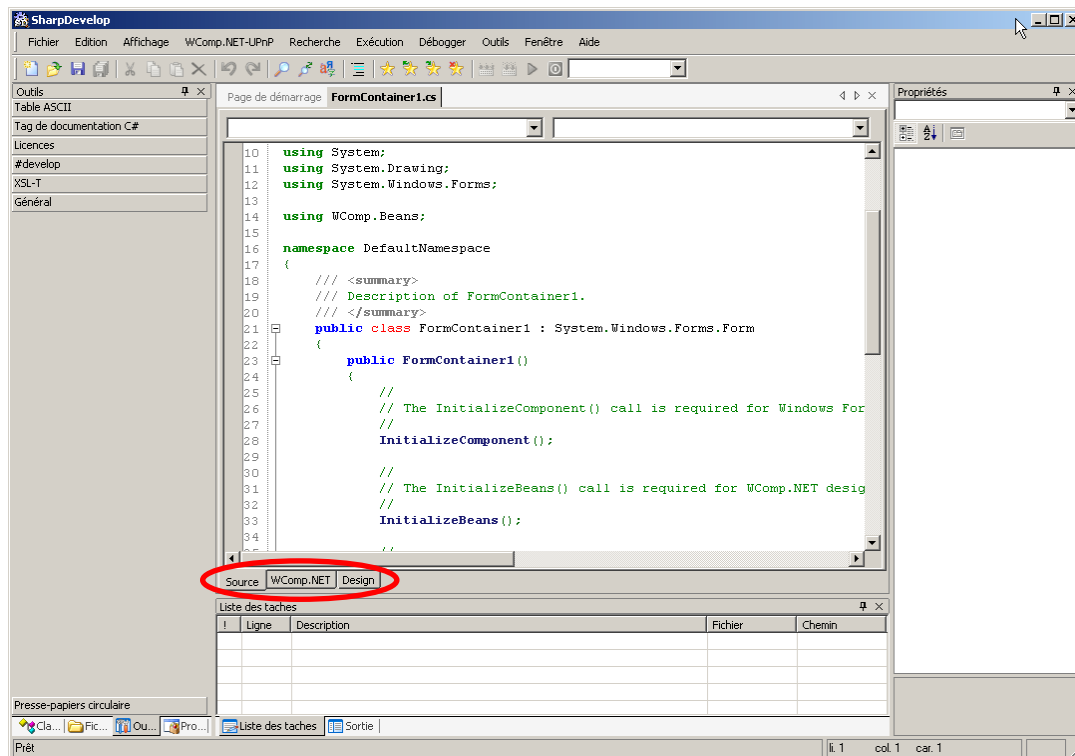


Figure 12: Accès aux différents concepteurs (Source, SharpWComp, Design)

## 2. Présentation du concepteur graphique SharpWComp

Pour travailler avec le concepteur SharpWComp, vous devez sélectionner l'onglet SharpWComp en bas de la vue principale (voir Figure 12). Vous obtiendrez alors la vue ci-dessous (voir Figure 13).

