

Fall 2011



Application Lifecycle & Delivery Management

Dick Wenning

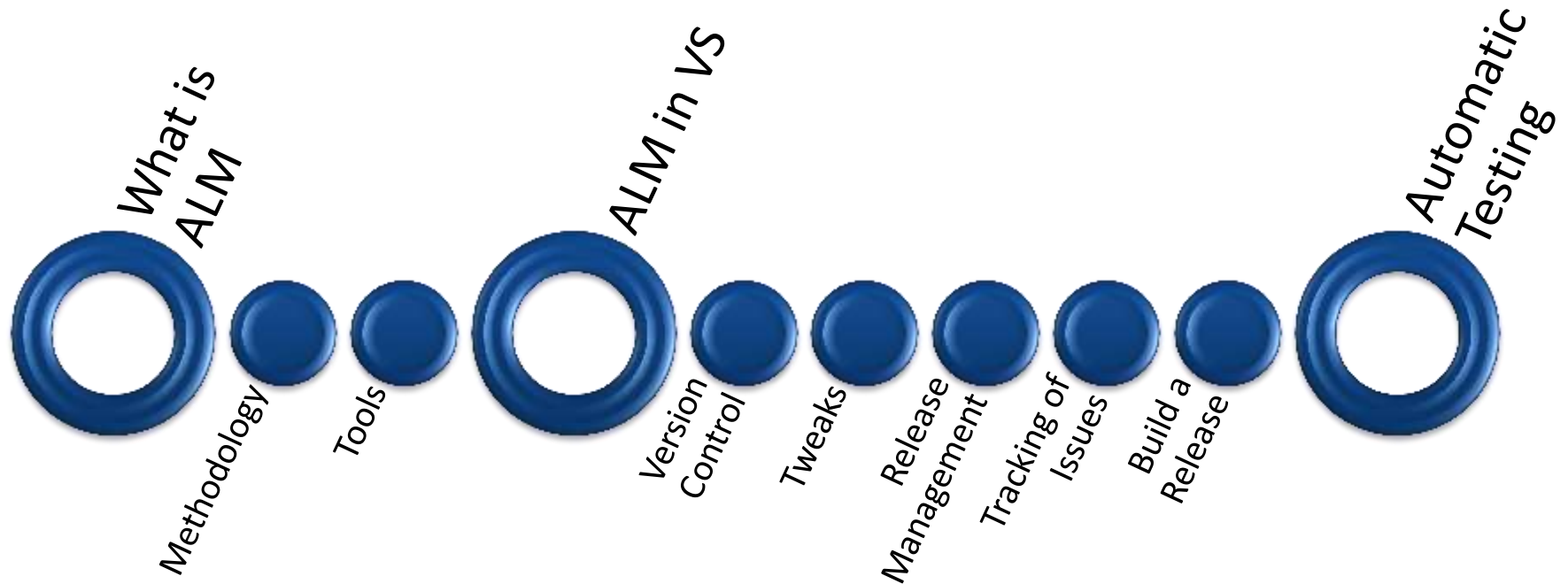
For Microsoft Dynamics AX

<http://www.axstart.com>



***Pro*mentum**

What will we talk about ..



What is ALM?

A continuous process of managing the lifecycle of an application through governance, development and maintenance.



What is ALM?

Marriage of business management and software engineering.



ALM & Methodology

A Methodology is not a product but a way of thinking supported by tools.



ALM Methodology Integration

The Methodology is the front-end process, creating developer tasks in TFS.

Tasks can be connected to a Change Set.
Change Sets can be merged/branched.
A Change Set contains a XPO of the type Table, Class, etc ...

ALM & Visual Studio

VS supports: Version Control, Build & Release Management, Tracking of issues and code changes, Automated Testing.



Do I really need it?

No!

In case that there is never more than 1 developer, who is never ill, able to answer all questions about a 1 year old release and we never have a single point of failure.

ALM & Version control

TFS, Version control, AX integration.

File in repository:

Read-only – AOT tells: **Check out**

Editable – AOT tells: **Checked out**

No xpo – AOT tells: **Create**

Version control parameters

General

● Team Foundation Server

Set up Team Foundation Server options

Team Foundation Server URL:

Team Foundation project name:

Branch folder:

Application root folder:

