Atelier 1 : Préparation de la campagne de tests

Sommaire

Introduction

Dans les précédentes versions de Visual Studio les tests avaient déjà une place importante. On retrouvait principalement les tests unitaires, les tests web et les tests de charges, outils puissants et complet permettant aux testeurs techniques de valider un grand nombre des aspects d'une application.

La gamme de produits **Visual Studio 2010** renforce et complète l'offre autour des tests en proposant un nouvel outil pour les testeurs fonctionnels, non obligatoirement techniques : **Microsoft Test and Lab Manager**.

Cet outil, indépendant de Visual Studio permet de mettre en place, d'exécuter et de suivre des campagnes de tests fonctionnels tout en restant connecté au reste de l'équipe par l'intermédiaire de **Team Foundation Server 2010**.

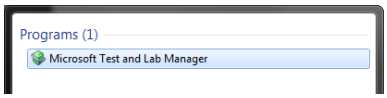
Dans l'atelier précédent (Atelier 0), vous avez vérifié que vous possédiez bien tous les outils nécessaires à la réalisation du coach. Vous avez également créé un projet dans **Team Foundation Server** et envoyé le code source de l'application qui servira de support sur le serveur.

Dans cet atelier, vous allez :

- Apprendre à vous connecter à un projet d'équipe depuis **Microsoft Test and Lab Manager**
- Ajouter un plan de test au projet
- Découvrir l'interface **Microsoft Test and Lab Manager**
- Créer des cas de tests fonctionnels et les organiser dans des suites de tests

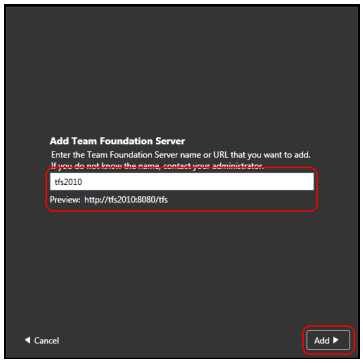
Vue d'ensemble : Microsoft Test and Lab Manager

Lancez **Microsoft Test and Lab Manager** (**MTLM** par la suite) depuis le menu Démarrer :



Si **MTLM** ne trouve pas automatiquement votre serveur **Team Foundation Server** (vous êtes sur une autre machine que celle utilisée dans l'atelier 0, par exemple), cliquez sur **Add Server** :

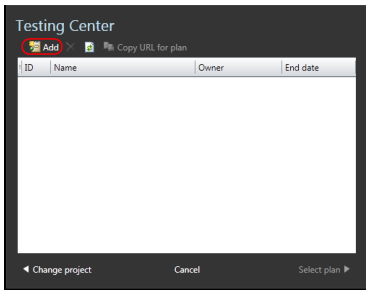
Dans la fenêtre qui s'affiche, entrez le nom de votre serveur **Team Foundation Server**, puis cliquez sur **Add** :



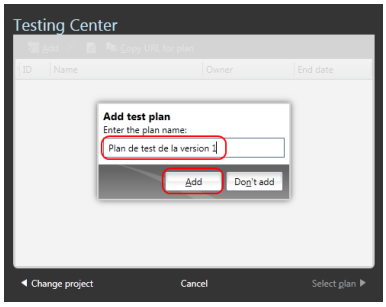
Choisissez le projet « Projet Coach tests » dans la liste des projets disponibles et cliquez sur **Connect now** :

Microsoft Test and Lab Manager est à présent connecté à votre projet. Vous allez pouvoir commencer à travailler. La première chose à faire est de créer un plan de test. Un plan de test est l'équivalent d'une campagne de test. Il permet de définir un ensemble de scénarii ayant pour objectif de valider fonctionnellement une application. C'est également le point d'entrée permettant de retrouver les bugs reportés à l'équipe de développement, suivre la disponibilité des nouvelles versions de l'application, exécuter et valider les cas de tests, etc.

Pour ajouter un plan de tests, cliquez sur le bouton **Add** :

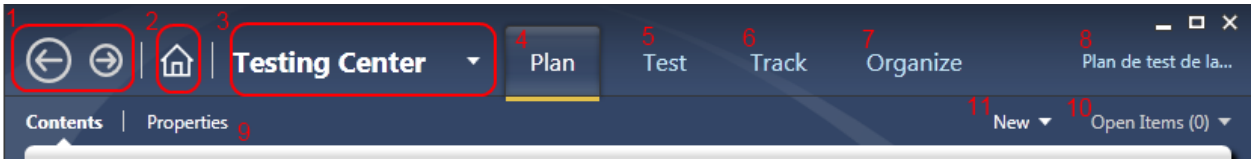


Donnez-lui un nom puis cliquez sur **Add** :



Sélectionnez alors le plan dans la liste et cliquez sur le bouton **Select Plan**. Une fois ceci fait, vous arrivez dans l'interface principale de Microsoft Test and Lab Manager. A partir de maintenant, tout ce que vous réaliserez dans l'outil sera fait dans le contexte « Projet d'équipe / Plan de test sélectionné ».

Commençons par nous pencher en détail sur le menu de cet outil qui vous permettra de naviguer facilement au sein de celui-ci :



Vous retrouverez ci-dessous la légende correspondant à la capture d'écran.

1. Zone de navigation : vous permet de revenir en arrière ou d'aller en avant
2. Vous permet de revenir à l'accueil de **Microsoft Test and Lab Manager**
3. Vous permet de passer du **Testing Center** au **Lab Center** (nous définirons ces éléments par la suite)

4. Onglet **Plan** : Permet de créer et d'organiser des suites de tests et d'y ajouter des cas de tests
5. Onglet **Test** : Permet d'exécuter les tests fonctionnels du plan de tests
6. Onglet **Track** : Permet d'assurer le suivi du plan de test, d'assigner une build (version intermédiaire de l'application) au plan de test ou encore de savoir quels sont les tests qu'il faut rejouer après que des corrections aient été apportées à l'application
7. Onglet **Organize** : Permet de gérer et d'organiser les différents plans de tests associés au projet, de gérer les différentes configurations disponibles pour un plan de test ou encore de définir des « Shared Steps ».
8. Permet de changer de plan de test courant
9. Sous-menu avec affichage de l'emplacement courant
10. Liste déroulante affichant les différents formulaires actuellement ouverts.
11. Liste déroulante permettant de créer n'importe quel type de Work Item

Bien entendu, nous reviendrons en détails sur ces éléments par la suite.

MTLM est composé de deux centres de fonctionnalités :

- Le **Testing Center**
- Le **Lab Center**

Dans ce coach nous nous focaliserons principalement sur la partie **Testing Center**, mais nous ferons tout de même quelques écarts dans la partie **Lab Center**, notamment dans le dernier atelier ou vous apprendrez à automatiser l'exécution de tests d'interface graphique dans un environnement dédié

La partie **Test Center** est celle qui va vous permettre la mise en place de suite de tests (unités d'organisation des tests), d'y ajouter des cas de tests, de les exécuter ou encore de faire remonter directement des bugs aux équipes de développement par le biais de **Team Foundation Server**. Il permettra également d'effectuer un suivi rigoureux des campagnes de test. Par exemple, vous réalisez un test fonctionnel sur une application, lequel échoue. A partir des informations recueillies lors du test, vous pourrez créer une fiche de bug dans **Team Foundation Server** et l'assigner à un développeur afin que ce dernier le corrige.

