



DECOUVERTE DE CODE AVEC LES OUTILS
D'ARCHITECTURE DE CODE

Sommaire

Avertissement	2
Aperçu	3
Introduction à la génération de graphiques de dépendances	4
Introduction à la navigation dans les graphiques de dépendances	16
Travailler avec les graphiques de nœuds et les groupements	26

Avertissement

Ce document est fourni uniquement à titre indicatif. MICROSOFT N'APPORTE AUCUNE GARANTIE, EXPRESSE OU IMPLICITE, A CE DOCUMENT. Les informations figurant dans ce document, notamment les URL et les références aux sites Internet, peuvent être modifiées sans préavis. Les risques d'utiliser ce document ou ses résultats sont entièrement à la charge de l'utilisateur. Sauf indication contraire, les sociétés, les entreprises, les produits, les noms de domaine, les adresses électroniques, les logos, les personnes, les lieux et les événements utilisés dans ce document sont fictifs. Toute ressemblance avec des entreprises, noms d'entreprise, produits, noms de domaine, adresses électroniques, logos, personnes ou événements réels serait purement fortuite et involontaire.

3 Découverte du code avec les outils d'architecture de code

Aperçu

Durée d'exécution du scénario : 30 minutes

Découverte de code avec les outils d'architecture de Visual Studio Ultimate

Dans ce scénario, les outils d'architecture de Visual Studio Ultimate 2012 vont permettre de générer et naviguer dans les graphiques de dépendances pour mieux comprendre et communiquer l'architecture du projet.

Ce scénario est décomposé en 3 parties :

- Introduction à la génération de graphiques de dépendances
- Introduction à la navigation dans les graphiques de dépendances
- Travailler avec les graphiques de noeuds et les groupements

4 Découverte du code avec les outils d'architecture de code

Introduction à la génération de graphiques de dépendances

1. Se connecter en tant que **Claire**. (le mot de passe de tous les utilisateurs est **P2ssw0rd**)
2. Ouvrir Microsoft Visual Studio depuis **Démarrer | Tous les programmes | Microsoft Visual Studio 2012 | Visual Studio 2012**.
3. Sur la page d'accueil, cliquer sur **Se connecter à Team Foundation Server...**

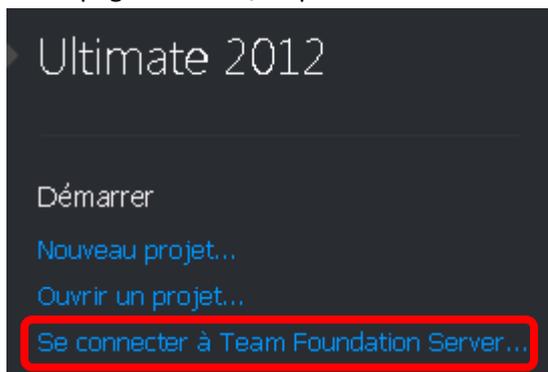


Figure 1 : Se connecter à TFS

4. Dans la fenêtre **Connexion** de **Team Explorer**, double cliquer sur le projet d'équipe **Tailspin Toys**. Si une confirmation est requise, cliquer sur le bouton **Oui**.



Figure 2 : Choix du projet pour la connexion

5 Découverte du code avec les outils d'architecture de code

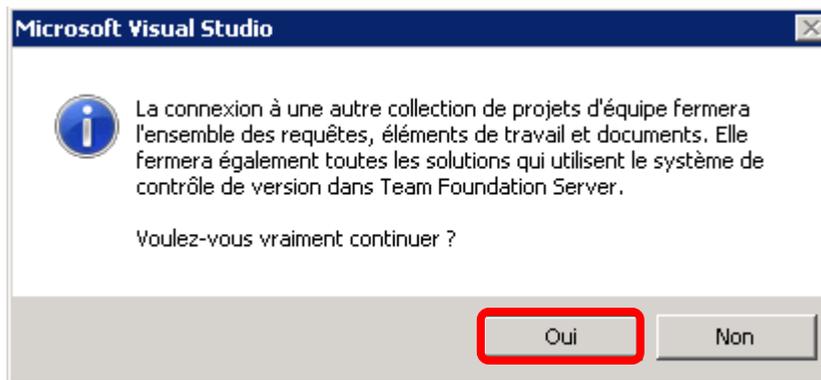


Figure 3 : Confirmation de changement de collection de projets d'équipe

5. Ouvrir **Team Explorer** et cliquer sur le bouton **Début** (icône d'une maison) pour afficher la page d'accueil. S'il n'est pas ouvert, passer par le menu **Affichage -> Team Explorer**. Puis cliquer sur le lien **Explorateur du contrôle de code source**.

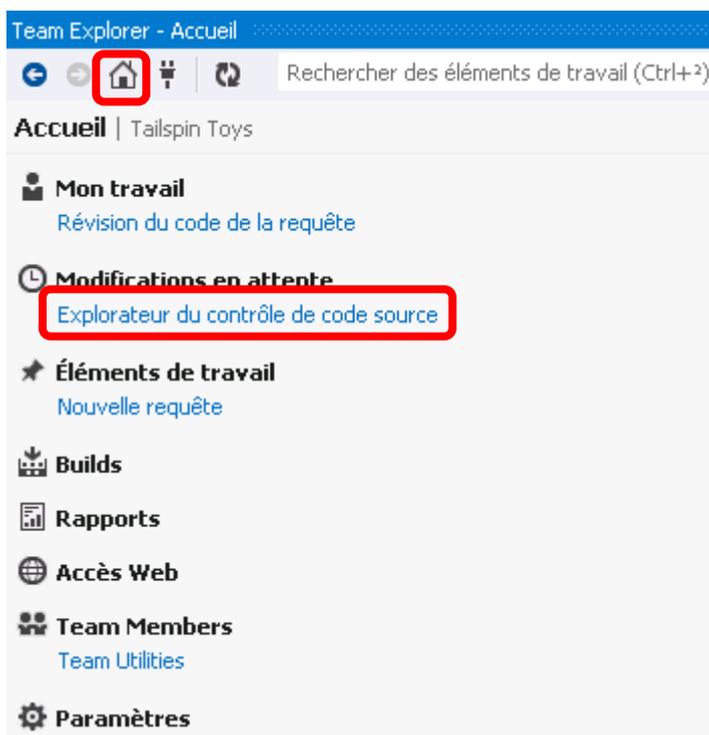


Figure 4 : Page d'accueil du Team Explorer

6 Découverte du code avec les outils d'architecture de code

6. Ouvrir la solution **TailspinToys** en double cliquant sur le fichier **TailspinToys.sln** depuis la branche **Developpement | Iteration 2** dans l'**Explorateur du contrôle de code source**.

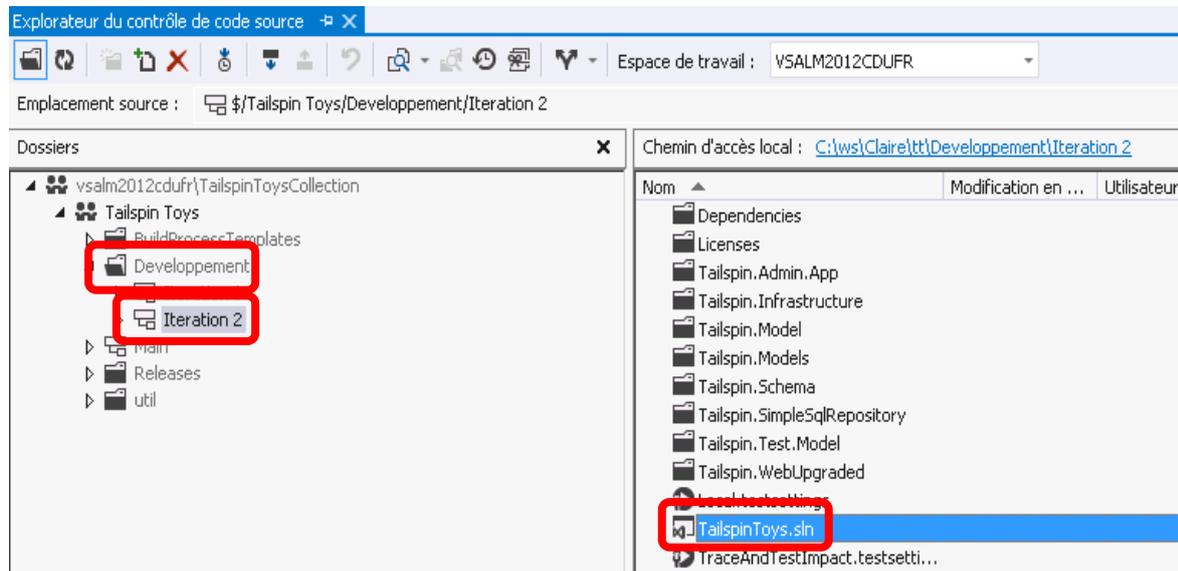


Figure 5 : Sélection de la solution Tailspin Toys

7. Régénérer la solution en sélectionnant **Régénérer la solution** dans le menu **Générer**. Cette opération peut durer quelques minutes.

7 Découverte du code avec les outils d'architecture de code

8. Pour générer le graphique de dépendances complet pour la solution, aller dans le menu **Architecture** puis **Générer un graphique de dépendance** puis **Pour la solution**. Le graphique de dépendances est stocké au format **Directed Graph Markup Language** (dans la suite du scénario appelé **DGML**) qui permet de l'utiliser avec Visual Studio mais aussi d'autres outils. La génération du graphique peut prendre quelques minutes. Lors de la première génération du graphique, un index du code avec les dépendances est construit, ce qui permet d'améliorer les performances pour les opérations futures.