ALM & Version control

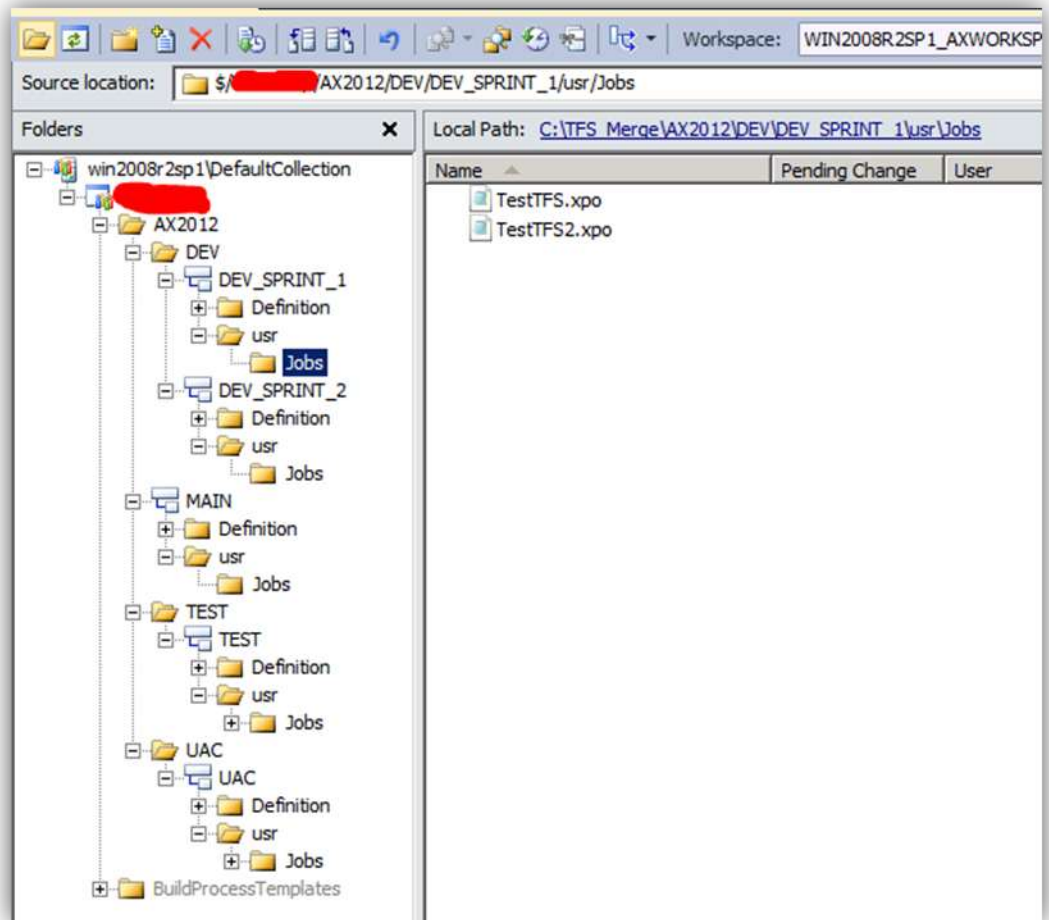
TFS, Version control, AX integration.

The screenshot displays the 'System settings' window with the 'General' tab selected. The 'Version control settings' section is expanded, showing the following configuration:

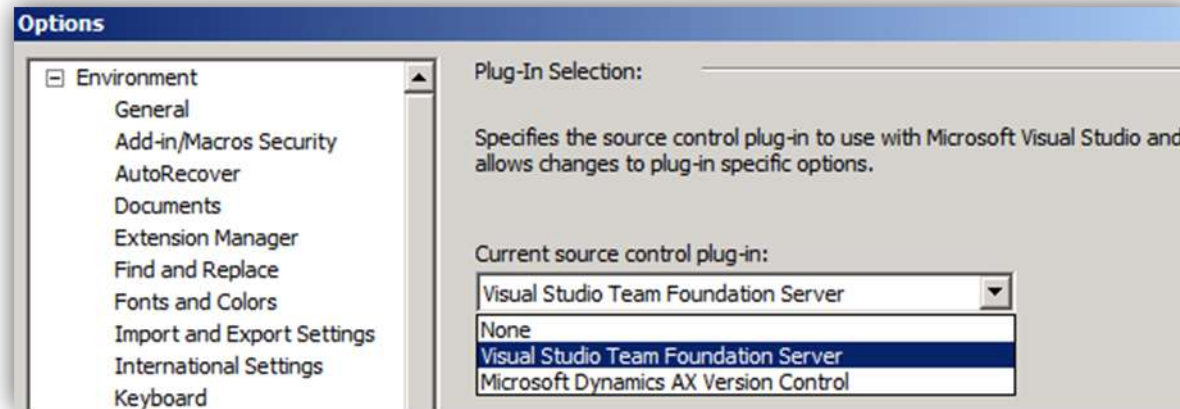
- Version control settings**
 - Status line: My Project
- Best practice settings**
 - Compiler warnings: Accept
 - Compiler to-dos: Accept
 - Compiler errors: Accept
 - Best practice errors: Accept
 - Run title case update:
 - Test project: [Empty dropdown]
- Labels**
 - Label file ID: [Empty dropdown]
 - Label interval
 - From: 0
 - To: 0

TFS Version control

Checks in:
XPO of
Tables,
Classes,
Forms, etc ...