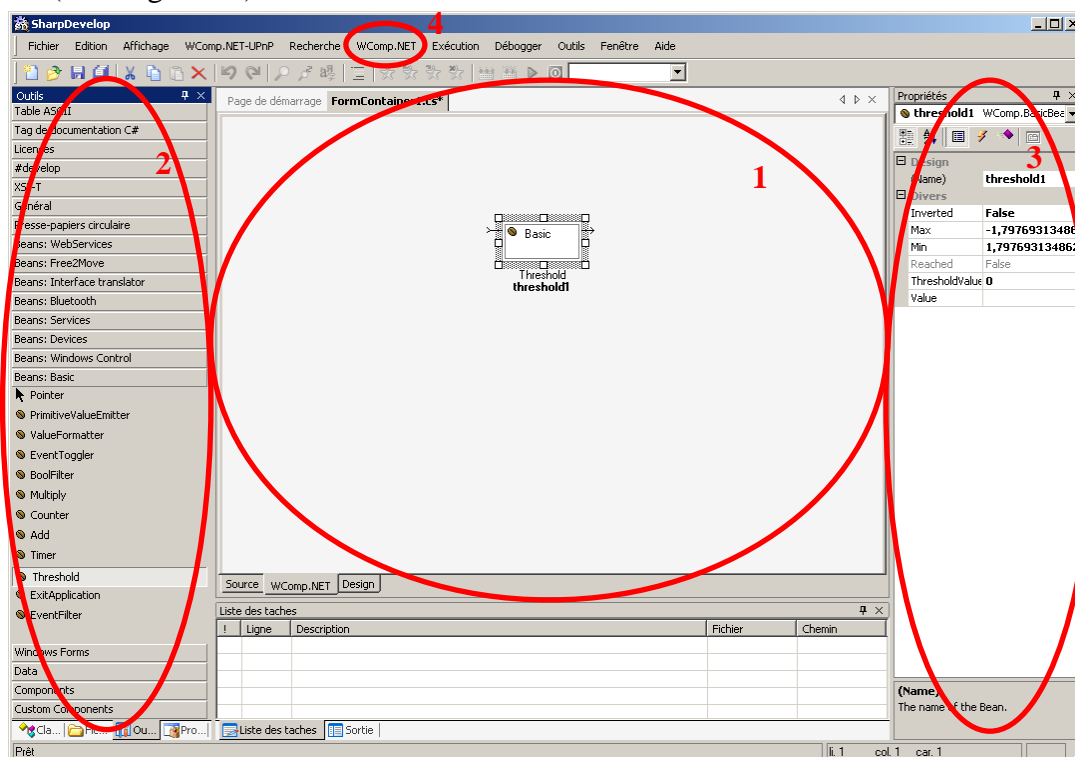


Figure 13: Concepteur SharpWComp

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.
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.
4. le **menu SharpWComp** : ce menu permet d'importer ou d'exporter un assemblage de composants qui a été réalisé (format wcc qui permet de sauvegarder l'assemblage indépendamment du code généré).

### 3. Création d'assemblages dans SharpWComp

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.

#### 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 la panneau de paramétrage sur la droite, une fois l'objet sélectionné.

#### 3.2. Création d'un lien entre deux composants

La création de lien 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 14). 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 les événements qui reçoivent les mêmes paramètres que l'événement source) sont mises en gras, et les méthodes sans argument en police normale (voir Figure 15).
5. La sélection de la destination (voir Figure 16) 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.

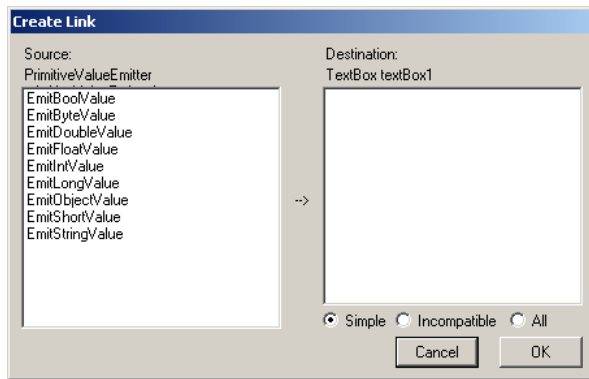


Figure 14: Boîte de dialogue création de lien

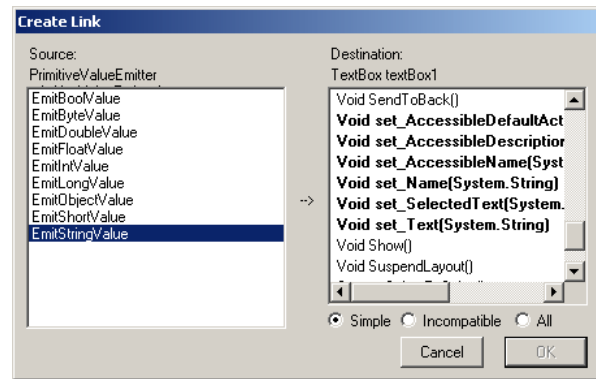


Figure 15: Sélection de la source

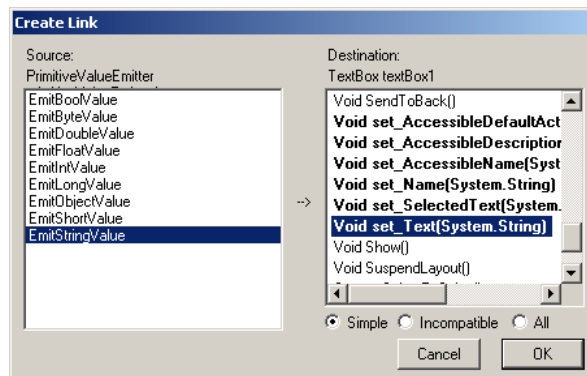


Figure 16: Sélection de la destination

### 3.3. Création d'un lien incompatible entre deux composants

1. Sélectionner **Incompatible** : les méthodes supplémentaires sont celles qui nécessitent un argument de type différent de celui correspondant à l'événement source sélectionné (voir Figure 17).
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 18).