Figure 6 : Création du graphique de dépendance

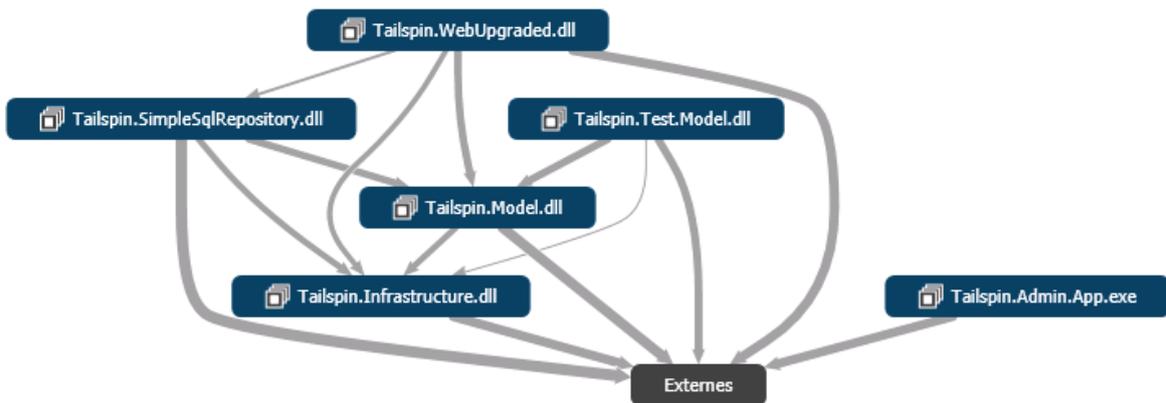


Figure 7 : Graphique de dépendances généré pour la solution

8 Découverte du code avec les outils d'architecture de code

- Le graphique des dépendances montre les dépendances entre assemblies de la solution et ceux externes. Les différentes épaisseurs de trait entre les assemblies montrent le degré de d'interdépendances, plus épais signifie plus d'interdépendances. Chaque assembly peut être étendu pour montrer les enfants ce qui sera vu dans la suite du scénario.

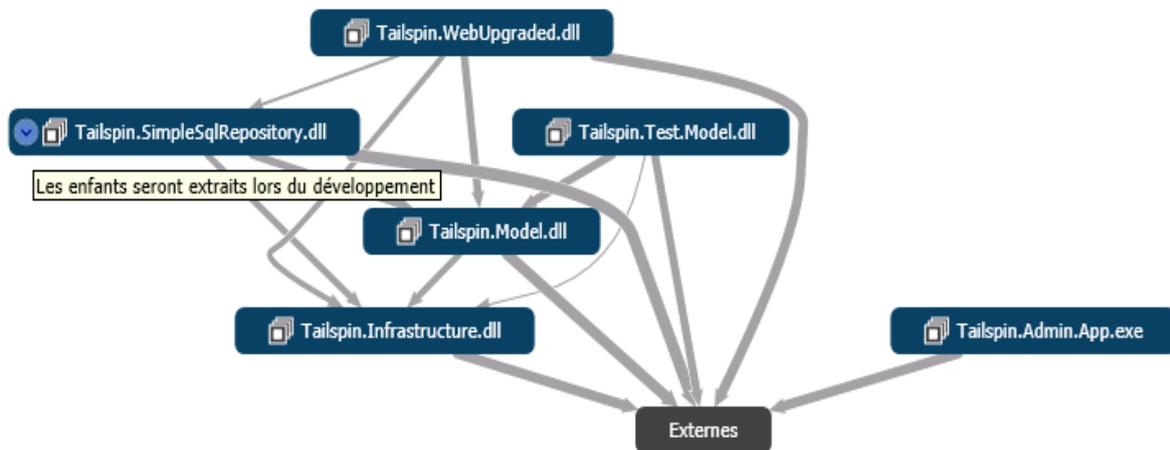


Figure 8 : Possibilité de développer les enfants les assemblies

- Effacer le graphique courant en faisant **Ctrl + A** puis touche **Suppr.**
- Il est possible de générer le graphique des dépendances depuis l'**Explorateur de solutions** ou le **Navigateur de l'architecture**. Cela permet de sélectionner un sous ensemble de types/membres pour créer un nouveau graphique ou ajouter de nouveaux éléments au graphique existant.
- La classe **Product** dans le projet **Tailspin.Model** sera le sujet du scénario. Aller dans l'**Explorateur de solutions**, sélectionner le projet **Tailspin.Model** puis faire clic droit et sélectionner l'option **Limité à**.



Figure 9 : Limité le contenu de l'Explorateur de solutions

9 Découverte du code avec les outils d'architecture de code

13. Dans la zone de recherche l'Explorateur de solutions, saisir **Product** pour lancer une recherche. Cela va automatiquement localiser et surligner la chaîne de caractères recherchée.

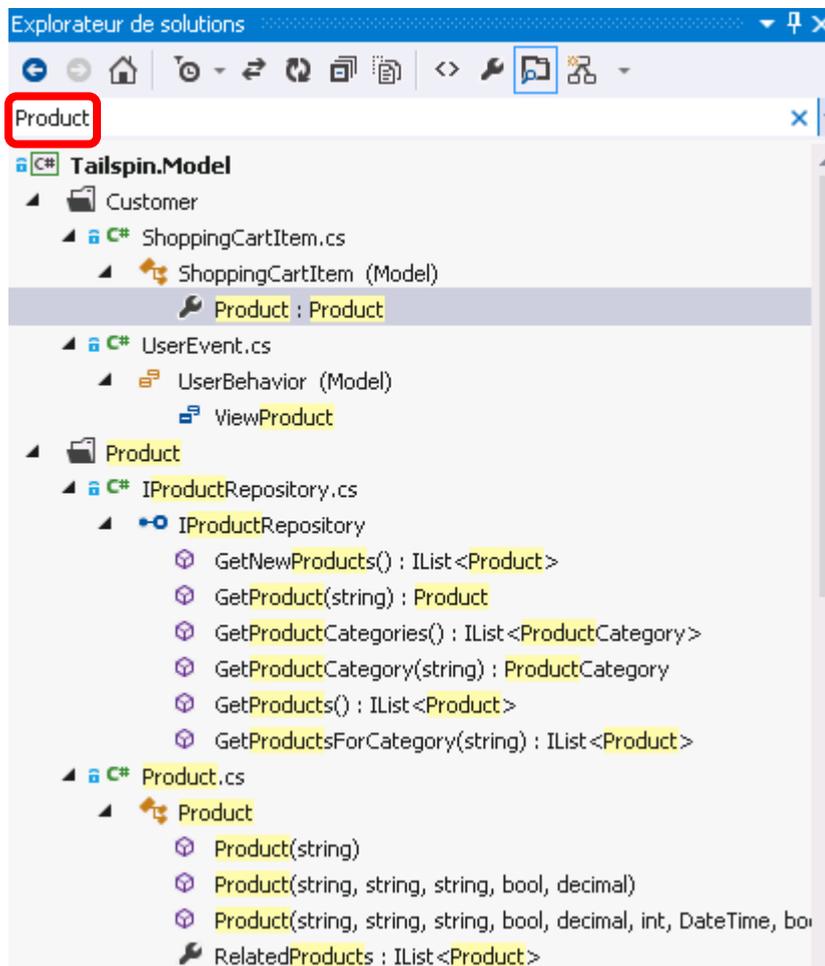


Figure 10 : Résultats de la recherche

10 Découverte du code avec les outils d'architecture de code

14. Sélectionner les résultats de la recherche en cliquant dans la fenêtre de résultats puis faire **Ctrl + A**.

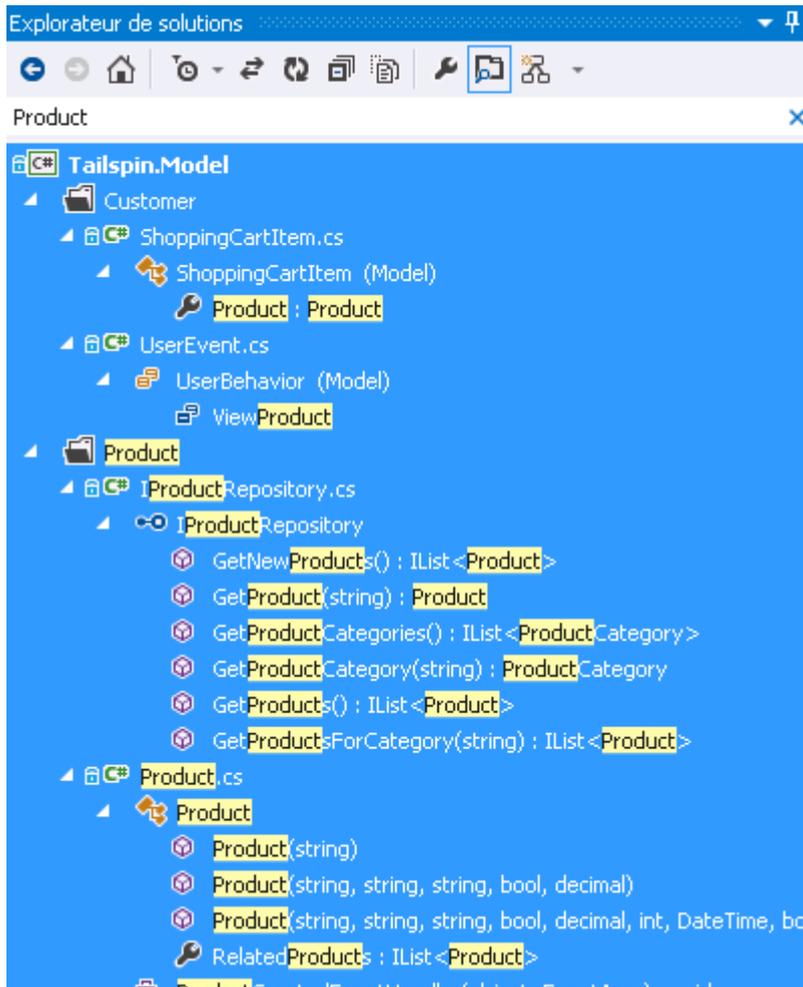


Figure 11 : Sélection des résultats

15. Dans la barre d'outils de la fenêtre de l'**Explorateurs de solutions**, sélectionner le bouton **Créer un document de graphique** puis choisir **Ajouter à un graphique de dépendance actif**. Si le choix **Ajouter à ..** n'est pas affiché, refaire l'opération après avoir sélectionné le fichier DGML.
- 16.

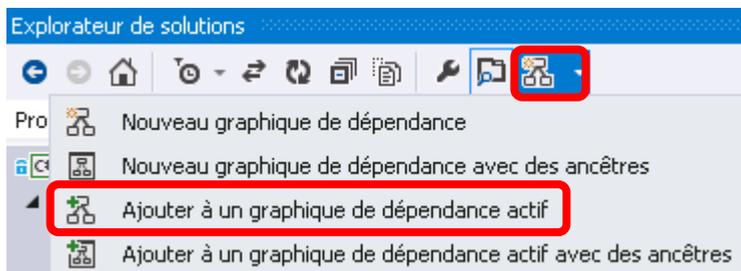


Figure 12 : Ajouter à un graphique de dépendance actif

12 Découverte du code avec les outils d'architecture de code

19. Afin de voir les relations existantes entre les types et méthodes de **Product** du projet **Tailspin.Model** et le projet **Tailspin.Web**, sélectionner **Affichage de classes** dans le **Navigateur de l'architecte**. Cela permet de visualiser l'ensemble des espaces de noms (Namespace).