TFS Version control



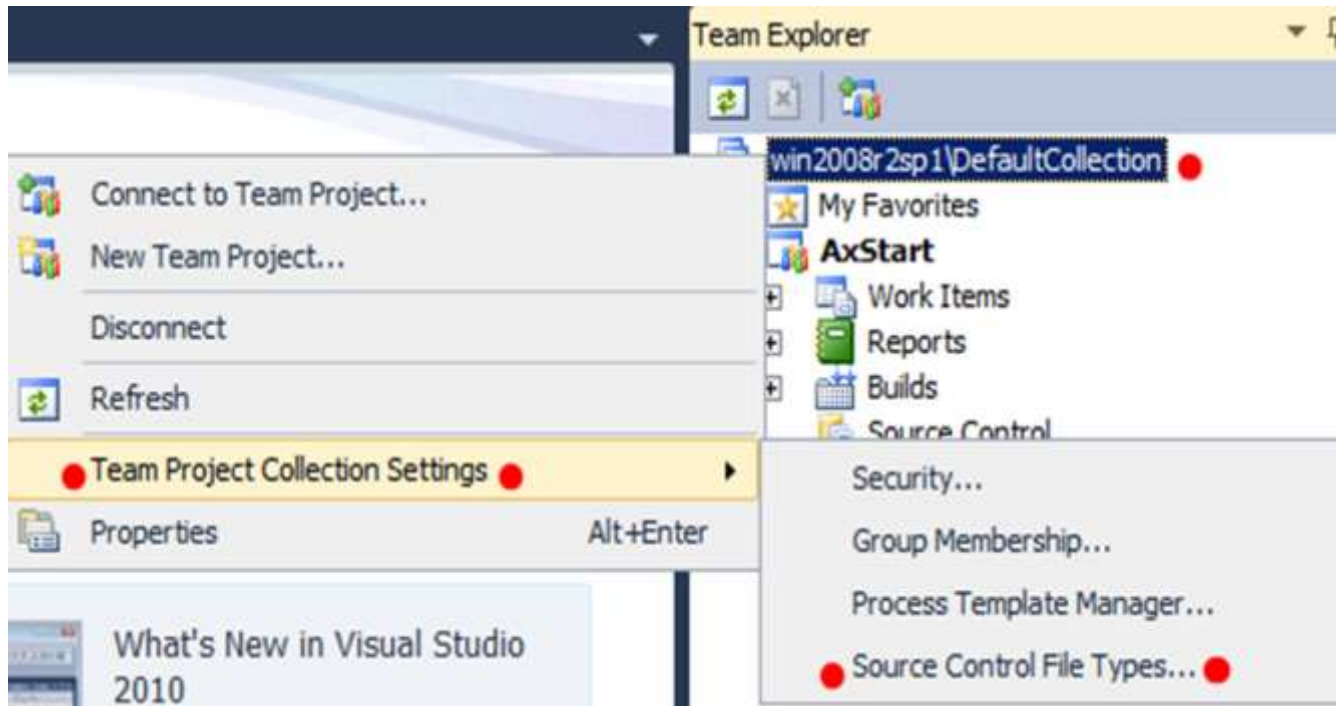
Set Dynamics AX for code of AOT,
Set TFS for Branching, Merging and
Building.