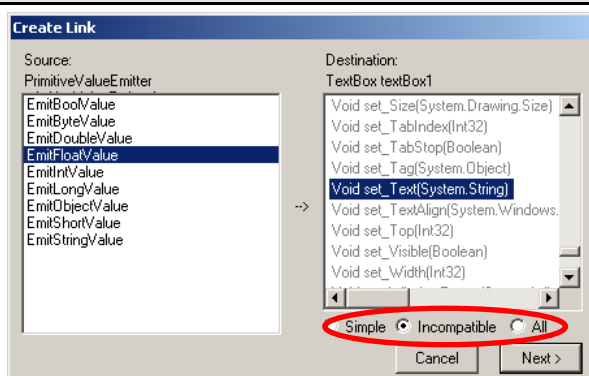


Figure 17: Sélection d'un lien incompatible

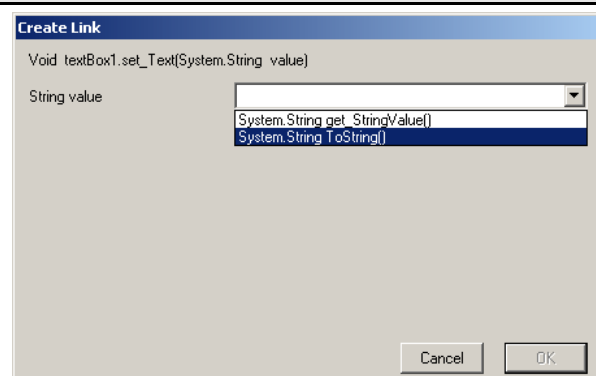


Figure 18: Choix de la méthode de conversion

Dans le cas où l'on sélectionne **All** la liste des méthodes de destination est remplacée par un arbre où vous pouvez choisir les méthodes des valeurs de retour de chaque méthode (et ce de manière récursive).

## 4. Sauvegarde et Chargement d'un assemblage

### 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 19).
2. Sélectionner le fichier à charger dans le concepteur SharpWComp (voir Figure 20). Un dossier Exemples dans le répertoire d'installation de SharpDevelop a été ajouté lors de l'installation du module SharpWComp. Celui-ci contient des exemples d'assemblages de composants.

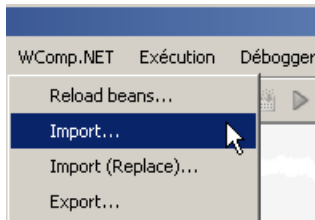


Figure 19: Importation d'un schéma d'assemblage

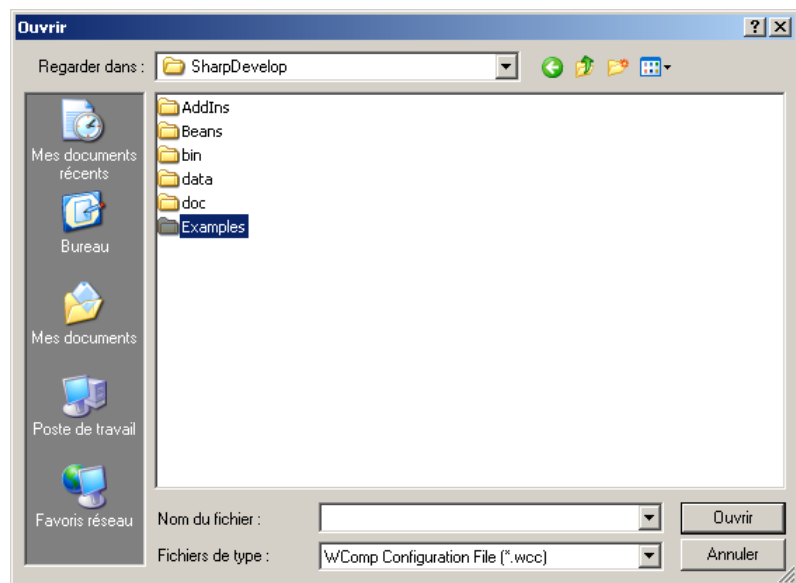


Figure 20: Sélection d'un fichier d'assemblage (wcc)

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 21, Figure 22 et Figure 23).