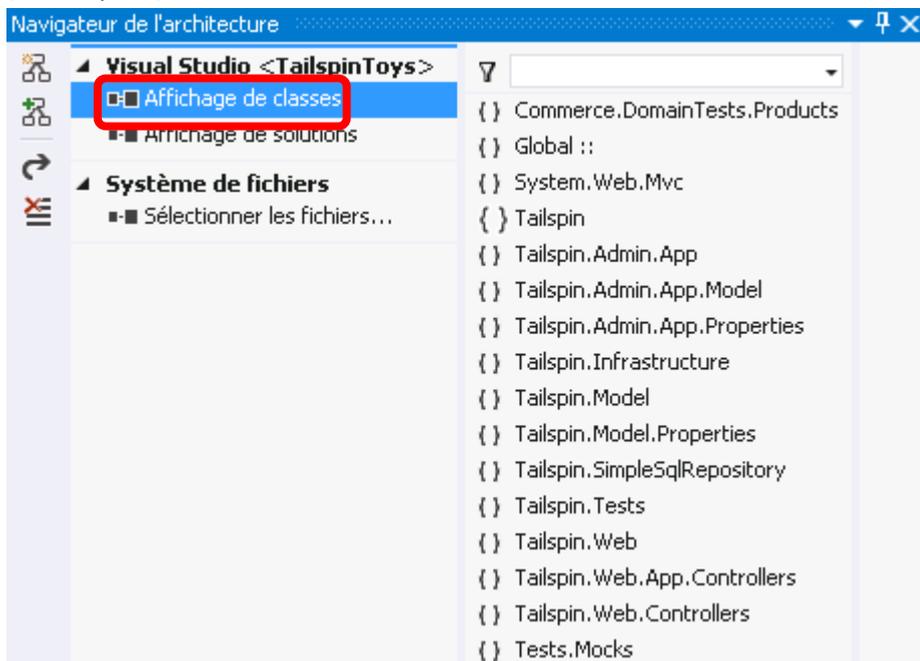


Figure 15 : Affichage des classes du Navigateur de l'architecture

20. Dans la zone de recherche située au-dessus de la liste des espaces de noms, saisir **Tailspin.Web** puis **Entrée** pour visualiser uniquement les espaces de noms du projet **Tailspin.Web**.

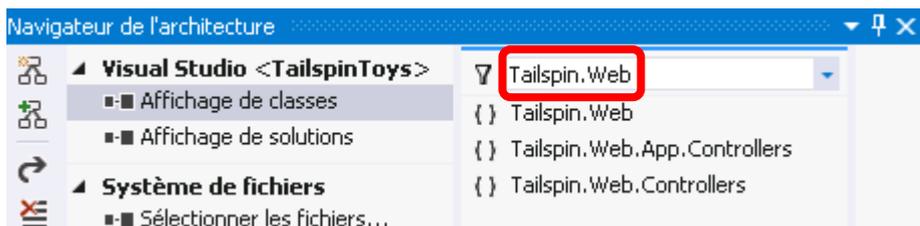


Figure 16 : Filtre sur Tailspin.Web

13 Découverte du code avec les outils d'architecture de code

21. Sélectionner l'ensemble des espaces de noms en faisant **Ctrl + A** pour lister l'ensemble des types qu'ils contiennent.

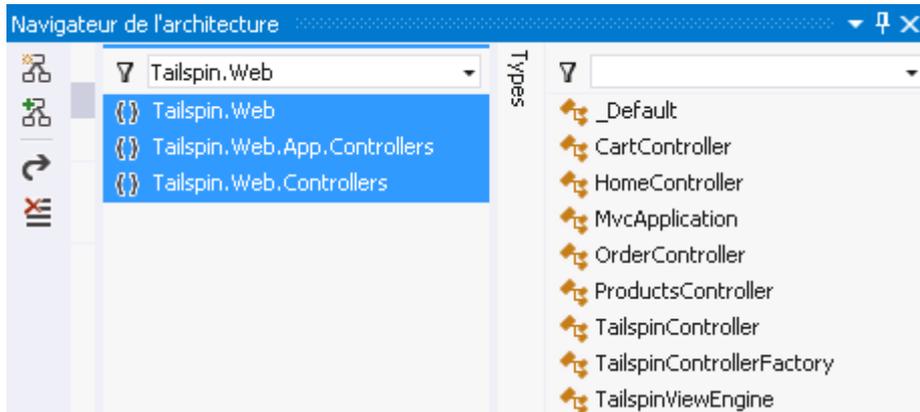


Figure 17 : Liste des types

22. Lorsque les types des espaces de noms **Tailspin.Web.*** sont listés, faire de nouveau **Ctrl + A** pour les sélectionner.

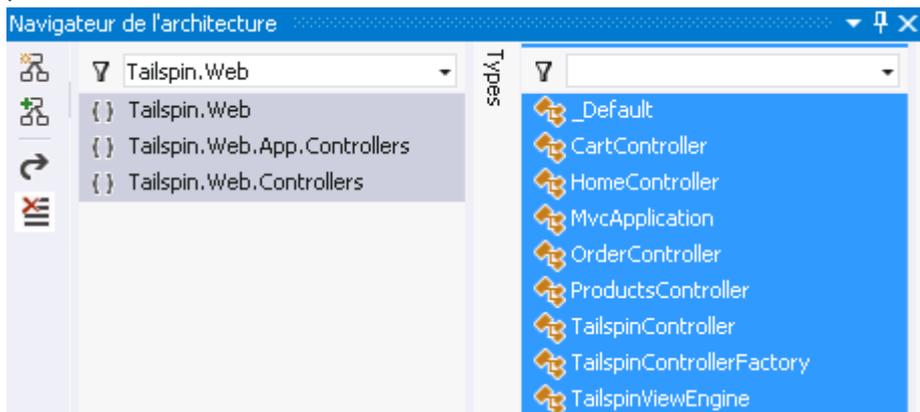


Figure 18 : Ensemble des types sélectionnés

14 Découverte du code avec les outils d'architecture de code

23. Cliquer sur les types sélectionnés et maintenir le clic pour les glisser et les déposer sur le graphique de dépendance précédemment créé. Avant de relâcher le clic, maintenir la touche **Ctrl** appuyée. La touche Ctrl permet de grouper l'ensemble des types dans un container.

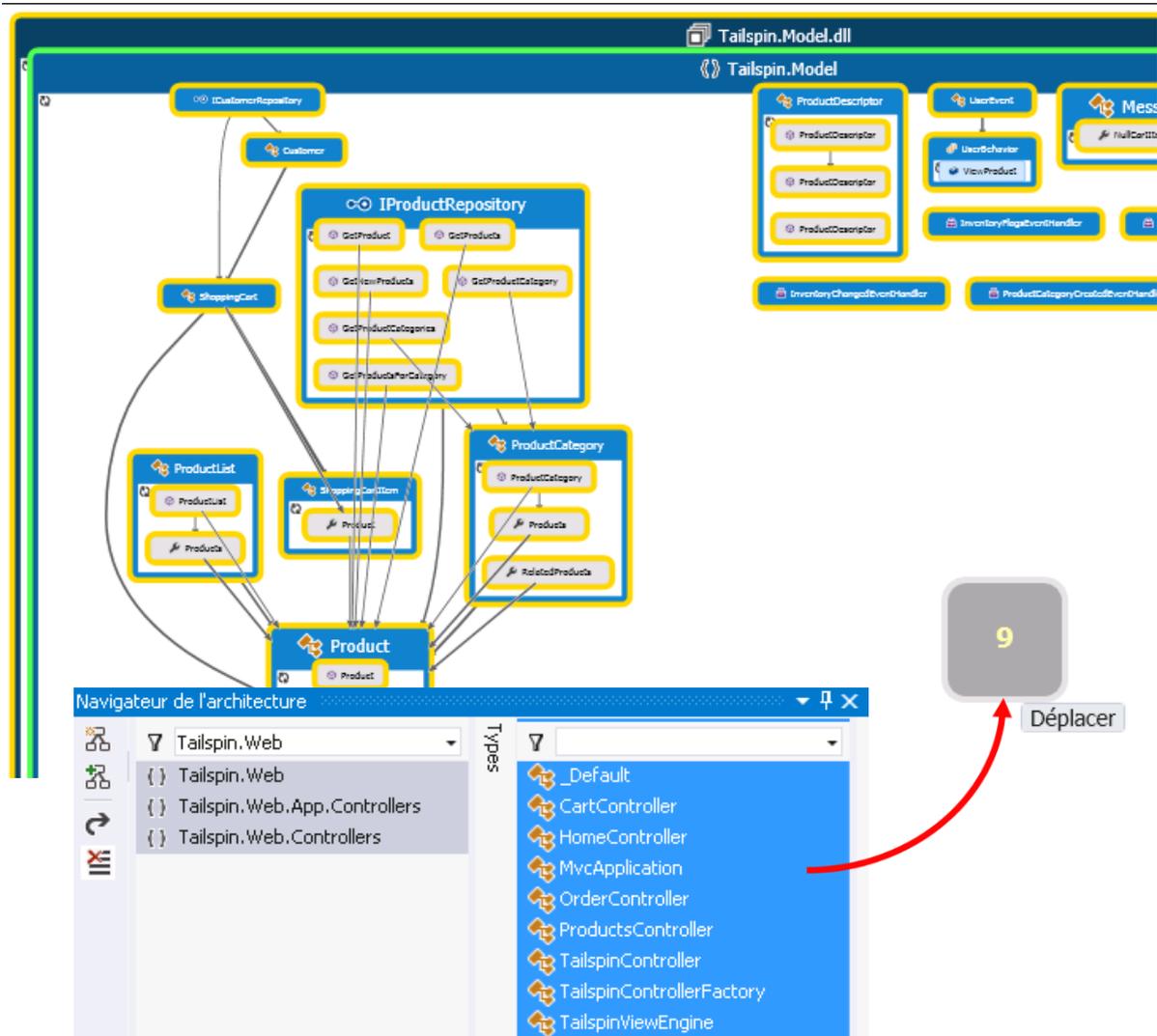


Figure 19 : Insertion des types dans le graphique de dépendances

15 Découverte du code avec les outils d'architecture de code

24. Le résultat produit permet de visualiser les relations qui entre la classe **Product** et le projet **Tailspin.Web** qui est l'application Web.