Lorsque celui-ci aura effectué son travail, vous recevrez l'information dans le **Test Center** et vous pourrez ainsi ré-exécuter le test afin de vous assurer que la correction est bien valide.

Vous avez également accès à des rapports détaillés sur les campagnes de tests en cours, ainsi qu'à des pictogrammes vous permettant de connaître l'état d'un test ou d'une campagne d'un simple coup d'œil.

Le **Lab Center** permet de mettre en place une infrastructure de test virtualisée ou physique. Basé sur **SCVMM** (System Center Virtual Machine Manager), ce dernier vous offre la possibilité de réaliser des tests fonctionnels dans différents environnements, représentés par des machines virtuelles, déployées et

montées à chaud. Ainsi, vous allez pouvoir rejouer vos tests fonctionnels automatiquement sur plusieurs machines prenant ainsi en compte toutes les contraintes que peuvent rencontrer vos utilisateurs. Il est également possible de gérer via cet outil un « laboratoire de tests », c'est-à-dire un ensemble de machines physiques sur lesquelles sont installées des agents de test orchestrés par un contrôleur de test. Ce point sera détaillé dans l'atelier 3.

N'hésitez pas à parcourir l'application afin de découvrir les différentes fonctionnalités qu'elle propose.

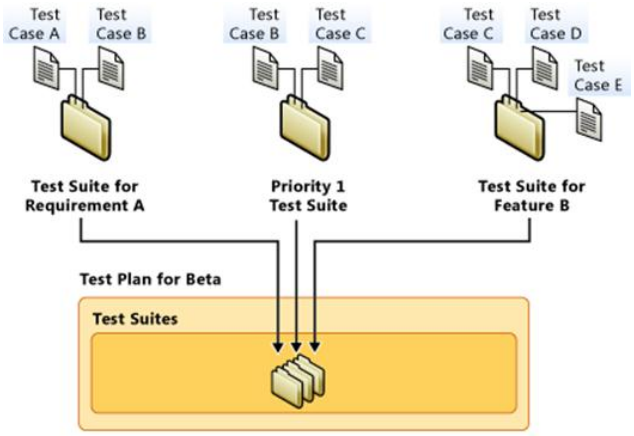
Après ce tour d'horizon de Microsoft Test and Lab Manager, vous allez pouvoir organiser votre plan de test.

Mise en place du plan de test

Nous l'avons vu précédemment, un plan de test est avant tout représenté par des suites de tests, elles même composées de cas de test.

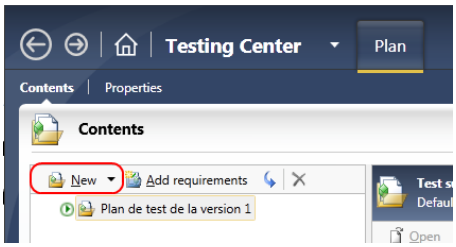
Une suite de tests est un regroupement logique de cas de tests. On peut voir cela comme un répertoire regroupant différents tests fonctionnels qui ont pour but de valider une même fonctionnalité ou une suite de fonctionnalités liées. Il existe trois types de suite de tests :

- Les suites simples : représentées par de simples répertoires
- Les « requirements » : récupérés depuis **Team Foundation Server**, ils peuvent être des « **User Stories** » (Méthode AGILE) ou des « **Requirements** » (Méthode CMMI). Ce sont en fait des scénarii utilisateur ou des exigences qui permettent d'organiser toutes les tâches liées à la réalisation du projet. Souvenez-vous, dans l'atelier 0, vous avez créé un projet possédant par défaut deux « user stories ». Vous allez donc pouvoir les importer directement. L'avantage de ces suites est la création automatique d'un lien entre un Requirement et les cas de test qu'on ajoute dans la suite.
- Les suites basées sur une requête de Test Cases dans Team Foundation Server : il est possible que vous possédiez déjà des cas de tests dans votre serveur. Dans ce cas, il est tout à fait possible d'écrire une requête pour les récupérer et les organiser en suites de tests.

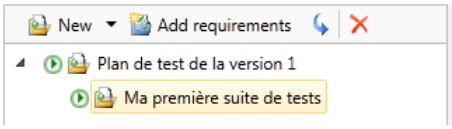


Création d'une suite de tests simple

Pour ajouter une suite de tests simple, cliquez sur le bouton **New** dans le panneau **Contents** de l'onglet **Plan** du **Testing Center** :



Vous pouvez alors donner un nom à cette suite :

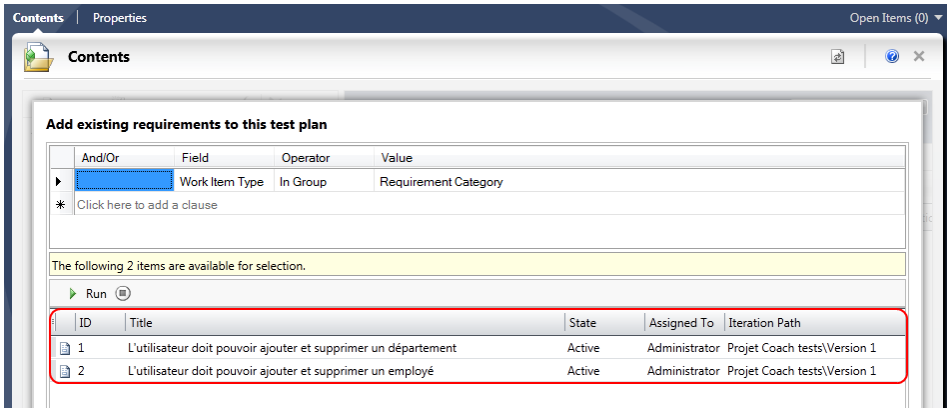


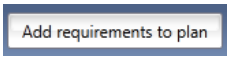
Comme vous pouvez le voir, il n'y a aucune autre propriété, le résultat est exactement le même que lorsque vous créez un répertoire dans votre système de fichiers Windows.

Création d'une suite à partir d'un requirement

En règle générale, on préfèrera utiliser le bouton **Add requirements** qui va permettre de créer des liens logiques entre les scénarii utilisateur et les cas de tests. D'ailleurs Team Foundation Server (et plus particulièrement les modèles de projet Agile) fournissent de base une requête permettant d'identifier toutes les « user stories » n'étant pas liées à des cas de tests. Ceci nous permet d'avoir rapidement la liste des scénarii utilisateurs non couverts par des tests, et donc implicitement ceux pour lesquels la qualité n'est pas vérifiée.

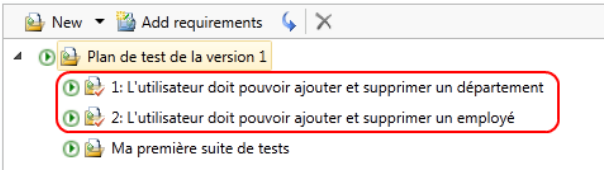
Cliquez sur le bouton **Add requirements** pour récupérer les scénarii utilisateurs dans **Team Foundation Server** :



Sélectionnez les deux éléments dans la liste et cliquez sur le bouton 

en bas à droite de la fenêtre.