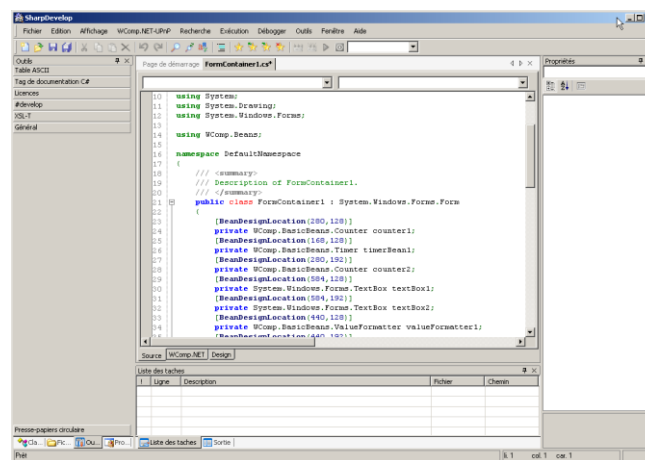


Figure 21: Concepteur Source après importation d'un assemblage (binary-counter-3.wcc)

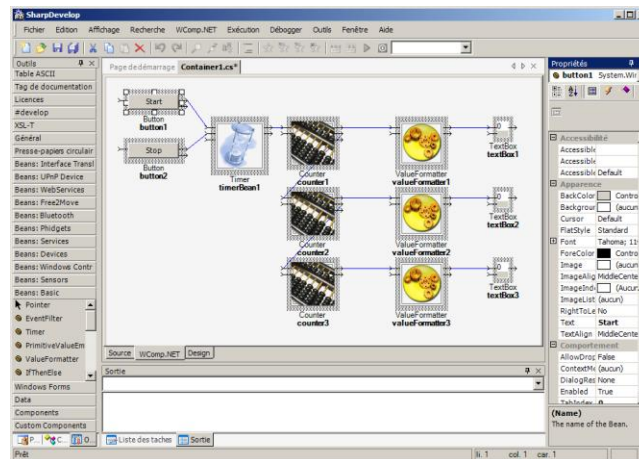


Figure 22: Concepteur SharpWComp après chargement d'un assemblage (binary-counter-3.wcc)

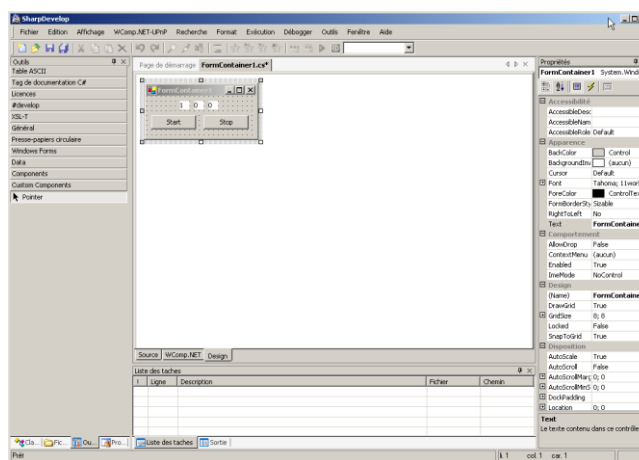


Figure 23: Concepteur Design après chargement d'un assemblage (binary-counter-3.wcc)

## 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.

## 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 ne sauvegarderez que le code source C# généré et vous ne pourrez pas retrouver l'assemblage de composants lors du rechargement du code source C#.

Cette limitation est due au fait que nous pouvons faire la transformation de modèle de l'assemblage vers le code source, mais pas l'inverse. Certaines modifications du code source ne peuvent bien évidemment pas se traduire dans le modèle de WComp (Bean4WComp).