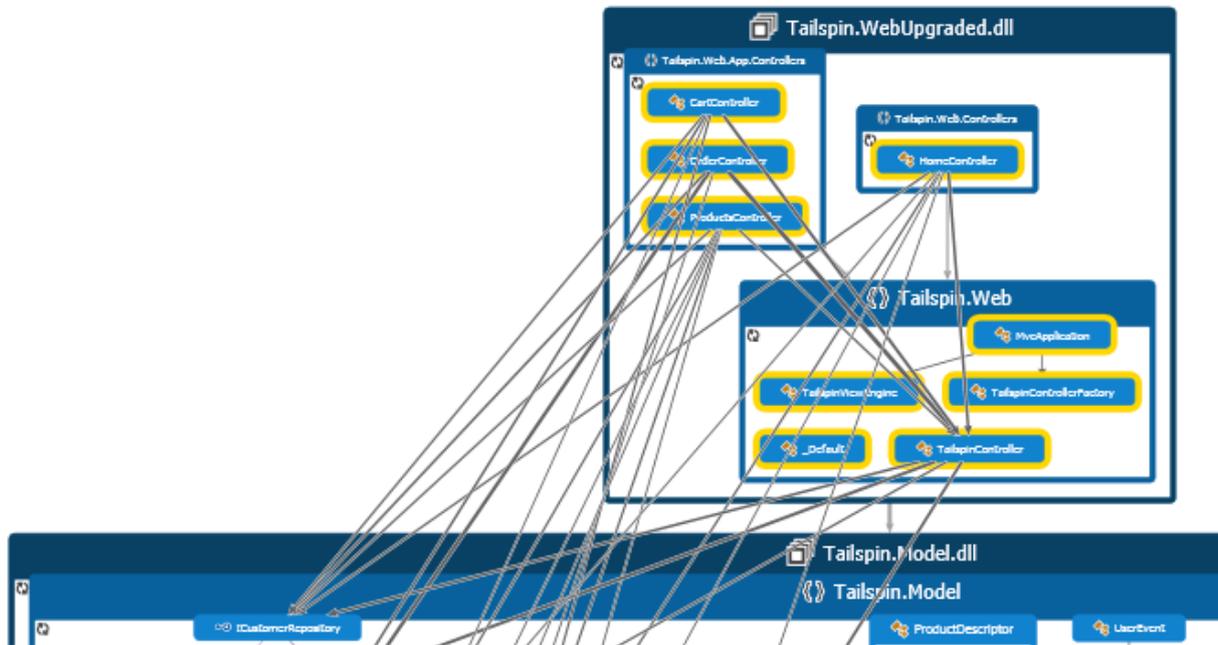


Figure 20 : Résultat du graphique de dépendance

25. Quand l'analyse est terminée, la fenêtre des **Résultats d'analyse des clones de code** apparaît et liste tous les candidats potentiels classés par type de correspondance. Cliquer sur la flèche pour visualiser les fichiers trouvés.
26. Un autre scénario est disponible sur l'**Analyse de code avec le navigateur de l'architecture** qui permet de découvrir plus en détails le fonctionnement du Navigateur de l'architecture.

Introduction à la navigation dans les graphiques de dépendances

Dans cette partie du scénario, la navigation et la manipulation des graphiques de dépendance seront abordées.

1. Reprendre le graphique de dépendance précédemment créé, zoomer, si nécessaire, pour visualiser la classe **HomeController**. Zoomer en utilisant les boutons de la barre d'outils du graphique ou simplement en utilisant les boutons de défilement de la souris.

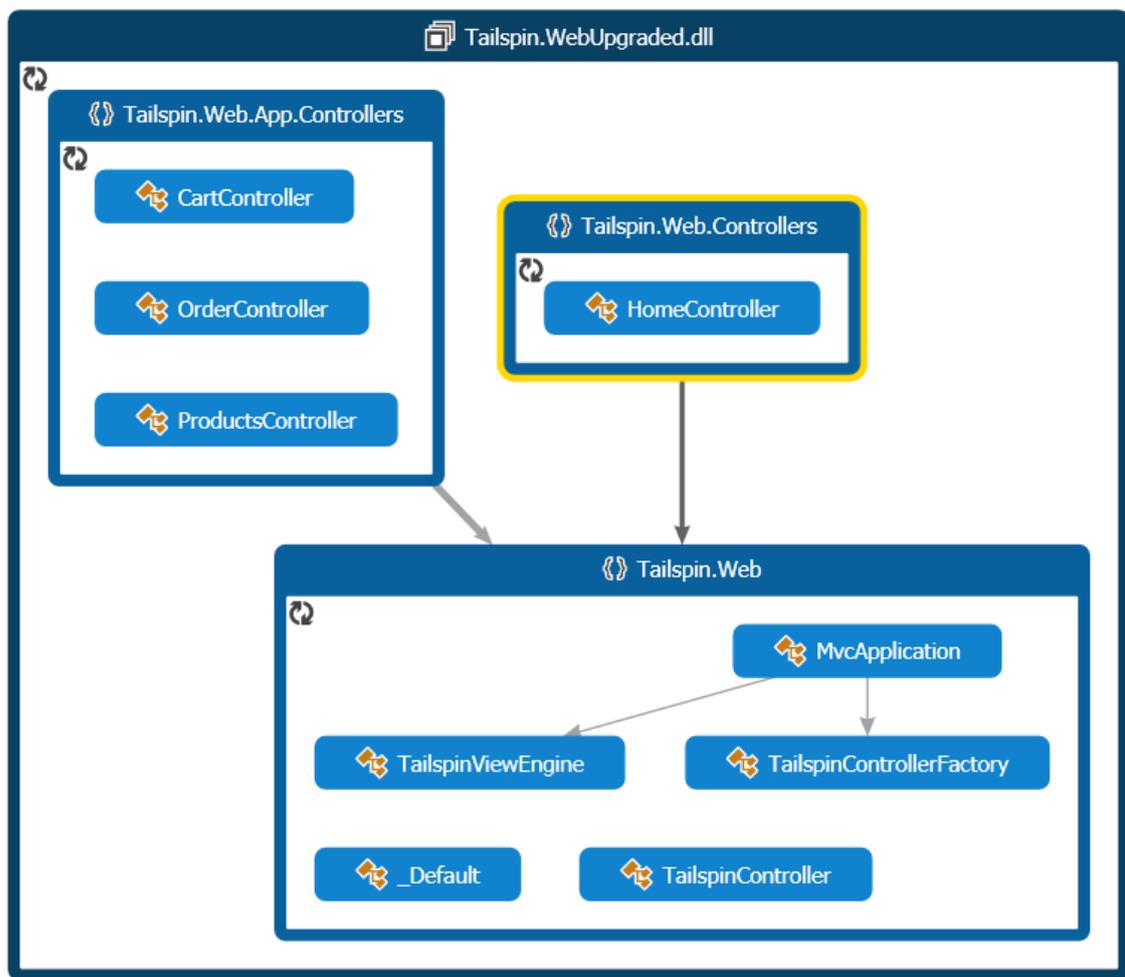


Figure 21 : Vue de la classe **HomeController** dans le graphique de dépendance

2. Sélectionner la classe **HomeController** (en jaune) et passer le curseur de la souris sur les liens de la classe jusqu'à trouver celui qui a pour nœud cible **IProductRepository**.

17 Découverte du code avec les outils d'architecture de code

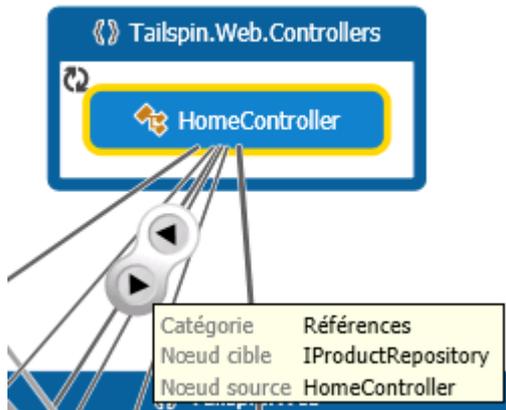


Figure 22 : Relation avec nœud cible IProductRepository

3. Le contrôle de navigation qui apparaît sur les liens fournit 2 actions possibles qui permettent de naviguer vers la source ou la cible du lien. Cliquer sur la flèche qui pointe en direction de la cible **IproductRepository**.

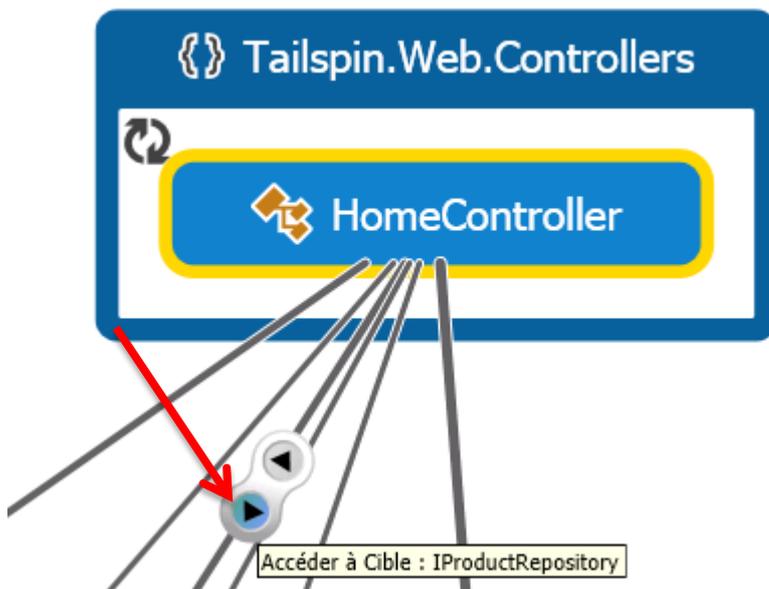


Figure 23 : Navigation vers un autre nœud

4. Utiliser le bouton **Zoom pour ajuster** depuis la barre d'outils du graphique afin que tous les éléments soient visibles.



Figure 24 : Emplacement du bouton Zoom pour ajuster

18 Découverte du code avec les outils d'architecture de code

- Il est maintenant possible de visualiser les relations qui existent entre la classe **HomeController** et l'interface **IProductRepository** par l'appel de **GetProduct** et **GetProductCategory**. Au cas où, tous les liens sont affichés, cliquer sur l'option **Afficher les liens entre les groupes sur les nœuds sélectionnés** depuis la barre d'outils.