Comme vous pouvez le constater ces deux scénarii sont convertis en suite de tests :



Sélectionnez la première suite de tests dans la liste. Dans le panneau de gauche, vous pouvez remarquer que celle-ci est bien associée à un requirement :

Si vous cliquez sur le lien « **(Requirement n)** » vous vous retrouverez directement dans la fiche de la user story à partir de laquelle la suite de tests a été créée.

NB : Les suites de tests peuvent être arborescentes. Ceci permet une meilleure gestion de l'organisation du plan de test

Ajout de cas de test

A présent que vous avez créé vos différentes suites de tests, vous allez devoir créer des cas de tests dans celles-ci. Un cas de test est tout simplement un scénario visant à tester une fonctionnalité précise. Il est important de comprendre qu'un cas de test est un Work Item (élément de travail) stocké dans Team Foundation Server, au même titre que les Tâches, les Scénarii utilisateurs, les Bug, etc.

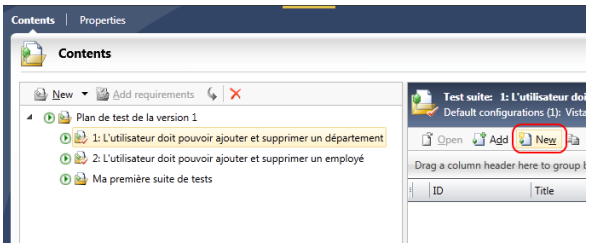
Par exemple, la suite de tests basée sur la user story « **L'utilisateur doit pouvoir ajouter et supprimer un département** » sera composée de deux cas de tests :

- Ajout d'un département
- Suppression d'un département

De la même façon, la suite de tests basée sur la user story « **L'utilisateur doit pouvoir ajouter et supprimer un employé** » sera composée de trois cas de tests :

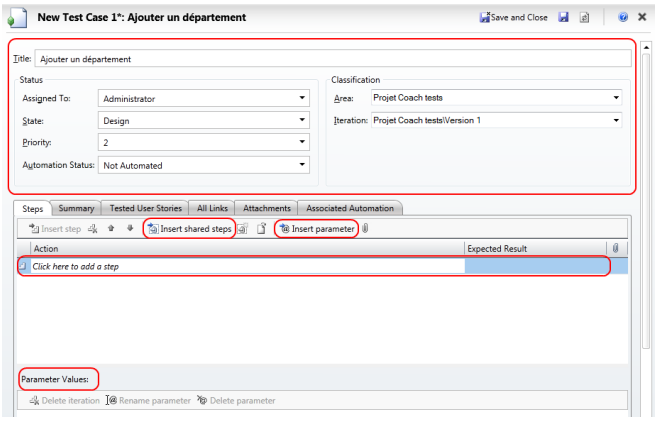
- Ajout d'un employé
- Ajout d'un manager
- Suppression d'un employé

Pour créer un cas de test, sélectionnez une suite dans la liste de gauche, puis cliquez sur le bouton **New** dans le panneau de droite :



*NB : il est également possible d'ajouter des cas de tests déjà existants dans **Team Foundation Server** en cliquant sur le bouton **Add**.*

Une fiche va alors s'afficher et vous permettre de créer un cas de test et de l'enregistrer dans **Team Foundation Server**. Comme vous pouvez le constater, cette fiche est composée de plusieurs parties bien distinctes :



Dans la partie haute, on retrouve les informations « générales » concernant le cas de test à savoir :

- **Title** : le nom du cas de test
- **Assigned To** : la personne en charge de sa réalisation
- **State** : l'état de la fiche (Design = en cours de création, Ready = Prêt à être exécuté)
- **Priority** : la priorité de réalisation du test
- **Automation status** : test automatisé ou non (NB : l'automatisation se fait à partir de Visual Studio 2010)
- **Area** : zone de travail dans laquelle stocker le cas de test dans TFS
- **Iteration** : itération à laquelle est rattaché le cas de test dans TFS

*NB : toutes ces informations vous sont certainement connues si vous avez déjà manipulé des tâches, bugs etc…dans **Team Foundation Server**, puisque, pour rappel, au même titre que ces derniers, les cas de tests sont des Work Items !*

En dessous de cette première zone, on retrouve une zone divisée en onglets, dont le premier nous intéresse tout particulièrement ici, puisque c'est celui qui va permettre de définir les différentes étapes que le testeur devra réaliser.

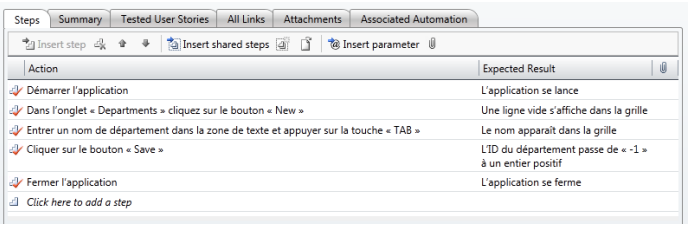
Une étape est définie par deux informations majeures :

- **Action** : l'action que doit réaliser le testeur
- **Expected Result** : le résultat attendu

Par exemple, on pourrait avoir ceci pour le cas de test « Ajouter un département » :

Action	Expected Result
Démarrer l'application	L'application se lance
Dans l'onglet « Departments » cliquez sur le bouton « New »	Une ligne vide s'affiche dans la grille
Entrer un nom de département dans la zone de texte et appuyer sur la touche « TAB »	Le nom apparaît dans la grille
Cliquer sur le bouton « Save »	L'ID du département passe de « -1 » à un entier positif
Fermer l'application	L'application se ferme

Plutôt que de recopier ce tableau ligne par ligne et cellule par cellule dans la fiche du cas de test en cours de création, vous pouvez directement copier toutes les lignes et les coller en sélectionnant la ligne complète dans le tableau correspondant dans MTLM :



Ainsi, si vous possédez déjà des fiches de tests fonctionnels organisées sous forme de tableaux, vous allez pouvoir les intégrer très rapidement dans ce nouvel outil !

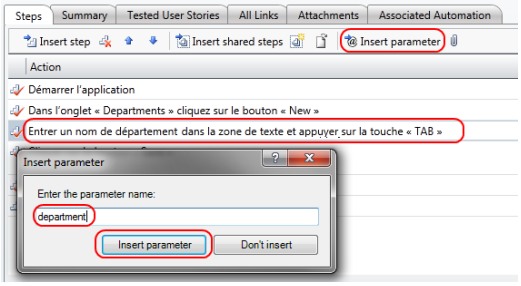
Pour information, il est également possible d'importer un ensemble cas de test à partir d'Excel via un outil nommé « Test Case Migrator ». Il a pour objectif de vous permettre d'importer vos cas de tests existants pour faciliter la migration à partir d'un autre outil.

Comme pour la plupart des **Work Items** dans **Team Foundation Server 2010**, vous avez également la possibilité d'ajouter d'autres informations au cas de tests (liens, pièces jointes, description etc…).