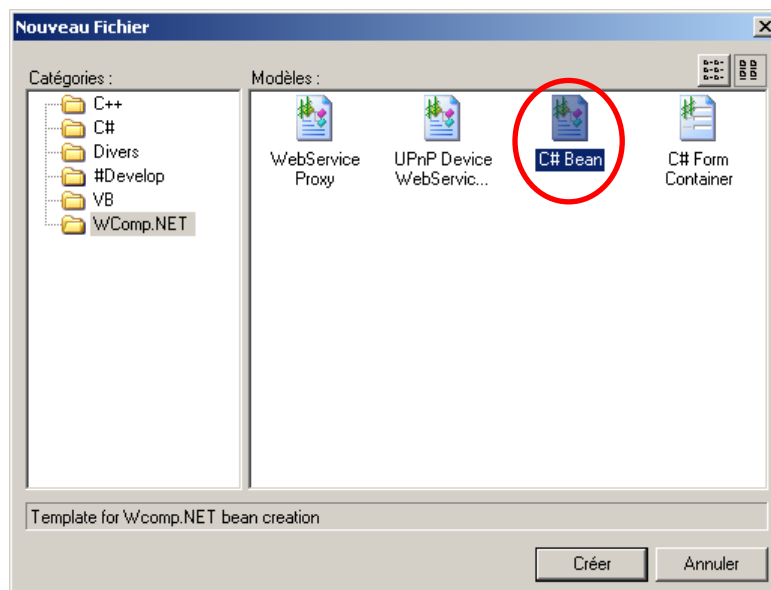
## VI. Création de composants SharpWComp

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

### 1. Code d'un composant à partir d'un modèle

Il est toutefois nécessaire de respecter un format spécifique pour le code source du composant. 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 9)
2. ou Cliquer sur l'icône **Nouveau Fichier** (voir Figure 10)
3. Sélectionner la catégorie **SharpWComp** dans la boîte de dialogue qui s'est ouverte.
4. Cliquer sur le modèle **C# Bean** (voir Figure 24).



**Figure 24:** 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>
    /// Description of Bean1.
    /// This is a sample bean, which has an integer evented property.
    /// </summary>
    [Bean(Category="Basic")]
    public class Bean1 : Drawable
    {
        /// <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>
        /// 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);
        public event IntValueEventHandler PropertyChanged;