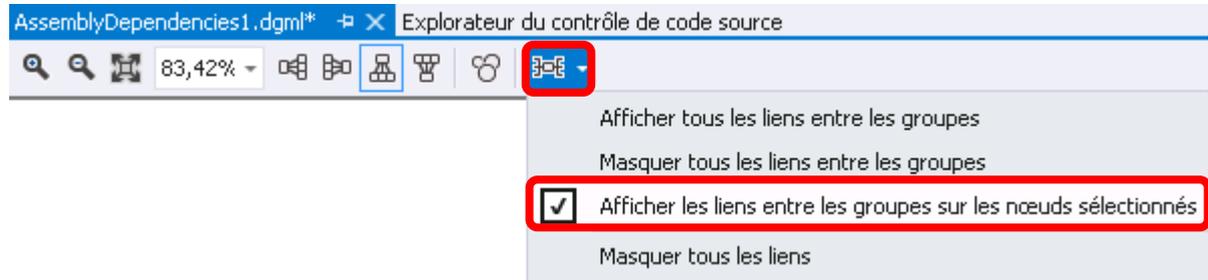


Figure 25 : Afficher les liens entre les groupes sur les nœuds sélectionnés

19 Découverte du code avec les outils d'architecture de code

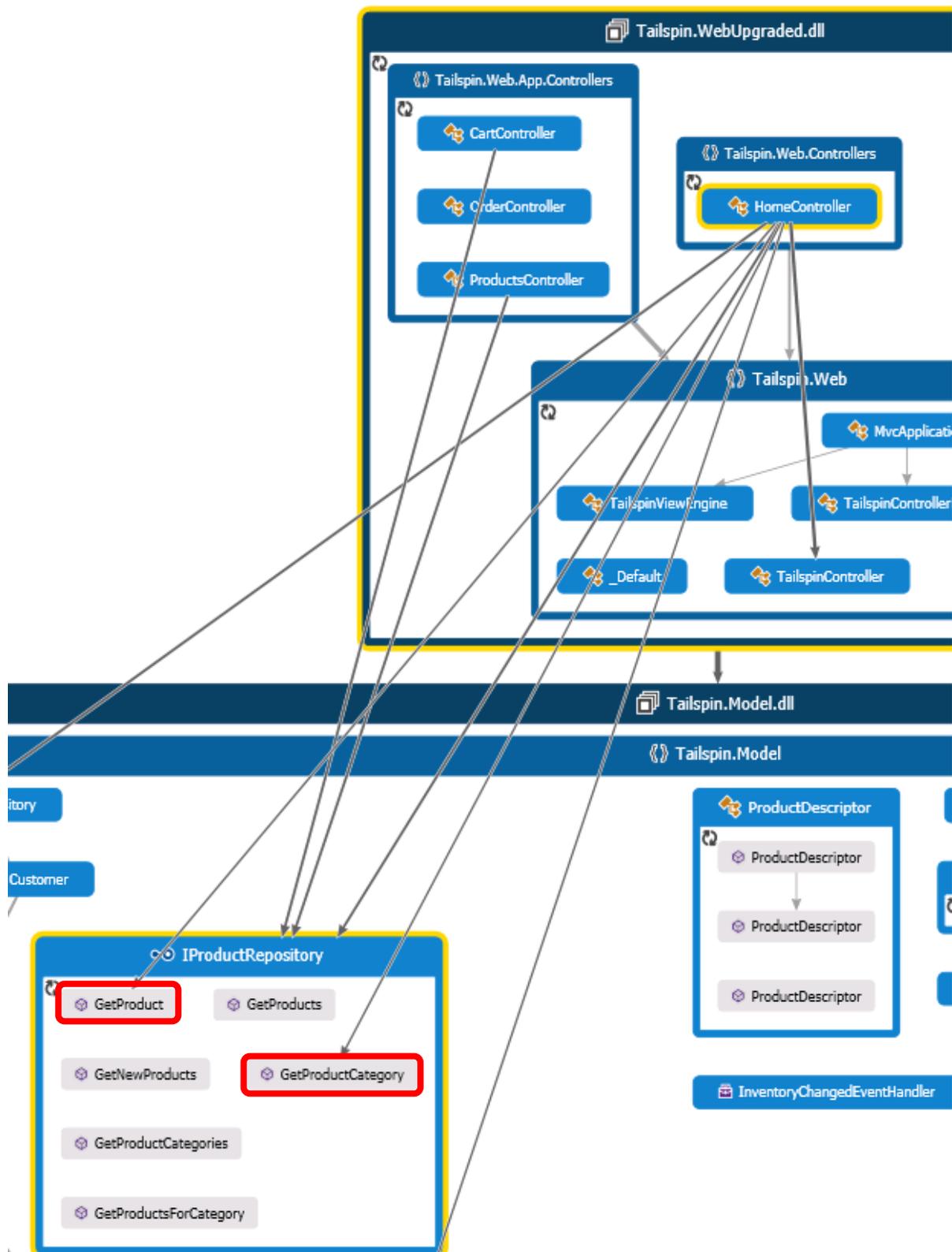


Figure 26 : Graphique de dépendance entre la classe HomeController et l'interface IProductRepository

20 Découverte du code avec les outils d'architecture de code

- Il est possible de naviguer depuis le graphique directement au code associé. Faire un clic droit que le nœud **HomeController** et sélectionner **Atteindre** -> **Atteindre la définition** ou utiliser la touche **F12**.



Figure 27 : Atteindre la définition dans le code

- Le code est affiché et permet de voir comment les méthodes définies par l'interface sont utilisées dans le code.

```
[HandleError]
public class HomeController : TailspinController {

    IProductRepository _productRepository;
    ICustomerRepository _customerRepository;

    public HomeController(ICustomerRepository customerRepository,
        IProductRepository productRepository) : base(customerRepository)
    {

        _productRepository = productRepository;
        _customerRepository = customerRepository;

    }

    public ActionResult Index(string slug) {
        if (!string.IsNullOrEmpty(slug))
        {
            this.Products = this._productRepository.GetProductCategory(slug).Products;
            return View("Listing");
        }
    }
}
```

Figure 28 : Code de la classe HomeController

21 Découverte du code avec les outils d'architecture de code

- La navigation dans un graphique de dépendance peut permettre de déterminer si le projet utilise des assemblies externes. Générer un nouveau graphique de dépendance par le menu **Architecture -> Générer un graphique de dépendance -> Pour la solution.**
- Ouvrir le groupe **Externes** en cliquant sur le bouton  pour afficher l'ensemble des assemblies externes utilisés par l'application **Tailspin**.

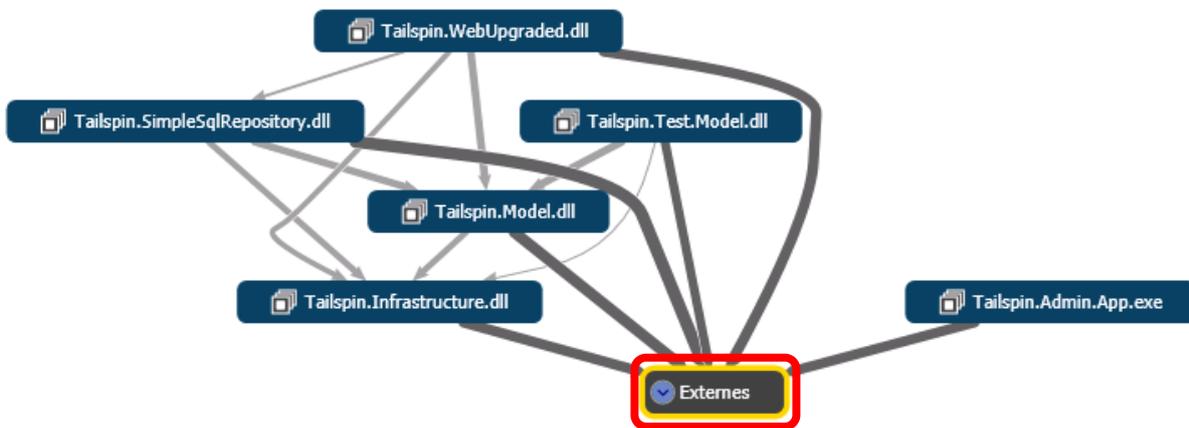


Figure 29 : Assembly Externes

- Zoomer et ouvrir le nœud **System.Web.dll** dans le groupe Externes.