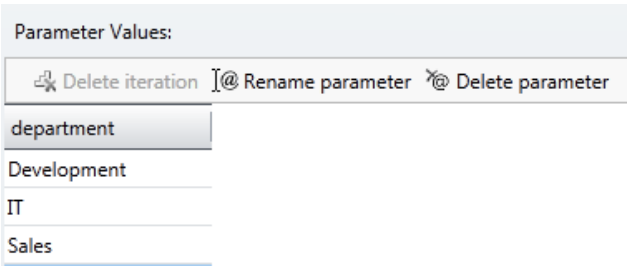
Paramétrer un test fonctionnel

La dernière zone de cette fiche de test vous permet d'ajouter des valeurs de paramètre au test fonctionnel. En effet, il peut être intéressant de jouer plusieurs itérations d'un même test avec différentes valeurs « métier ». Dans le test renseigné ci-dessus, nous pourrions facilement paramétrer la 3ème étape afin de fournir différents noms de département au testeur.

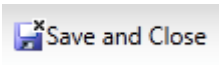
Pour ce faire, placez-vous sur l'étape que vous souhaitez paramétrer puis cliquez sur le bouton **Insert parameter**. Dans la fenêtre qui s'affiche, renseignez le nom du paramètre puis cliquez sur **Insert parameter** :



Comme vous pouvez le constatez, la chaîne de caractères « **@department** » vient s'insérer en fin de ligne. Vous remarquerez également que la zone « **Parameter Values** » en bas de la fenêtre a évolué puisqu'une colonne « **department** » a été créée permettant de renseigner les différentes valeurs de ce paramètre :



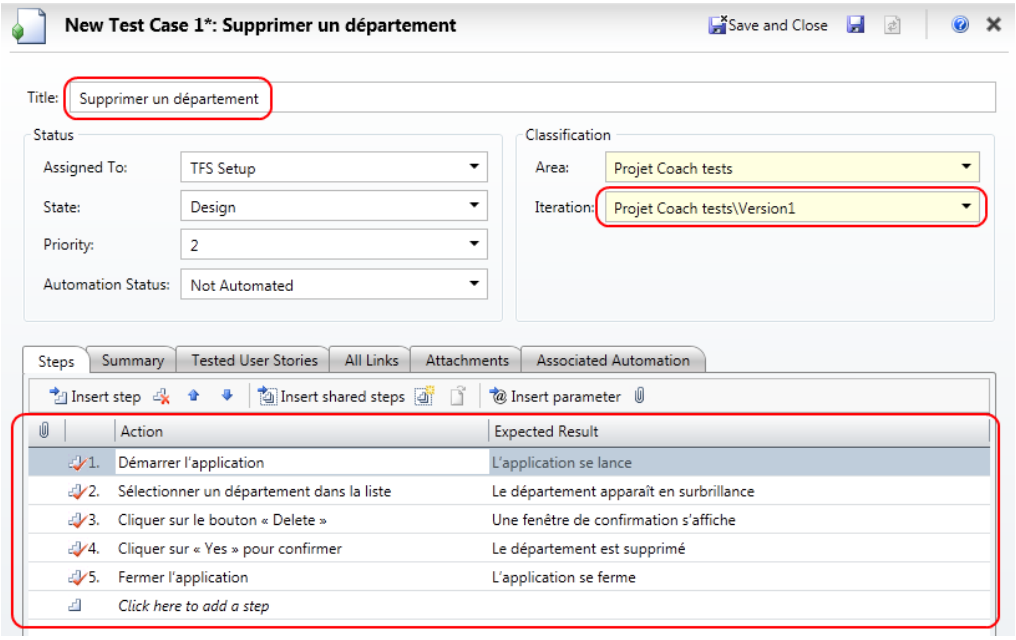
Ajoutez alors trois paramètres : Development, IT et Sales comme le montre l'écran ci-dessus.

Vous pouvez à présent enregistrer ce cas de test en cliquant sur le bouton 

en haut à droite de la fiche.

De la même manière, ajouter un nouveau cas de test à la suite « **L'utilisateur doit pouvoir ajouter et supprimer un département** ». Nommez le « **Supprimer un département** ». Voici son contenu :

Action	Expected Result
Démarrer l'application	L'application se lance
Sélectionner un département dans la liste	Le département apparaît en surbrillance
Cliquer sur le bouton « Delete »	Une fenêtre de confirmation s'affiche
Cliquer sur « Yes » pour confirmer	Le département est supprimé
Fermer l'application	L'application se ferme



Comme précédemment, cliquez sur le bouton **Save and Close** pour enregistrer le cas de test dans TFS.

Utilisation des Shared Steps

A présent nous allons créer 3 cas de test pour la suite « **L'utilisateur doit pouvoir ajouter et supprimer un employé** ».

Tout d'abord, voici le descriptif des trois cas suscités :

1. **Ajouter un employé** :

Action	Expected Result
Démarrer l'application	L'application se lance
Cliquer sur l'onglet « Employees »	Le formulaire de gestion des employés s'affiche
Cliquer sur le bouton « New »	Une ligne s'affiche dans la grille
Renseigner le nom de l'employé et appuyer sur « TAB »	Le nom est renseigné dans la grille
Renseigner le prénom de l'employé et appuyer sur « TAB »	Le prénom est renseigné dans la grille
Renseigner la date de naissance de l'employé et appuyer sur « TAB »	La date de naissance est renseignée dans la grille
Cliquer sur le bouton « Save »	L'ID de l'employé passe de « -1 » à un entier positif
Quitter l'application	L'application se ferme

2. **Ajouter un manager**

Action	Expected Result
Démarrer l'application	L'application se lance
Cliquer sur l'onglet « Employees »	Le formulaire de gestion des employés s'affiche

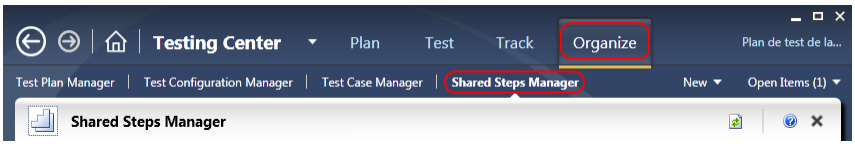
Cliquer sur le bouton « New »	Une ligne s'affiche dans la grille
Renseigner le nom de l'employé et appuyer sur « TAB »	Le nom est renseigné dans la grille
Renseigner le prénom de l'employé et appuyer sur « TAB »	Le prénom est renseigné dans la grille
Renseigner la date de naissance de l'employé et appuyer sur « TAB »	La date de naissance est renseignée dans la grille
Cocher la case « Is Manager »	La cache se coche également dans la grille
Cliquer sur le bouton « Save »	L'ID de l'employé passe de « -1 » à un entier positif
Quitter l'application	L'application se ferme