ALM Tweaks

Optimized for Microsoft Dynamics AX 2012

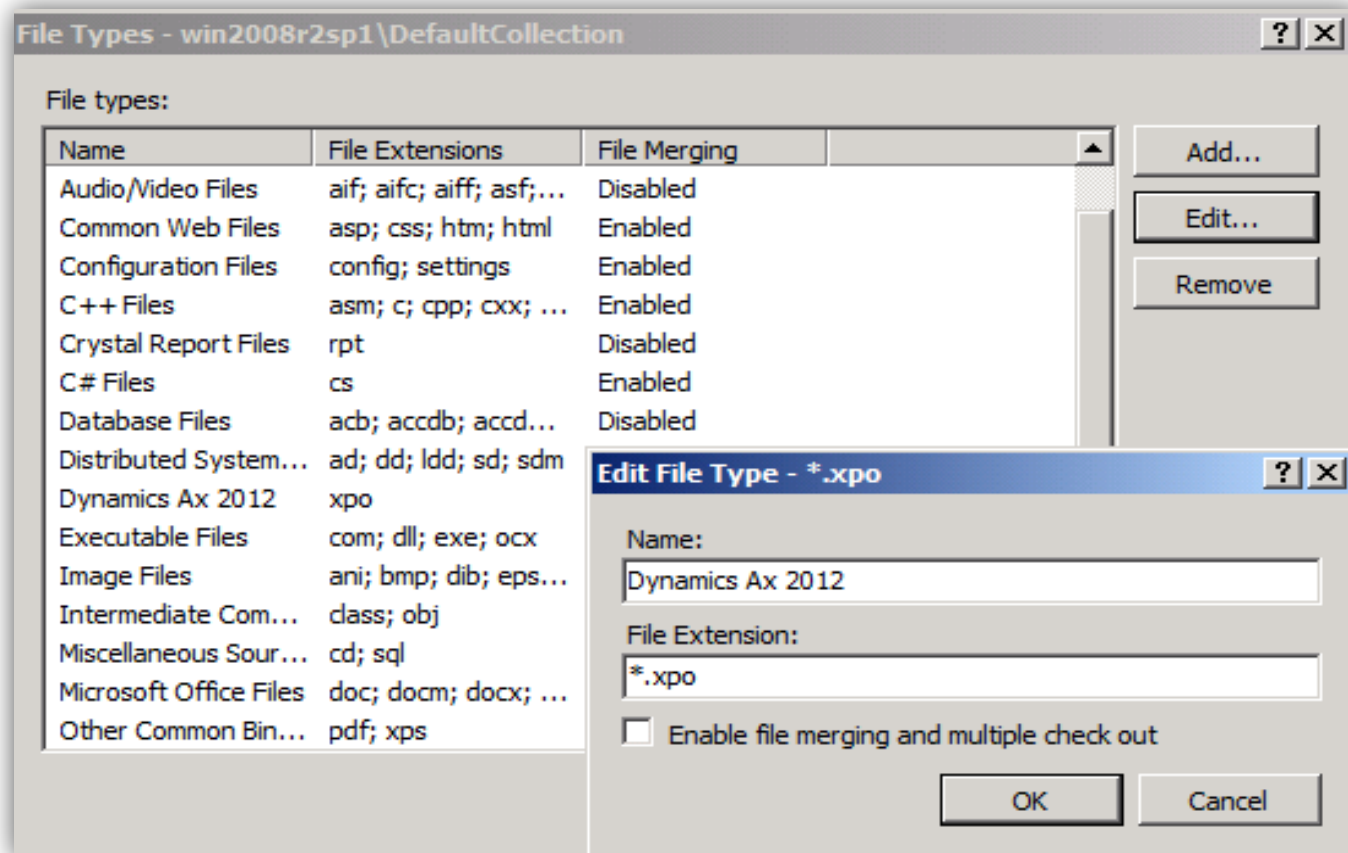
- Checking code in is **NOT** exclusive,
- Checking code in will not update the AOT and repository with latest version.
- TFS will auto-resolve merge conflict
- Check in code not related to a task.