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

```

## 2. Rendre ce composant utilisable dans SharpWComp

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).



2. Pour utiliser ce composant, il suffit de copier la dll générée dans le dossier Beans de SharpDevelop (ex : C:\Program Files\SharpDevelop\Beans). Ce répertoire a été automatiquement ajouté à l'installation de SharpWComp (voir IV).

**Attention :** il est impératif de ne pas avoir deux assembly (donc deux dll) correspondant au même composant. Cela provoquera une erreur au chargement du concepteur SharpWComp.

### **3. Icône d'un composant dans l'assemblage**

Depuis la version 0.94, il est possible de modifier la représentation graphique du composant dans l'assemblage et en particulier d'associer l'icône au composant développé (où existant). Pour bénéficier de cette fonctionnalité, la classe Bean1 dérive de la classe Drawable (comportement par défaut de la génération du squelette d'un composant en utilisant le modèle).

Pour associer l'icône, il suffit alors de :

1. Créer un dossier icons à l'endroit où se trouve la dll contenant le Bean développé (ex : C:\Program Files\SharpDevelop\Beans\icons).
2. Ajouter un fichier image (par exemple png) avec : soit 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).

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

## VII. UPnP2Bean

UPnP2Bean est un Wizard permettant de générer automatiquement un composant pour l'environnement SharpWComp à partir d'un serveur ou d'un client 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 VIII « WSDL2Bean », page 16).

### 1. UPnP et SharpWComp

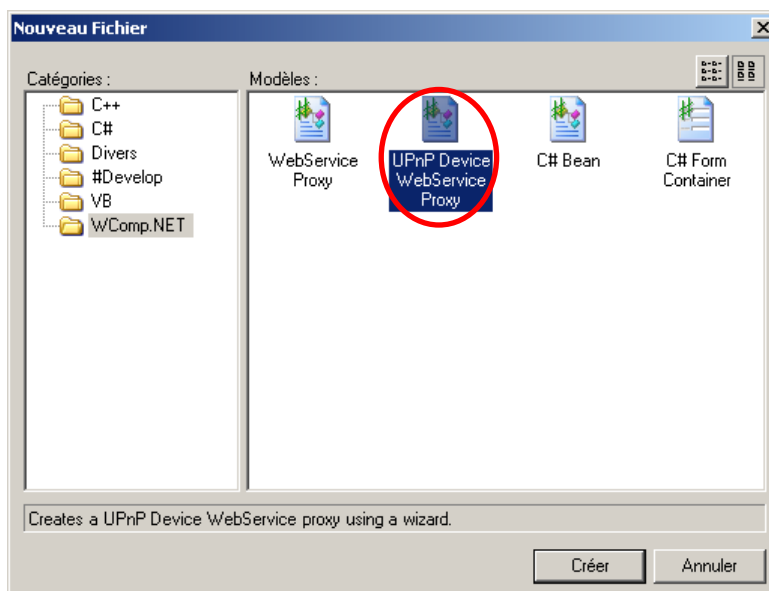