3. Supprimer un employé

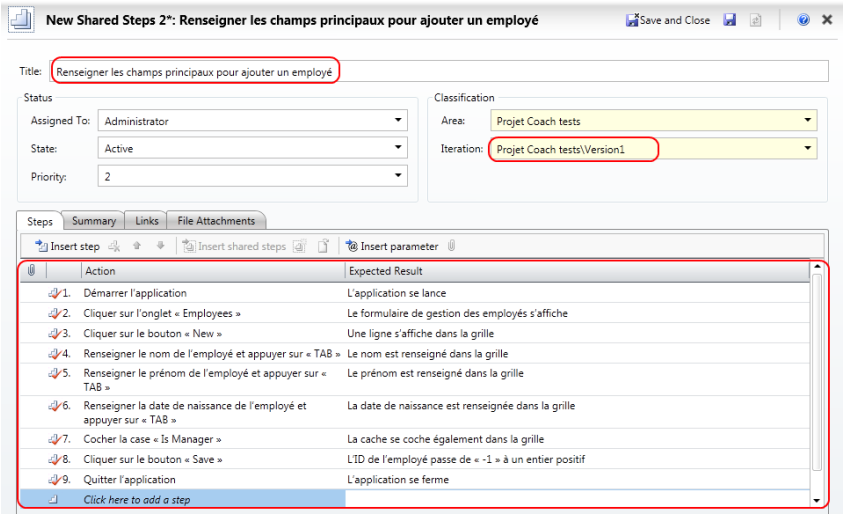
Action	Expected Result
Démarrer l'application	L'application se lance
Cliquer sur l'onglet « Employees »	Le formulaire de gestion des employés s'affiche
Sélectionner un employé dans la liste	L'employé apparaît en surbrillance
Cliquer sur le bouton « Delete »	Une fenêtre de confirmation s'affiche
Cliquer sur « Yes » pour confirmer	L'employé est supprimé
Quitter l'application	L'application se ferme

Comme vous pouvez le constater, les 6 premières étapes des tests 1 et 2 leurs sont communes. Vous allez donc pouvoir les regrouper dans ce que l'on appelle des **Shared Steps**. Celles-ci permettent de regrouper des étapes entre elles afin de les réutiliser dans plusieurs cas de test, plutôt que de devoir les réécrire à chaque fois.

Pour gérer les **Shared Steps**, cliquez sur l'onglet **Organize** dans MTLM puis cliquez sur le sous menu **Shared Steps Manager** :



Dans l'écran qui s'affiche, cliquez sur le bouton **New** afin d'ajouter une nouvelle shared steps :

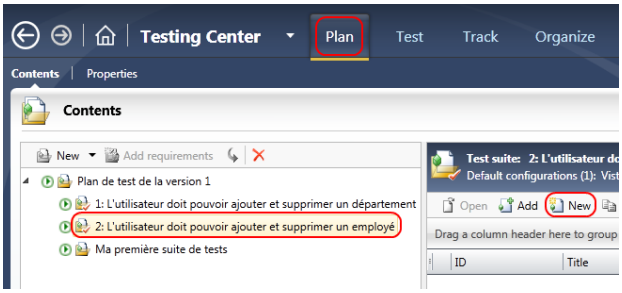


Une fois les champs remplis comme ci-dessus, vous pouvez cliquer sur le bouton **Save and Close**.

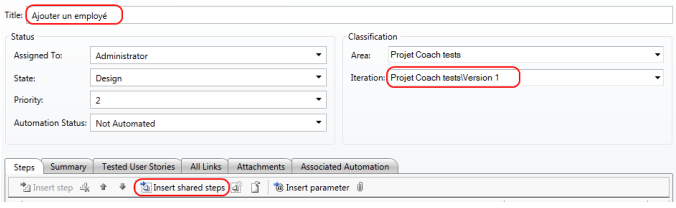
NB : là encore, vous pouvez directement copier / coller les étapes dans le tableau de Steps depuis ce document afin de ne pas avoir à les réécrire toutes.

NB 2 : vous avez également pu remarquer qu'il était possible de paramétrer les shared steps comme vous avez appris à le faire plus haut. Dans ce cas, dès que vous réutiliserez la shared step, celle-ci embarquera les paramètres dans le cas de test qui la contient.

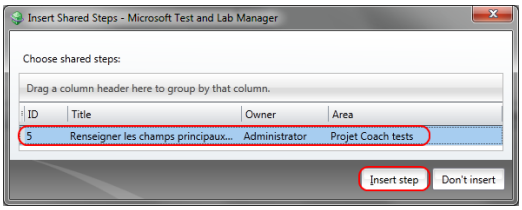
Une fois la sauvegarde effectuée, retournez dans l'onglet **Plan** et sélectionnez la suite de test « L'utilisateur doit pouvoir ajouter et supprimer un employé ». Cliquez alors sur le bouton **New** pour y ajouter un nouveau cas de test :



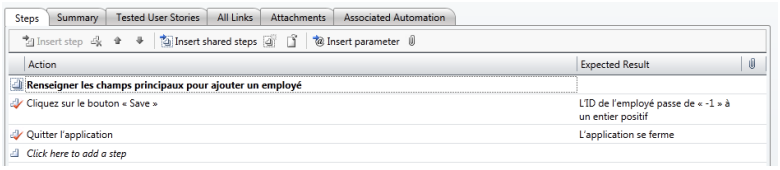
Dans l'écran qui s'affiche renseignez les informations générales, à savoir le titre ainsi que l'itération courante puis, au niveau de la zone de création des étapes, cliquez sur le bouton **Insert shared steps** :



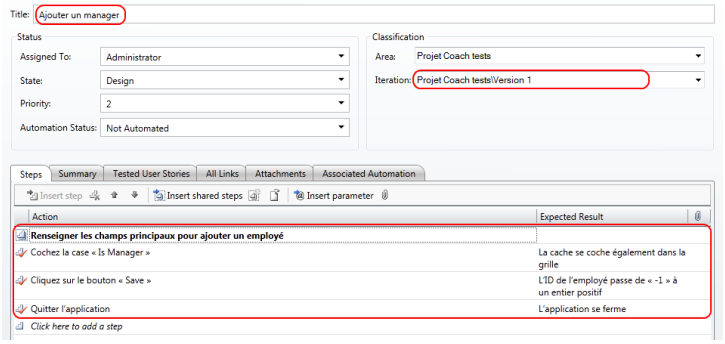
Dans la fenêtre de sélection, choisissez la shared step créée ci-dessus puis cliquez sur **Insert step** :



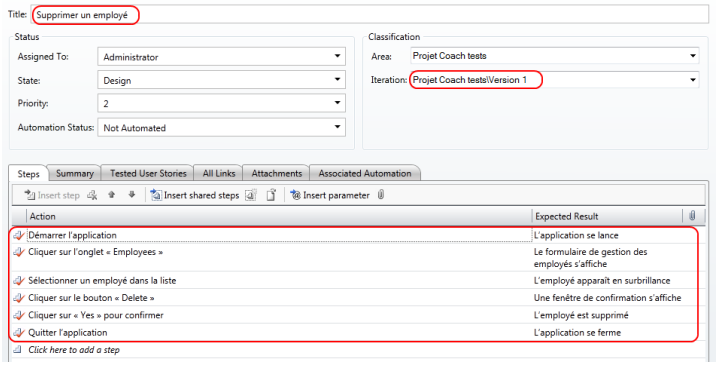
Vous pouvez ensuite copier / coller les deux dernières actions et cliquer sur le bouton **Save and Close** pour valider la création du cas de test :