TFS Version control



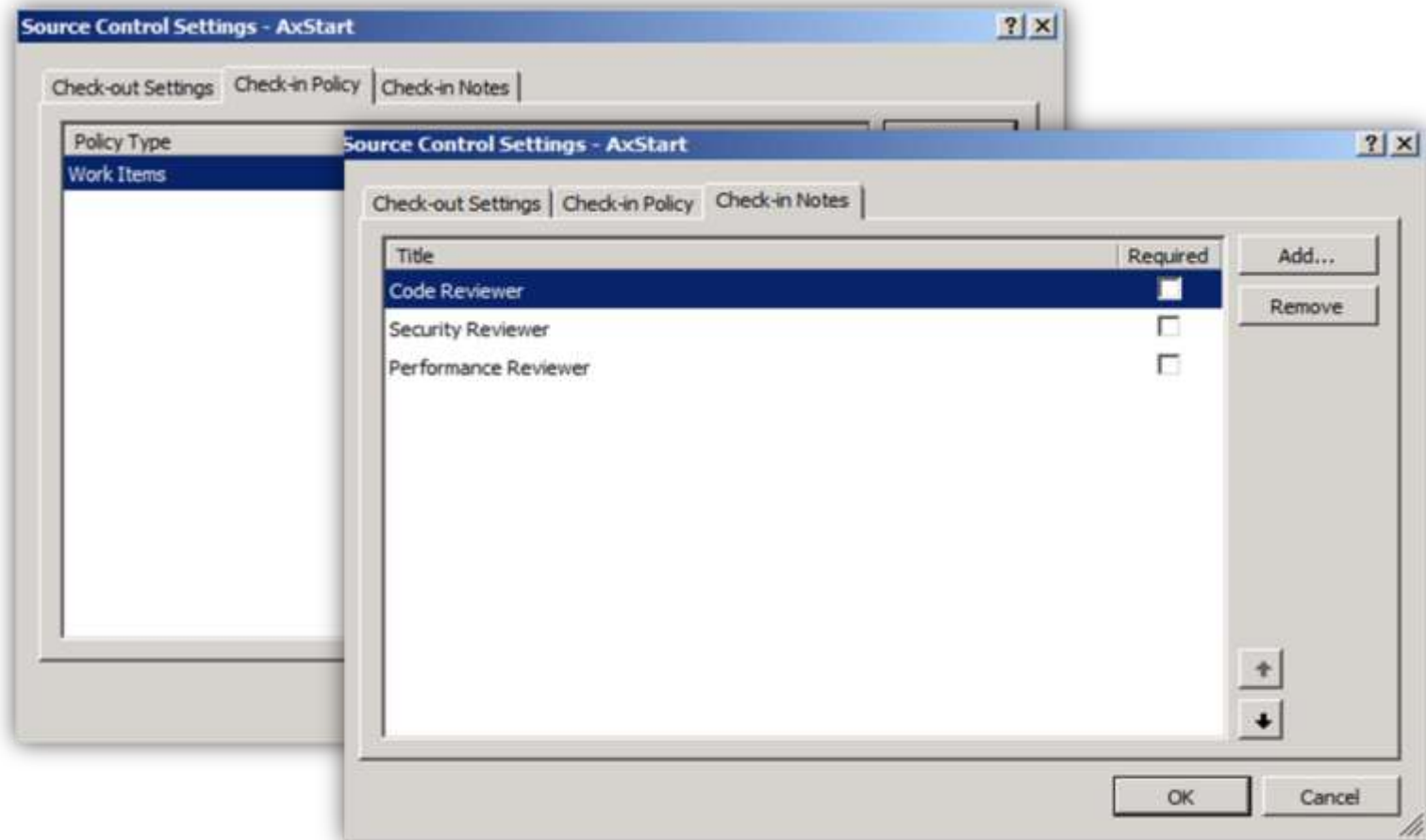
Exclusive Lock

TFS Version control



Exclusive Lock

TFS Version control



Missing integration

TFS Version control

Shared AOS:

Change all the code that use:

`SysVersionControlParameters.RepositoryFolder`

```
static str TFSPath(str _RepositoryFolder)
```

```
{
```

```
    UserInfo userinfo;
```

```
    select networkAlias from userinfo where userinfo.id == curUserId();
```

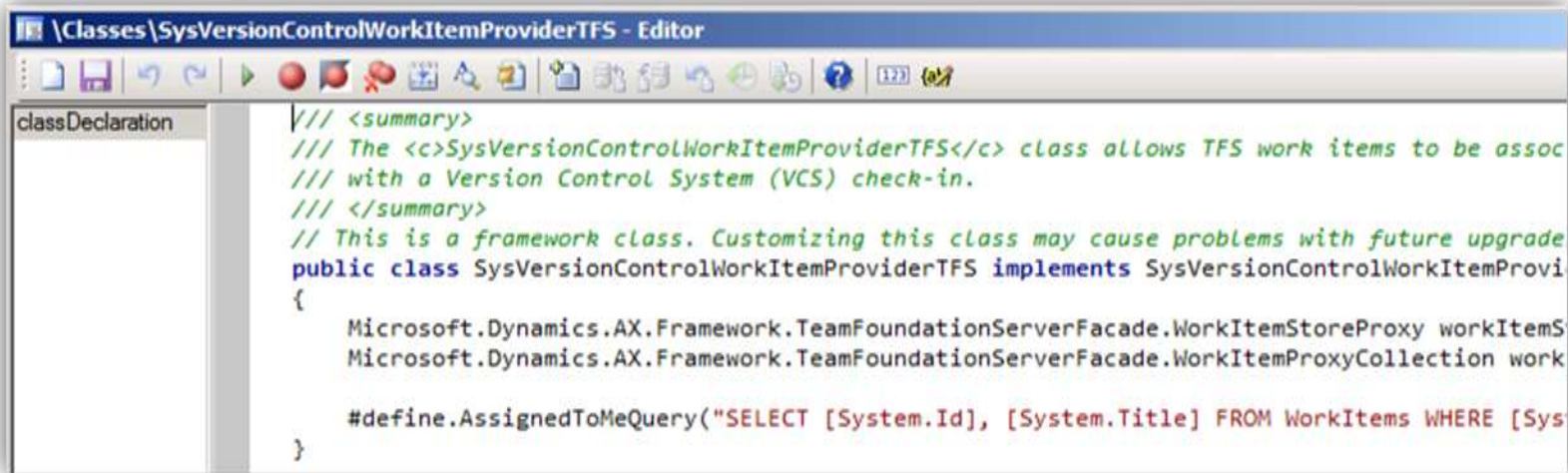
```
    return strFmt(_RepositoryFolder,userinfo.networkAlias);
```

```
}
```



TFS Version control

Change the Query to get what you want. NOT: Add code to a user story, Add code to test-case, A closed task.