UPnP signifie Universal Plug and Play. Cette technologie vise à fournir un protocole de communication inter-périphériques. Il est aussi parfois appelé : Webservice pour dispositifs.

### 2. Installation du module UPnP2Bean

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

### 3. Utilisation de UPnP2Bean

1. Dans le menu **Fichier**, sélectionner **Nouveau** et cliquer sur **Fichier...** (ou **Ctrl+N**) (voir Figure 9)
2. ou Cliquer sur l'icône **Nouveau Fichier** (voir Figure 10)
3. Sélectionner la catégorie **SharpWComp** dans la boîte de dialogue qui s'est ouverte.
4. Cliquer sur le modèle **UPnP Device Webservice Proxy** (voir Figure 1).



**Figure 25:** 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 final d'une Propriété, 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"

"SwitchPower.0001" qui donne accès à deux méthodes "GetStatus" et "SetTraget", et une variable d'état à évènement "Status".

5. Sélectionner le ou les dispositifs et la ou les méthodes ou variables pour lesquels vous souhaitez générer un composant (voir Figure 26).

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

6. 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, et compiler et placer la dll générée dans les Beans SharpDevelop (voir Figure 27).
7. 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 le Beans WComp.

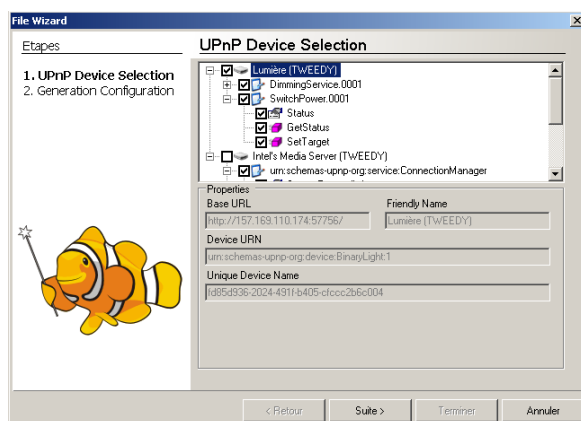


Figure 26: Sélection des dispositifs UPnP

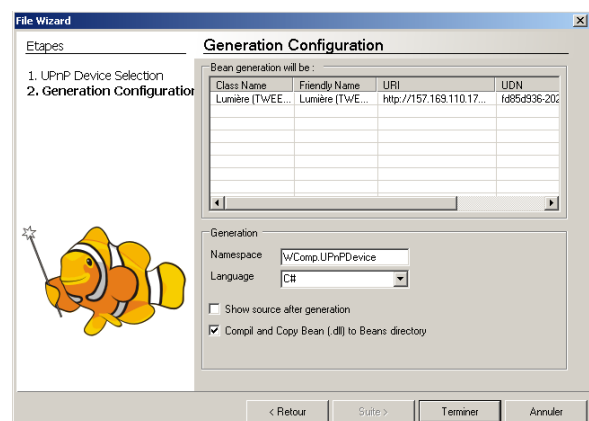


Figure 27: Configuration de la Génération

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).

## VIII. 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.

### 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.

### 2. Installation du module WSDL2Bean

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