Répéter l'opération pour le cas de test « Ajouter un manager » :



Enfin créez le dernier cas de test « Supprimer un employé » :



Si l'on résume à ce stade de l'exercice, vous devez avoir 3 suites de tests organisées comme ci-dessous :

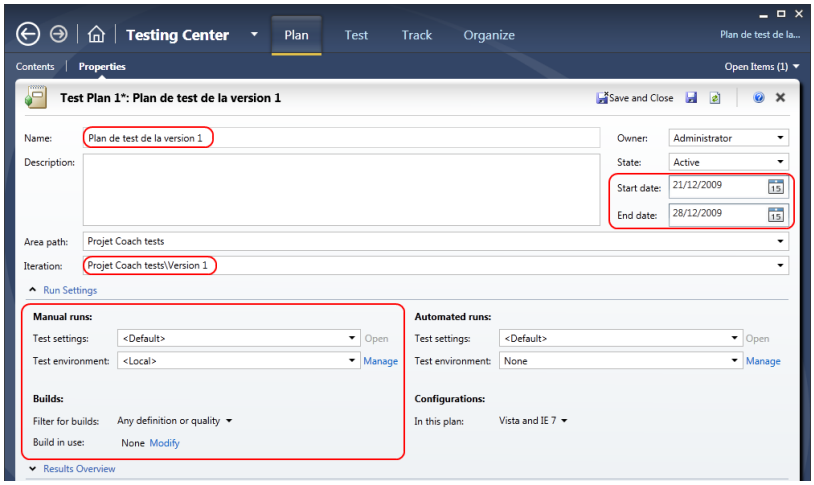
1. L'utilisateur doit pouvoir ajouter et supprimer un département (Requirement)
    a. Ajouter un département
    b. Supprimer un département
2. L'utilisateur doit pouvoir ajouter supprimer un employé (Requirement)
    a. Ajouter un employé

    b. Ajouter un manager
    c. Supprimer un employé
3. Ma première suite de test (Simple)

Les différentes suites de tests du plan étant créées, vous allez pouvoir passer à la dernière étape de cet atelier, la configuration et la planification du plan de test.

Configuration et planification du plan de test

Pour commencer, rendez-vous dans l'onglet **Plan** de Microsoft Test and Lab Manager et cliquez sur le sous-onglet « Properties » :



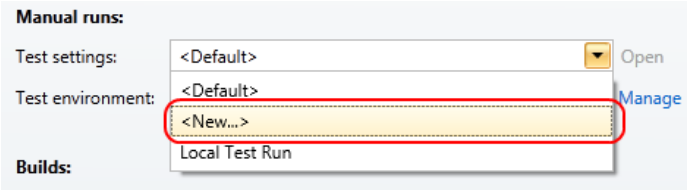
Vous allez retrouver toutes les informations propres au plan de test actuellement ouvert :

- Son nom
- Son état
- Ses dates de début et date de fin, qui permettront de planifier la campagne de test
- La zone et l'itération sur lesquelles il s'applique
- Les paramètres de configuration de l'exécution des tests, notamment pour les tests manuels.

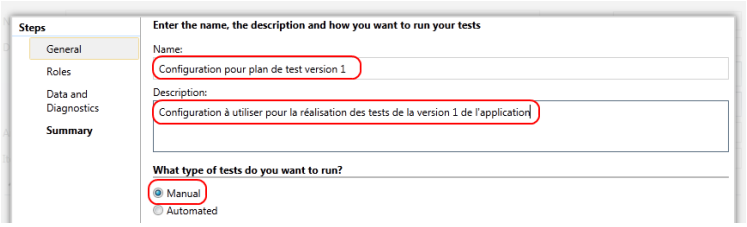
Création d'une configuration de test

Lors de la réalisation d'un test fonctionnel, il est possible de demander à Microsoft Test and Lab Manager d'enregistrer un grand nombre d'informations sur l'environnement, l'exécution…Pour cela, ce dernier utilise des collecteurs de données. Dans cette partie, vous allez apprendre à configurer votre plan de test pour définir quelles sont les données que vous souhaitez enregistrer tout au long des tests fonctionnels.

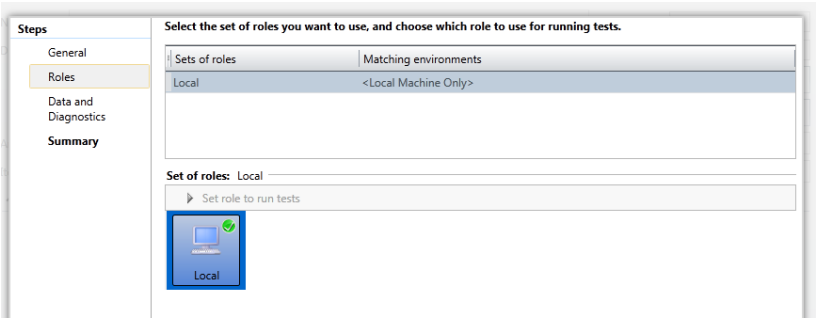
Pour cela, déroulez la liste « Test settings » configurée sur **<Default>** et cliquez sur l'item **<New…>** :



Un assistant s'ouvre alors pour vous aider à créer ces paramètres de test. Dans le premier écran, saisissez un nom, une description et le type de test auquel va s'appliquer cette configuration :

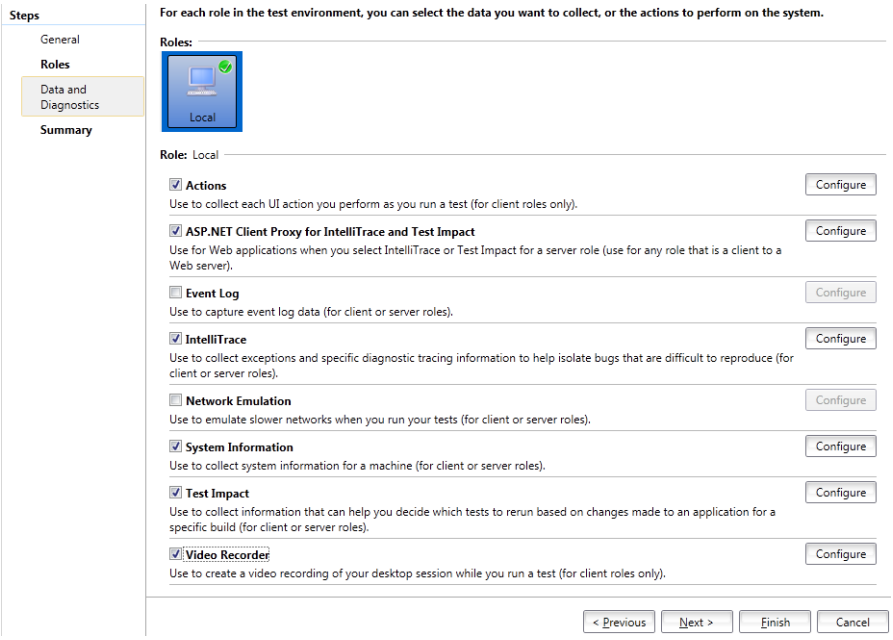


L'écran suivant vous permet de définir des rôles. Pour l'instant, vous utiliserez le rôle « Local » par défaut, c'est-à-dire la machine sur laquelle vous utilisez MTLM. Vous apprendrez dans l'atelier 3 de ce coach à définir d'autres rôles pour déporter l'exécution de vos tests.