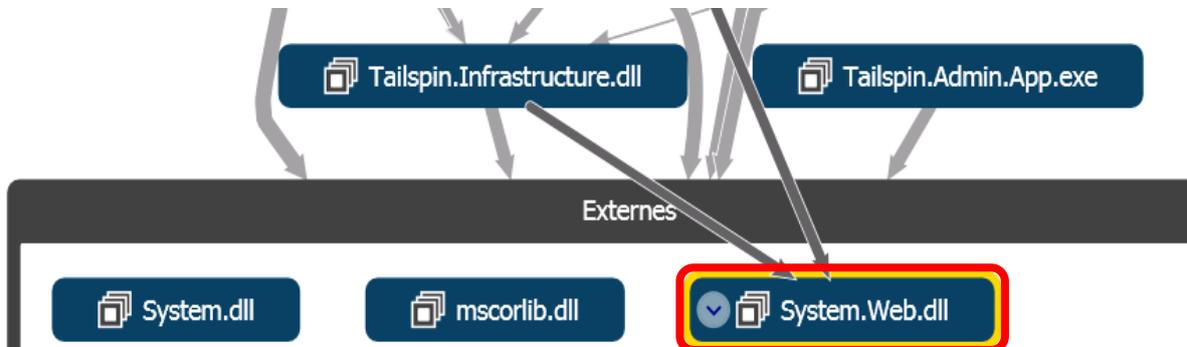


Figure 30 : l'assembly System.Web.dll

22 Découverte du code avec les outils d'architecture de code

11. Trouver et ouvrir le nœud **System.Web.Routing** pour visualiser les types contenus.

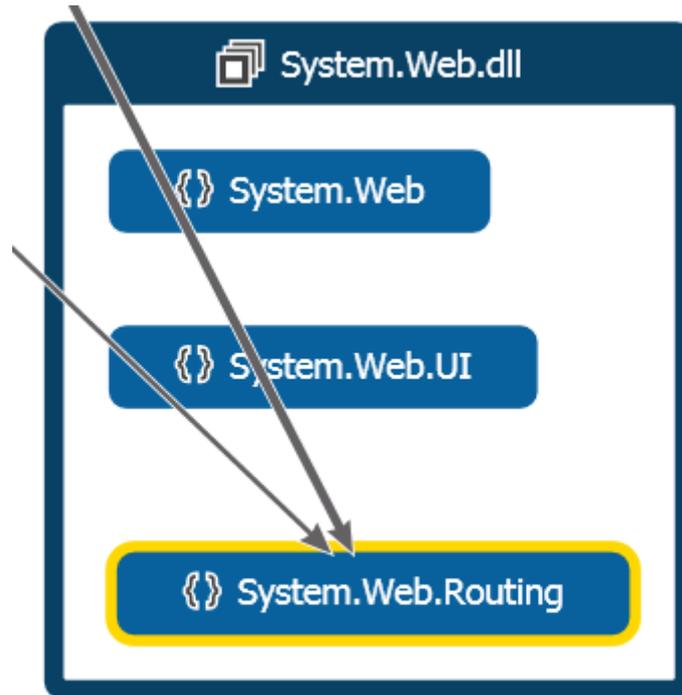


Figure 31 : Ouvrir le nœud `System.Web.Routing`

23 Découverte du code avec les outils d'architecture de code

12. Sélectionner la classe **RouteTable** pour voir ses dépendances.

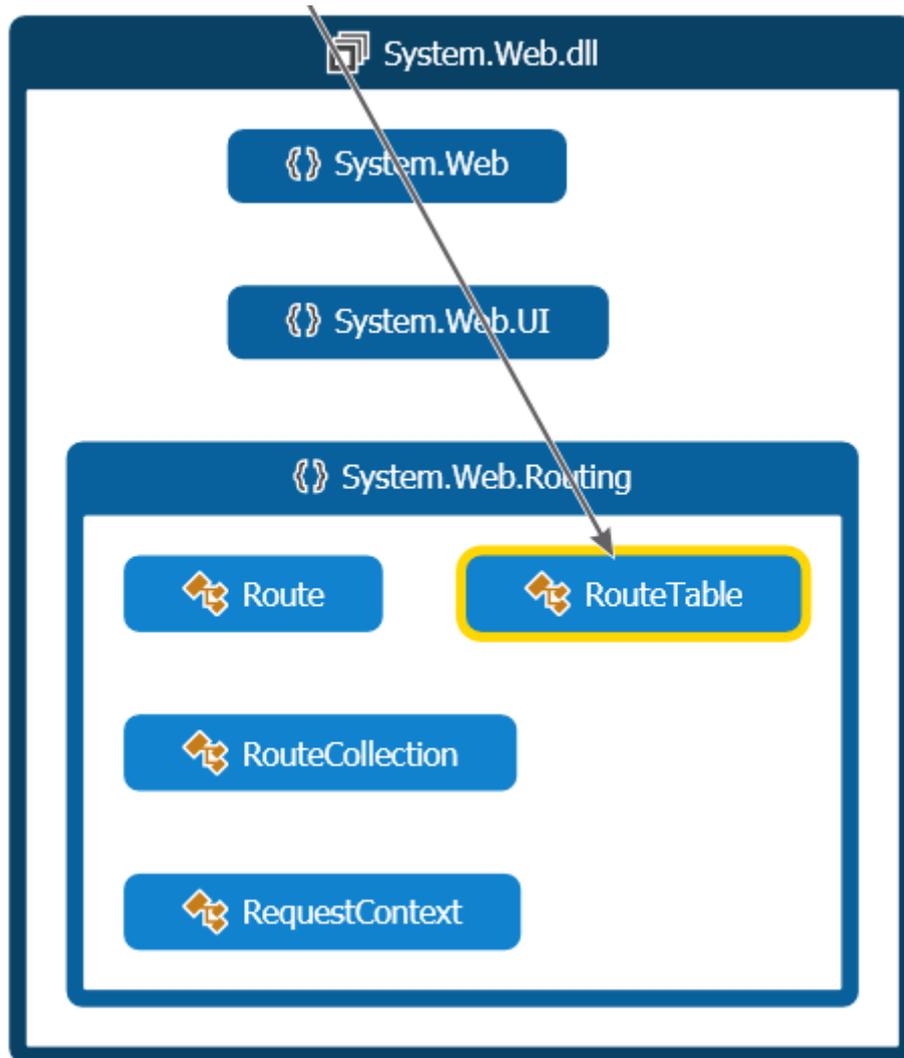


Figure 32 : La classe RouterTable et ses dépendances

24 Découverte du code avec les outils d'architecture de code

13. Sélectionner la relation qui a pour Nœud source **Tails핀.WebUpgraded.dll**. Il est possible, comme précédemment, de cliquer sur la flèche pour visualiser le nœud source. Cependant, un nouveau graphique peut être créé directement depuis la relation sélectionnée. Pour créer le graphique, faire **Shift** plus clic sur le bouton + du centre de la flèche sur la relation sélectionnée.



Figure 33 : Création d'un nouveau graphique à partir d'une relation

25 Découverte du code avec les outils d'architecture de code

14. Dans le graphique créé, la méthode **MvcApplication.Application_Start** appelle la méthode **get_Routes** (la propriété `RouteTable.Routes`).

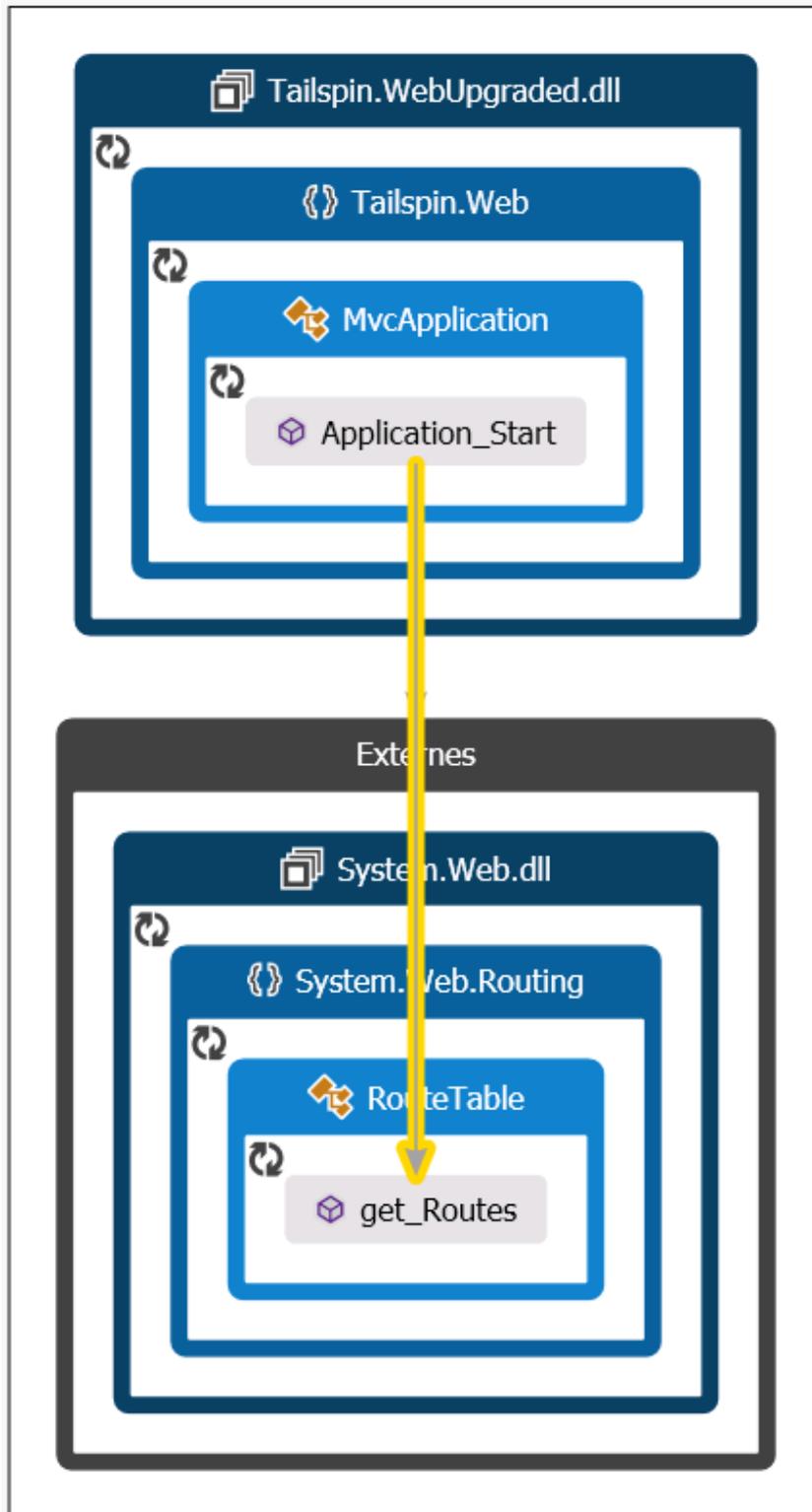


Figure 34 : Relation entre `ApplicationStart` et `get_Routes`

15. Fermer le diagramme de `ContributionLinks1.dgml` précédemment créé sans le sauvegarder.

Travailler avec les graphiques de nœuds et les groupements

Dans cette partie du scénario, il s'agira de réduire la complexité du graphique de dépendance en supprimant les nœuds non significatifs et en modifiant les propriétés du graphique.

1. Reprendre le graphique de dépendance créé précédemment en réduisant tous les nœuds en cliquant avec le bouton droit sur le graphique puis sélectionner **Grouper** -> **Réduire tout**. Sinon régénérer le graphique par le menu **Architecture** -> **Générer un graphique de dépendance** -> **Pour la solution**.