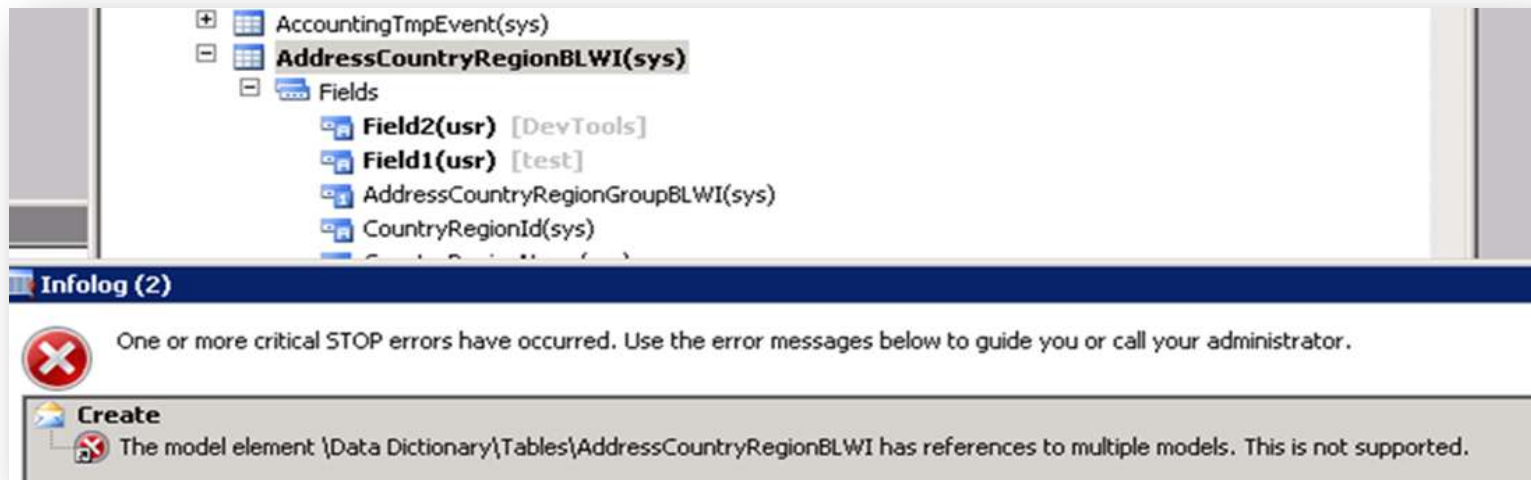


```
\\Classes\SysVersionControlWorkItemProviderTFS - Editor
// <summary>
// The <c>SysVersionControlWorkItemProviderTFS</c> class allows TFS work items to be assoc
// with a Version Control System (VCS) check-in.
// </summary>
// This is a framework class. Customizing this class may cause problems with future upgrade
public class SysVersionControlWorkItemProviderTFS implements SysVersionControlWorkItemProvi
{
    Microsoft.Dynamics.AX.Framework.TeamFoundationServerFacade.WorkItemStoreProxy workItemS
    Microsoft.Dynamics.AX.Framework.TeamFoundationServerFacade.WorkItemProxyCollection work

    #define.AssignedToMeQuery("SELECT [System.Id], [System.Title] FROM WorkItems WHERE [Sys
}
}
```

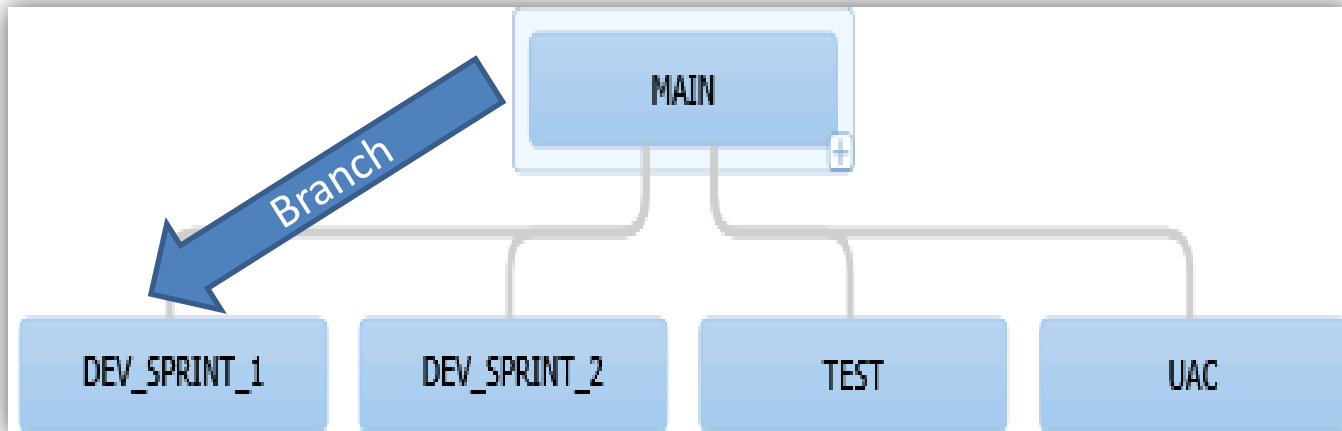
TFS Version control

When using models; Know what you do and why.