Dans l'écran **Data and Diagnostics**, vous allez pouvoir paramétrer les différentes données qui seront collectées lors de l'exécution d'un test manuel.

En effet, lorsque vous effectuerez votre test, vous aurez la possibilité d'enregistrer diverses informations relatives aux actions que vous effectuées sur l'application, de faire un enregistrement vidéo, de récupérer des données IntelliTrace ™ qui permettront aux développeurs de reproduire plus facilement un bug.



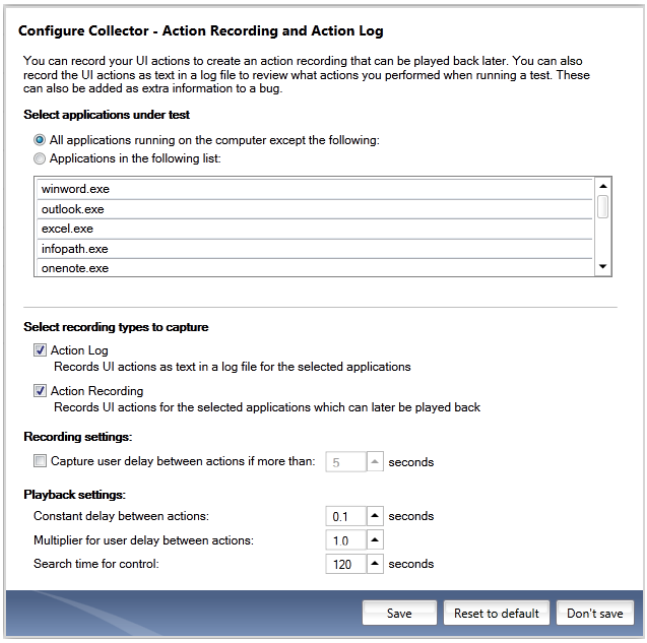
Assurez-vous de sélectionner les mêmes collecteurs que dans la capture d'écran ci-dessus.

Nous allons maintenant passer en revue les différents collecteurs :

Actions

MTLM offre la possibilité d'enregistrer toutes les actions que vous effectuez lors de la réalisation d'un test fonctionnel dans l'optique de pouvoir le rejouer automatiquement. Ainsi, lorsque le testeur cliquera sur un bouton, déplacera une fenêtre, saisira une information dans une zone de texte (etc.) ces actions seront enregistrées.

L'Action Log est un fichier d'historique qui recense toutes les actions que vous avez effectuées pendant le test alors que **l'Action Recording** et celui qui va permettre de le rejouer.



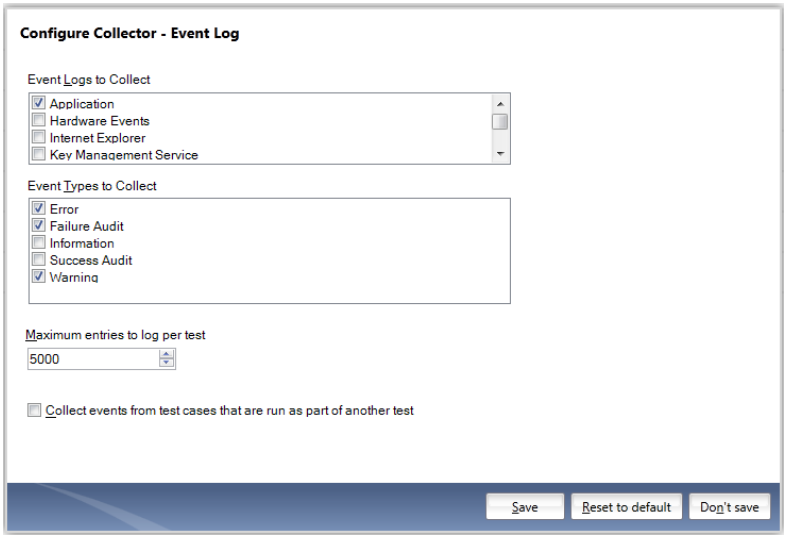
ASP.NET Client Proxy for IntelliTrace™ and Test Impact

Ce collecteur est nécessaire lorsque vous effectuez un test fonctionnel sur une application Web hébergée sur un serveur IIS (Internet Information Service) distant, puisqu'elle va permettre de se connecter à ce dernier au travers d'un proxy afin d'y récolter des informations relatives à l'IntelliTrace™ et aux tests impactés.

Elle ne nécessite pas de configuration particulière.

Event Log

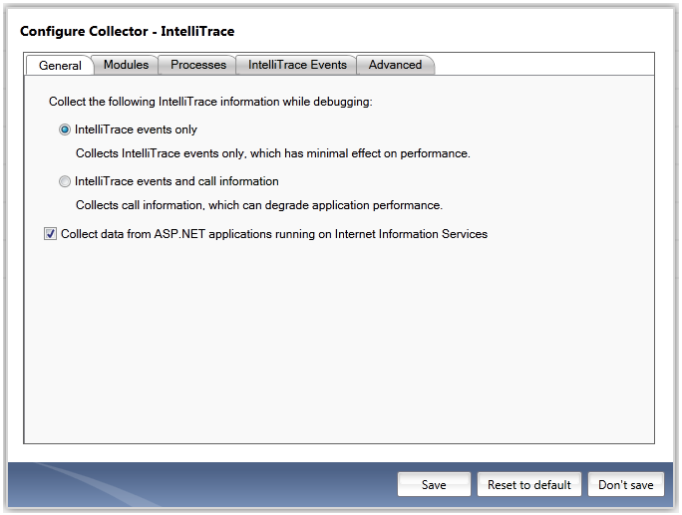
Permet de collecter les différents événements levés dans le gestionnaire d'événements de Windows lors de l'exécution du test fonctionnel. Sa fenêtre de configuration permet de choisir de manière précise quels sont les journaux d'événements qui vous intéressent :



IntelliTrace™

L'IntelliTrace™ est une des nouveautés de la gamme Visual Studio 2010. Elle permet, à partir d'un fichier d'enregistrement de toutes les actions et événements survenus lors de l'exécution d'un programme, de se remettre à tout moment en mode debug dans les conditions d'exécution de l'application à l'instant t auquel un bug s'est produit.

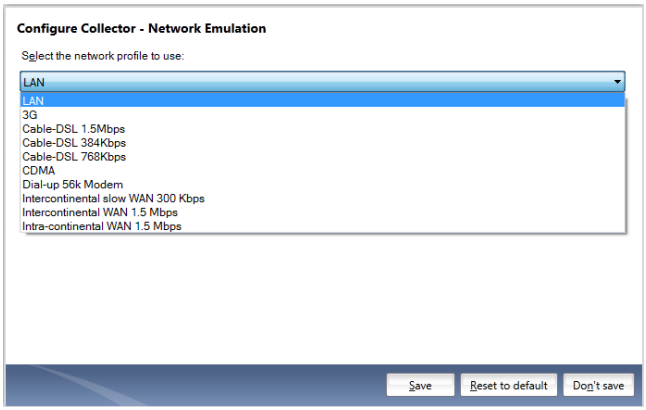
En effet, tout au long de l'exécution d'un test manuel, si cette option est activée, **MTLM** va collecter les différents événements qui se produise, enregistrer les différentes piles d'appels, etc… Ainsi, si un bug se produit (par exemple, une Exception non gérée), le développeur pourra en quelques clics se remettre dans les mêmes conditions que le testeur et donc corriger le bug.