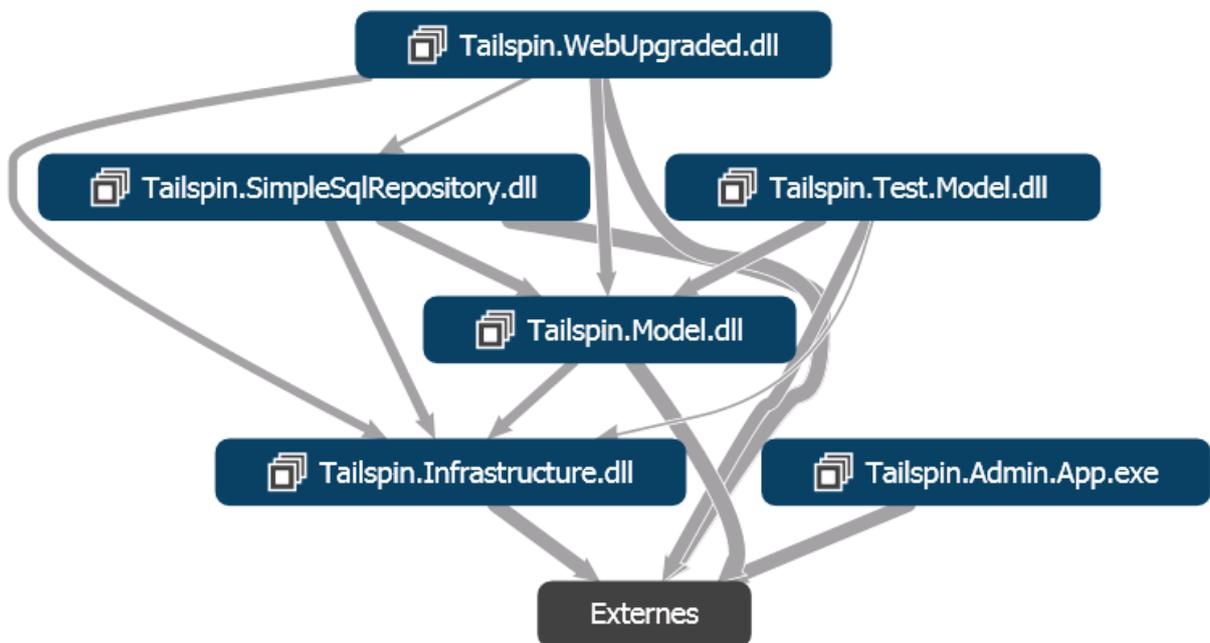


Figure 35 : Graphique de dépendance

2. Ouvrir le groupe **Externes** en cliquant sur le bouton  pour afficher l'ensemble des assemblages externes utilisés par l'application **Tailspin**. Les lignes grises représentant les relations entre les nœuds s'arrêtent à la limite des groupes. Cela permet de réduire la complexité visuelle du graphique. En supprimant le groupe Externes, les relations directes entre les différents assemblages externes et internes.

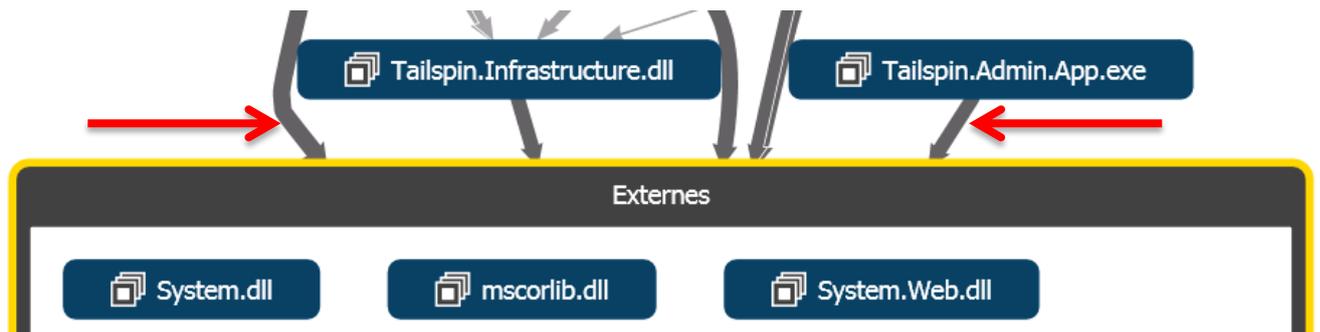


Figure 36 : Relations entre assemblages

27 Découverte du code avec les outils d'architecture de code

3. Sélectionner le nœud **microsoft.dll** et utiliser la touche **Suppr** pour le supprimer du graphique. Les nœuds externes apportent de la complexité au graphique. De plus, cette information n'est pas forcément nécessaire dans le cadre de la découverte de la structure du code.
4. Supprimer le nœud **System.dll** dans le groupe **Externes**.
5. Faire un clic droit sur le groupe **Externes** puis sélectionner **Grouper -> Supprimer le groupe**. Cela supprime le groupe mais pas les nœuds de celui-ci.

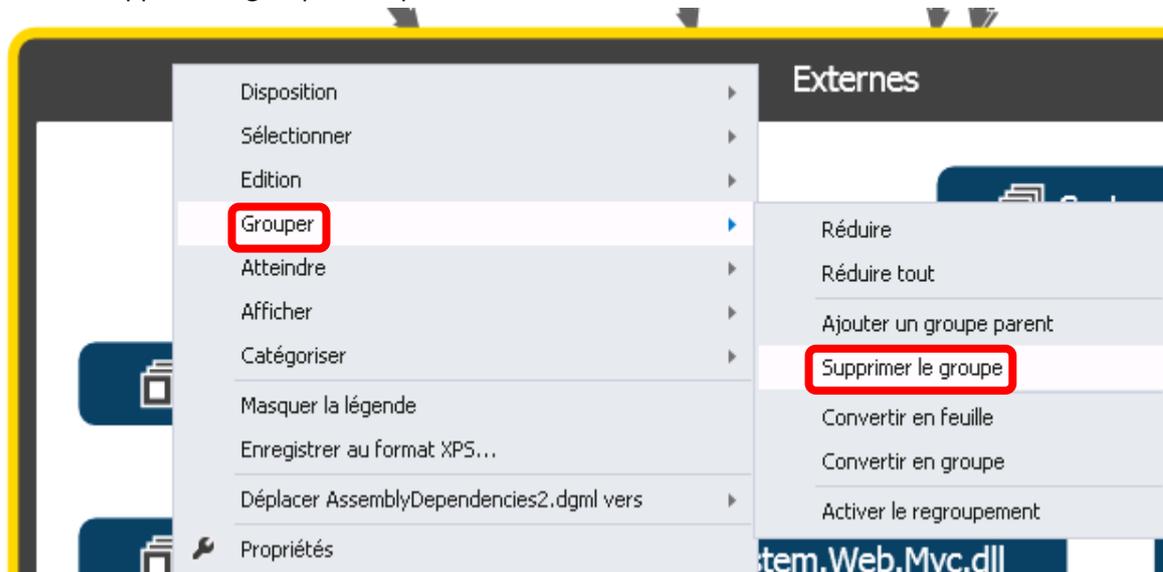


Figure 37 : Supprimer le groupe Externes

6. Supprimer le nœud **Tailspin.Test.Model.dll** du graphique. Cela permet de visualiser les dépendances de l'application déployée sans la partie tests.

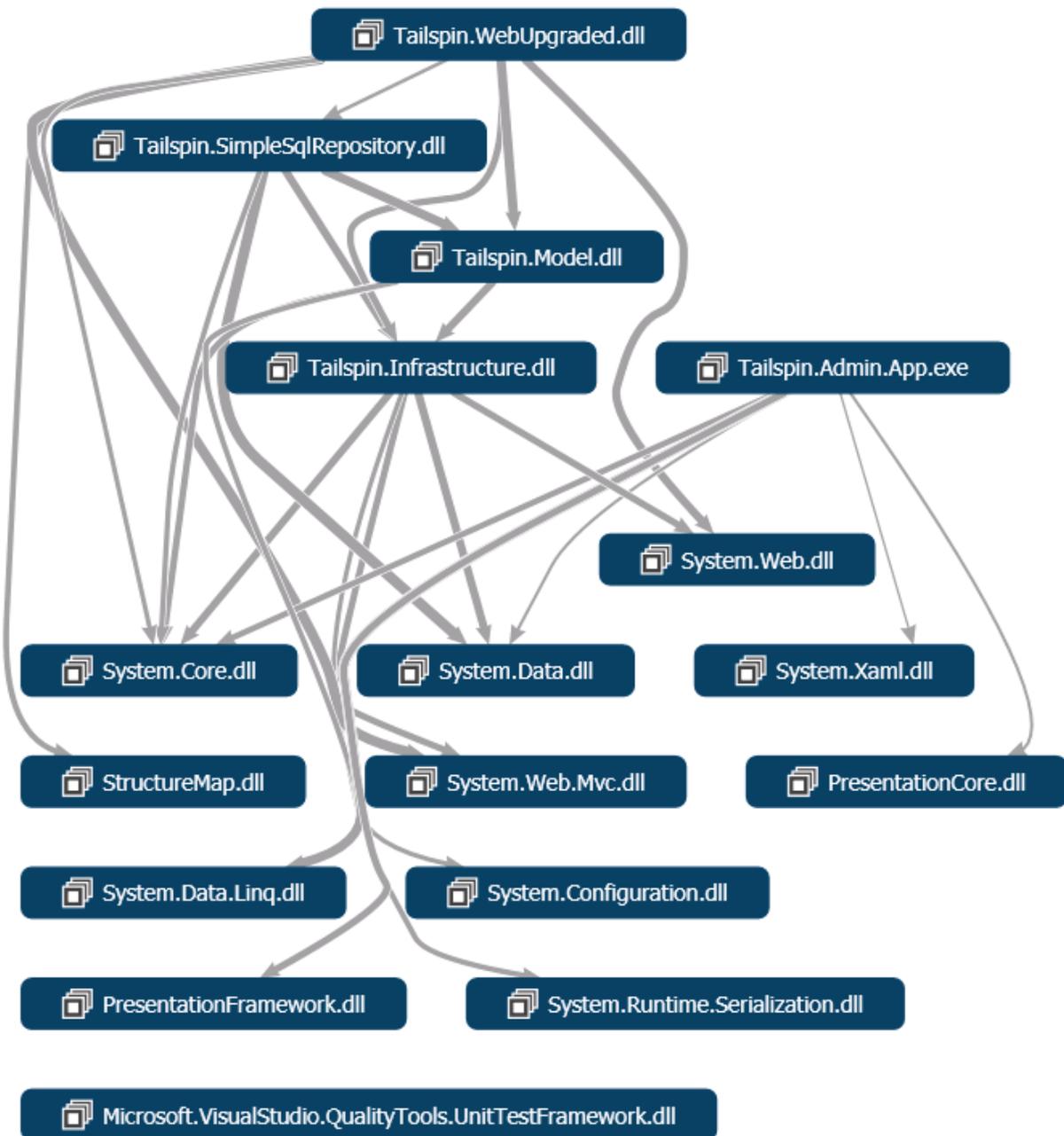


Figure 38 : Le graphique de dépendance simplifié

7. En supprimant le groupe **Externes**, il est plus difficile de distinguer les nœuds externes. Il est possible de mettre une propriété de couleur, par exemple, pour les nœuds qui sont externes. Cliquer sur le bouton + du panneau de légende et sélectionner **Propriétés du nœud** -> **Est externe**.