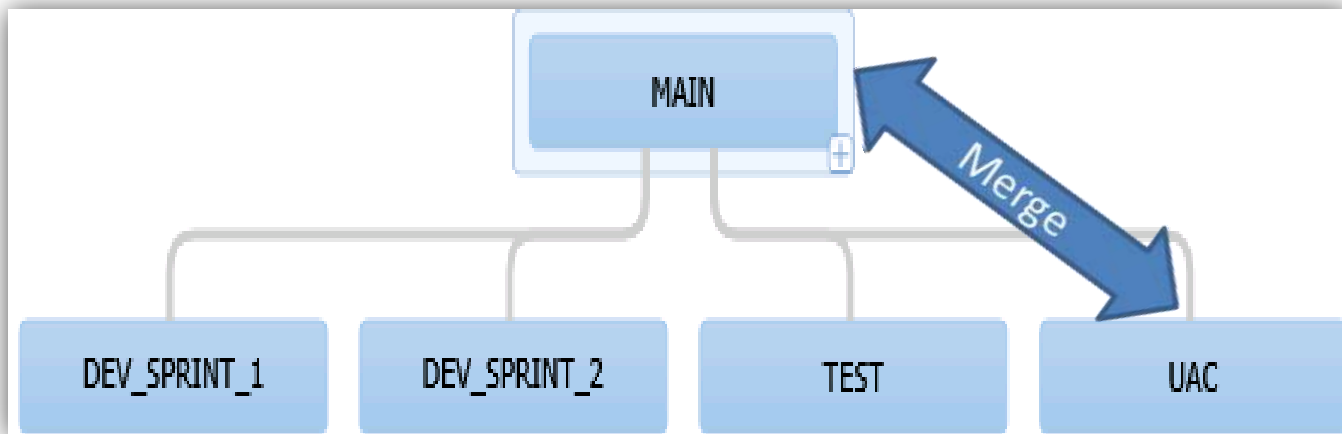
Release Management

Branching: Make a spin-off.



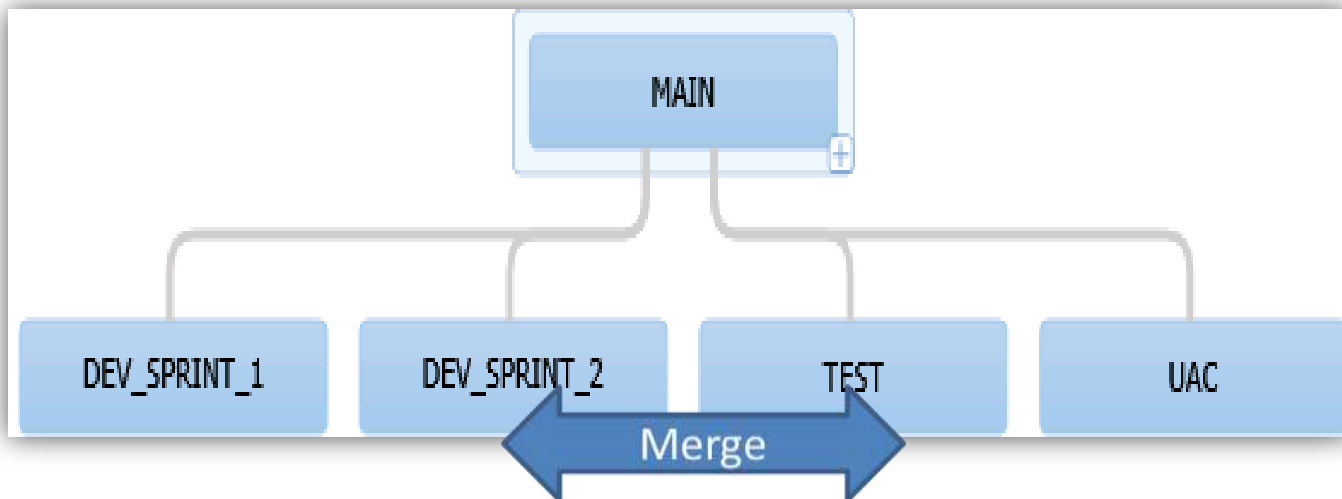
Release Management

Merging: Synchronization of spin-off with other spin-off of the same Parent node.