Network Emulation

Cette fonctionnalité permet d'émuler un type de connexion réseau lors du test d'une application.

On pourra notamment brider la vitesse de la connexion réseau, mais également définir le taux de perte de paquets, la latence, la taille de la file d'attente, etc.



System Information

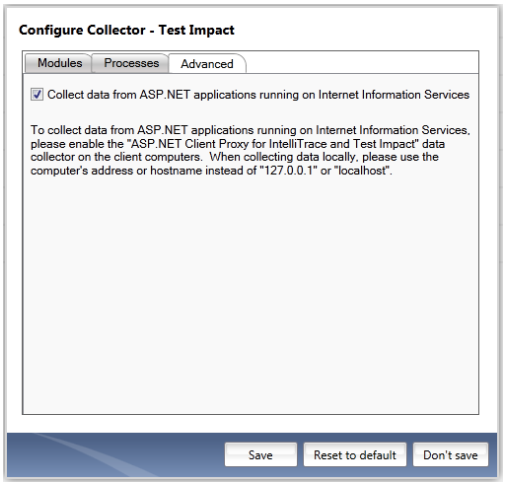
Permet de récolter les informations système du poste sur lequel le test est effectué. Ainsi si une anomalie se produit, celles-ci seront associées au Bug qui sera remonté au développeur et ce dernier pourra connaître la configuration de la machine sur laquelle le bug a eu lieu (mémoire, processeur, disque, OS, architecture…)

Cette fonctionnalité ne nécessite pas de configuration particulière.

Test Impact

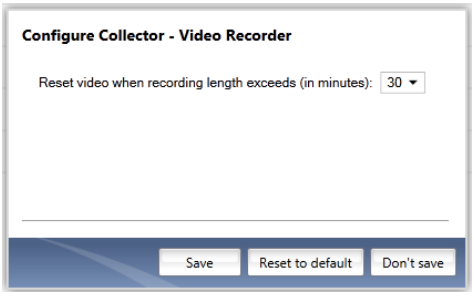
Lorsqu'un test est joué, qu'il soit manuel ou automatisé, la couverture de code est analysée et enregistrée. Grâce à cette information, on sait à tout moment quel test a exécuté quelle(s) ligne(s) de code. De ce fait, si le code en question est modifié par un développeur, on peut aisément connaitre la liste des tests (automatisés ou manuels) à exécuter pour valider la modification. Cette fonctionnalité est appelée : « Test Impact » ou « Analyse de tests impactés ». Elle permet de mettre en place un premier niveau de non-régression.

Pour pouvoir enregistrer les informations permettant d'exploiter le test impact, il faut activer ce collecteur.



Video Recorder

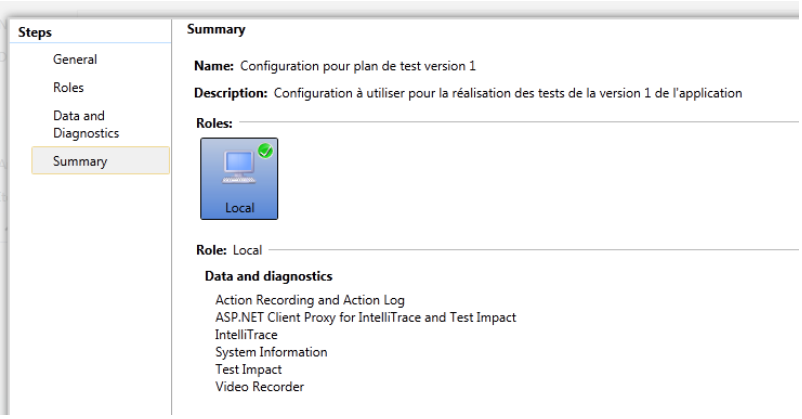
Permet d'enregistrer une vidéo de toutes les actions qui sont effectuées par le testeur pendant le test fonctionnel. Cette vidéo sera ensuite découpée étape par étape et chacune de ces étapes seront accessibles via des signets. Ainsi, il sera beaucoup plus facile de voir quel enchaînement d'actions est responsable de la découverte d'un bug.



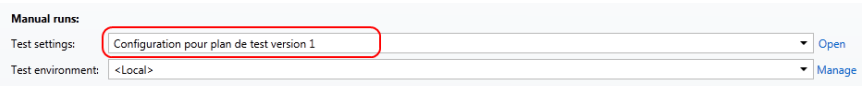
Comme vous avez pu le voir, toutes ces fonctionnalités participent grandement à l'amélioration de la communication entre testeurs et développeurs. Fini le temps ou le testeur remonte un bug que le développeur n'arrive pas à reproduire et qu'il considère donc comme résolu.

On englobe communément ces fonctionnalités sous les termes de scénario « No more no repro » (Plus jamais de « non-reproductible ») signifiant qu'il ne doit plus être possible de ne pas réussir à reproduire un bug !

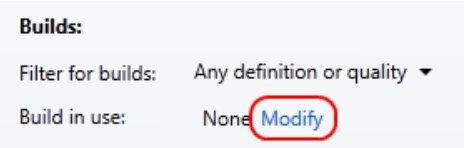
Une fois que vous avez configuré toutes ces fonctionnalités, vous pouvez passer à la dernière étape, **Summary**, qui résume la configuration :



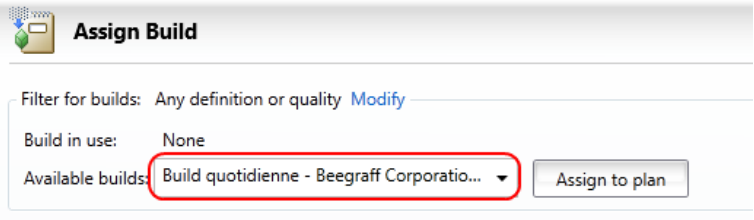
Cliquez alors sur **Finish** pour valider cette dernière. Vérifiez que la configuration a été sélectionnée comme étant celle sous laquelle les tests doivent s'effectuer :



Le sous menu **Properties** vous permet également d'assigner une build au plan de test. Pour ce faire, cliquez sur le lien **Modify** dans la section **Builds** :



Dans la fenêtre qui s'ouvre, vous avez accès à la build que vous avez exécuté lors de la réalisation de l'atelier 0 :



Pour l'assigner au plan de test, cliquez sur le bouton **Assign to plan**. En revenant sur la fenêtre **Properties** vous pourrez constater que celle-ci a bien été affectée et est donc désormais associée au plan de test :

La configuration et l'organisation du plan de test étant terminées, vous pouvez passer à l'atelier suivant : l'exécution des tests fonctionnels !