29 Découverte du code avec les outils d'architecture de code

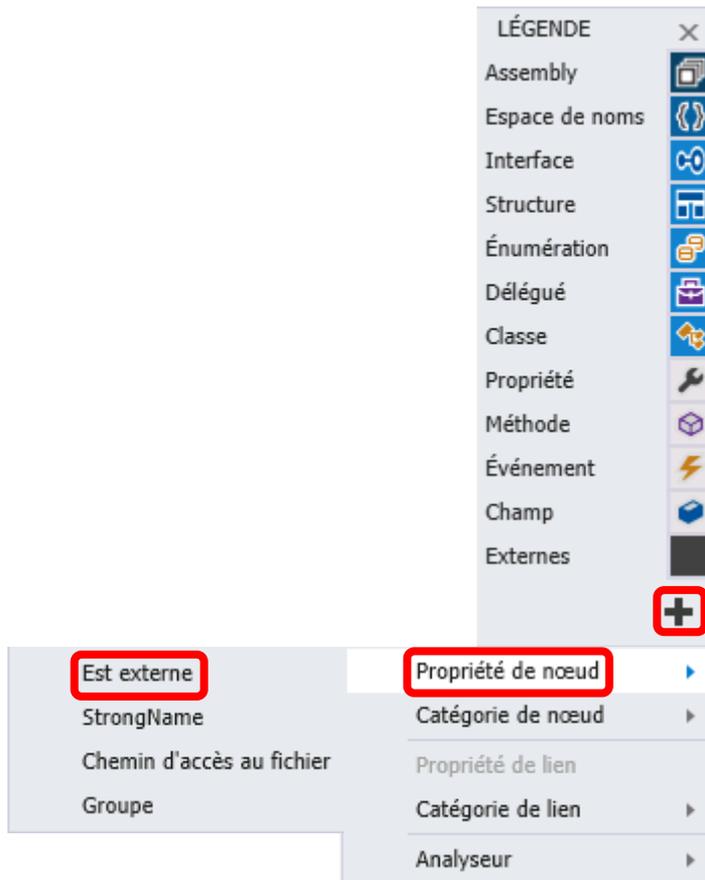


Figure 39 : Création d'une propriété de nœud

8. La propriété **Est externe** est ajoutée au panneau de légende. Faire un clic gauche dans la partie grise à droite de la propriété **Est externe** et sélectionner **Arrière-plan...** pour afficher le **Sélecteur de jeu de couleurs**.

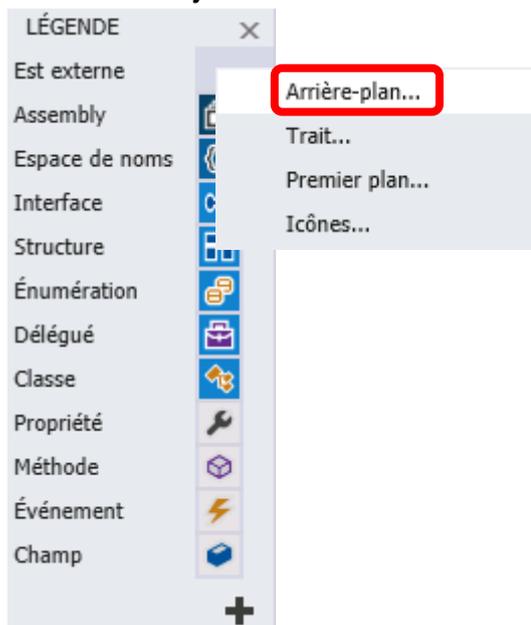


Figure 40 : Définition de la couleur d'arrière-plan

30 Découverte du code avec les outils d'architecture de code

9. Dans le **Sélecteur de jeu de couleurs**, choisir la couleur rouge pour la valeur **True** puis cliquer sur le bouton **OK**.

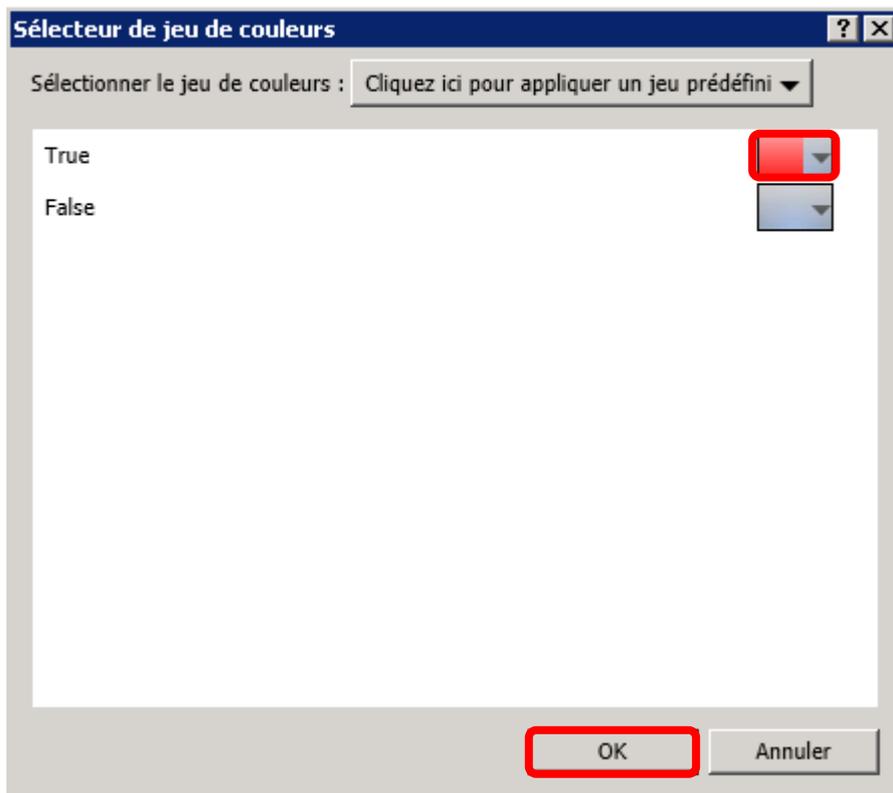


Figure 41 : Choix de la couleur pour les assemblies externes

31 Découverte du code avec les outils d'architecture de code

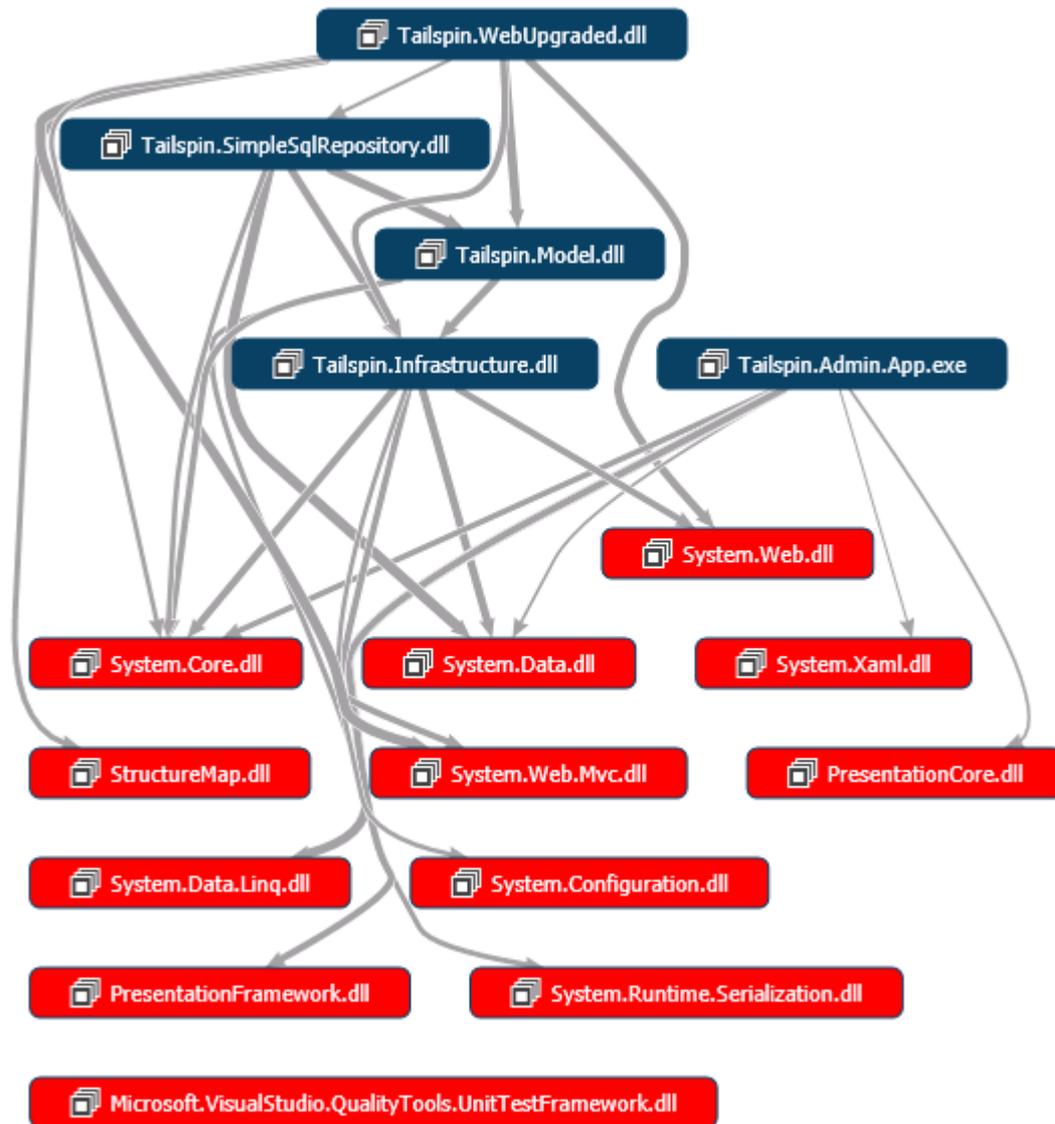


Figure 42 : Affichage des assemblages externes