### 3. Utilisation de WSDL2Bean

1. Dans le menu **Fichier**, sélectionner **Nouveau** et cliquer sur **Fichier...** (ou **Ctrl+N**) (voir Figure 9).
2. ou Cliquer sur l'icône **Nouveau Fichier** (voir Figure 10).
3. Sélectionner la catégorie **Wcomp.NET** dans la boîte de dialogue qui s'est ouverte.
4. Cliquer sur le modèle **WebService Proxy** (voir Figure 28).

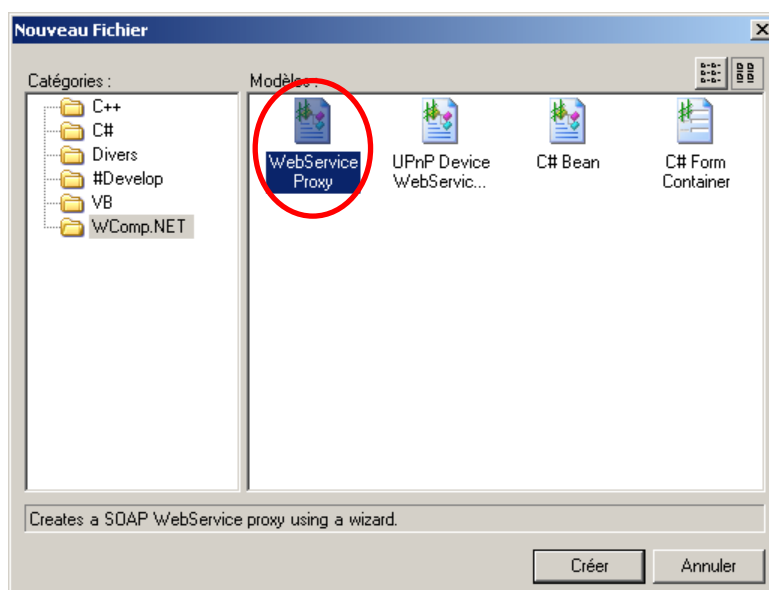


Figure 28: Lancement du Wizard WSDL2Bean

5. Fournir une URL ou un fichier local contenant la description du Webservice (fichier .wsdl). Remplir le champ namespace qui sera le nom du composant que vous allez créer (voir Figure 29).

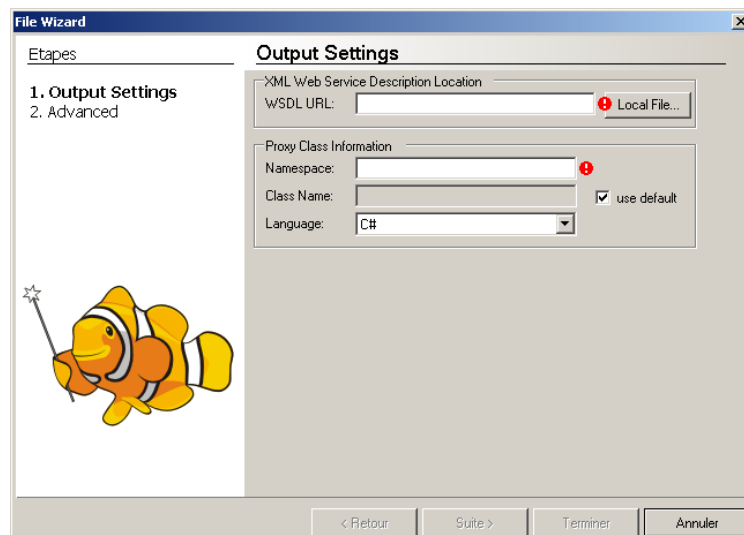


Figure 29 : Sélection du Webservice

6. 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 30 et Figure 31).
7. Dans le cas où l'option « *Compil and Copy Bean (dll) to Beans directory* » est validée, le code généré est automatiquement compilé et la dll est déplacée dans le dossier contenant le Beans Wcomp.

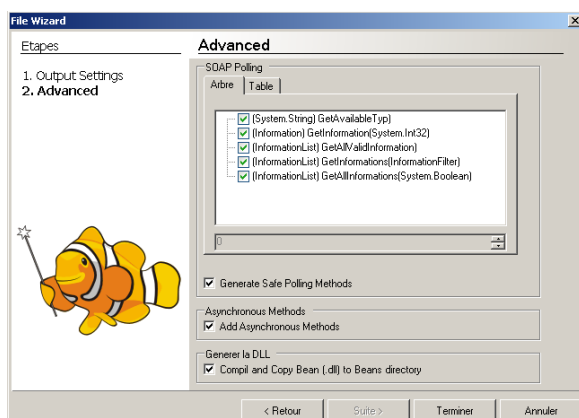


Figure 30 : Affichage sous forme d'arbre

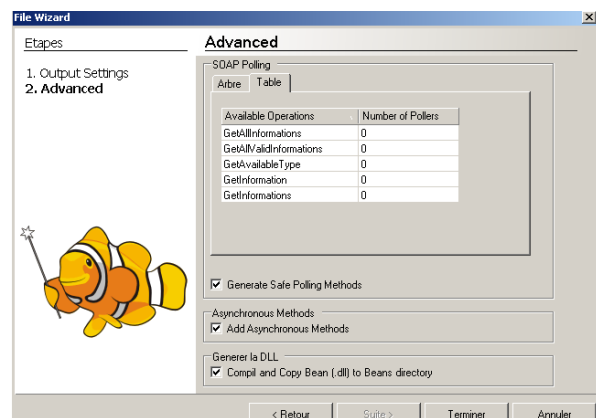


Figure 31: Affichage sous forme de tableau

8. Le code source correspondant à l'accès au Webservice est alors généré (voir Figure 32).
9. Dans le cas où l'option « *Compil and Copy Bean (dll) to Beans directory* » n'est pas validée, répéter l'opération d'écrite dans la section VI.2: « *Rendre ce composant utilisable dans SharpWComp* », page 12.

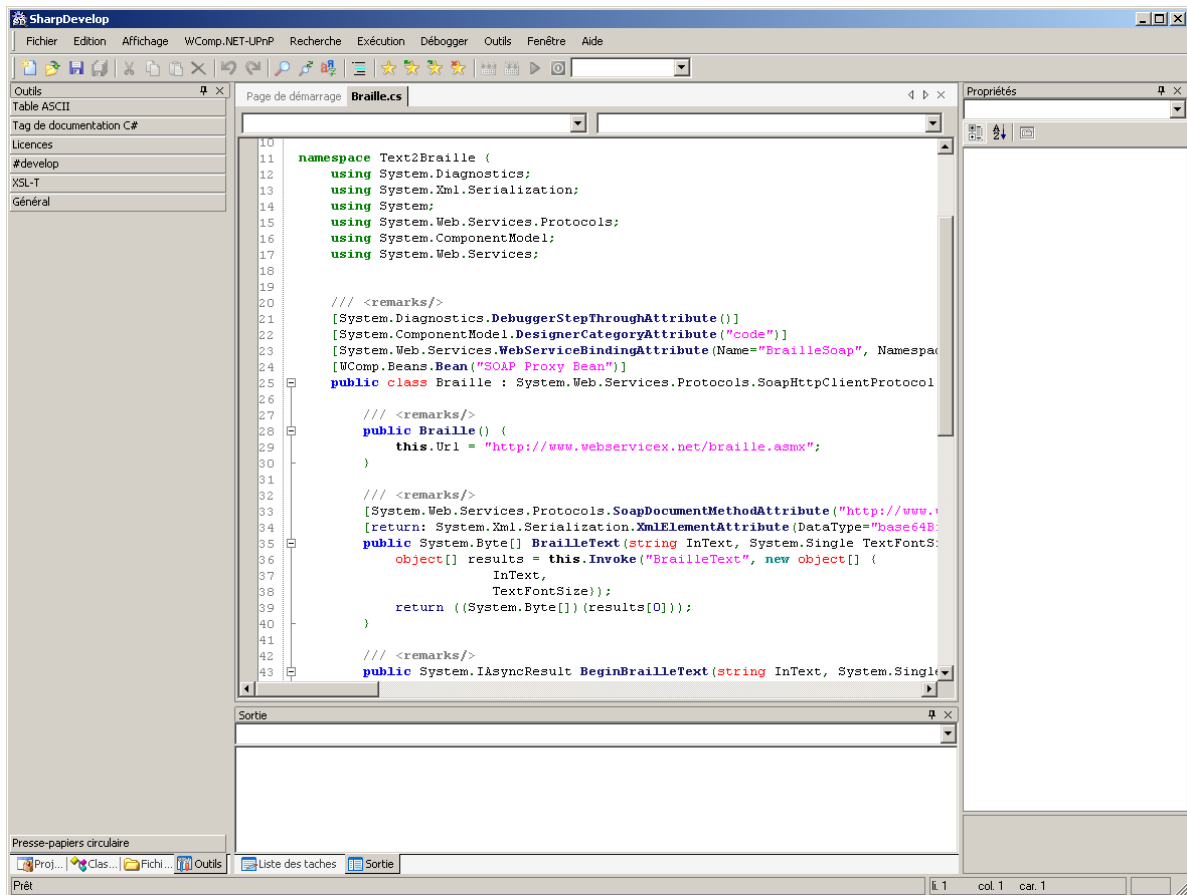


Figure 32: Code source d'un composant WebService généré