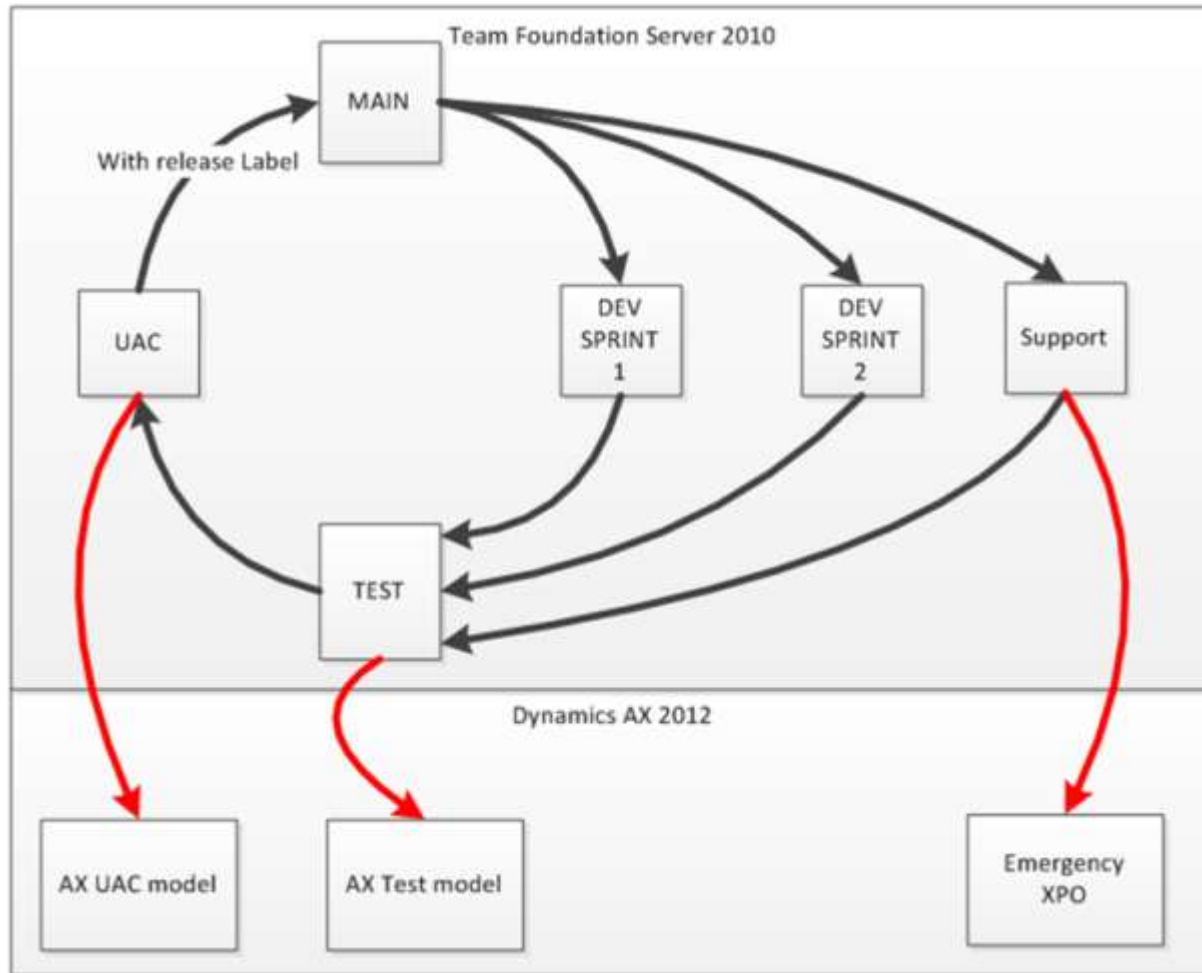


Release Management

Baseless Merging: Bi-directional synchronization of spin-off with a same Parent .



Release Management



Release Management

- **Merge Paranoia**—avoiding merging at all cost, usually because of a fear of the consequences.
- **Merge Mania** —spending too much time merging software assets instead of developing them.
- **Big Bang Merge** —deferring branch merging to the end of the development effort and attempting to merge all branches simultaneously.
- **Never-Ending Merge** —continuous merging activity because there is always more to merge.
- **Wrong-Way Merge** —merging a software asset version with an earlier version.
- **Branch Mania** —creating many branches for no apparent reason.
- **Cascading Branches** —branching but never merging back to the main line.
- **Mysterious Branches** —branching for no apparent reason.
- **Temporary Branches** —branching for changing reasons, so the branch becomes a permanent temporary workspace.
- **Volatile Branches** —branching with unstable software assets shared by other branches or merged into another branch.
- **Development Freeze** —stopping all development activities while branching, merging, and building new base lines.
- **Berlin Wall** —using branches to divide the development team members, instead of dividing the work they are performing.

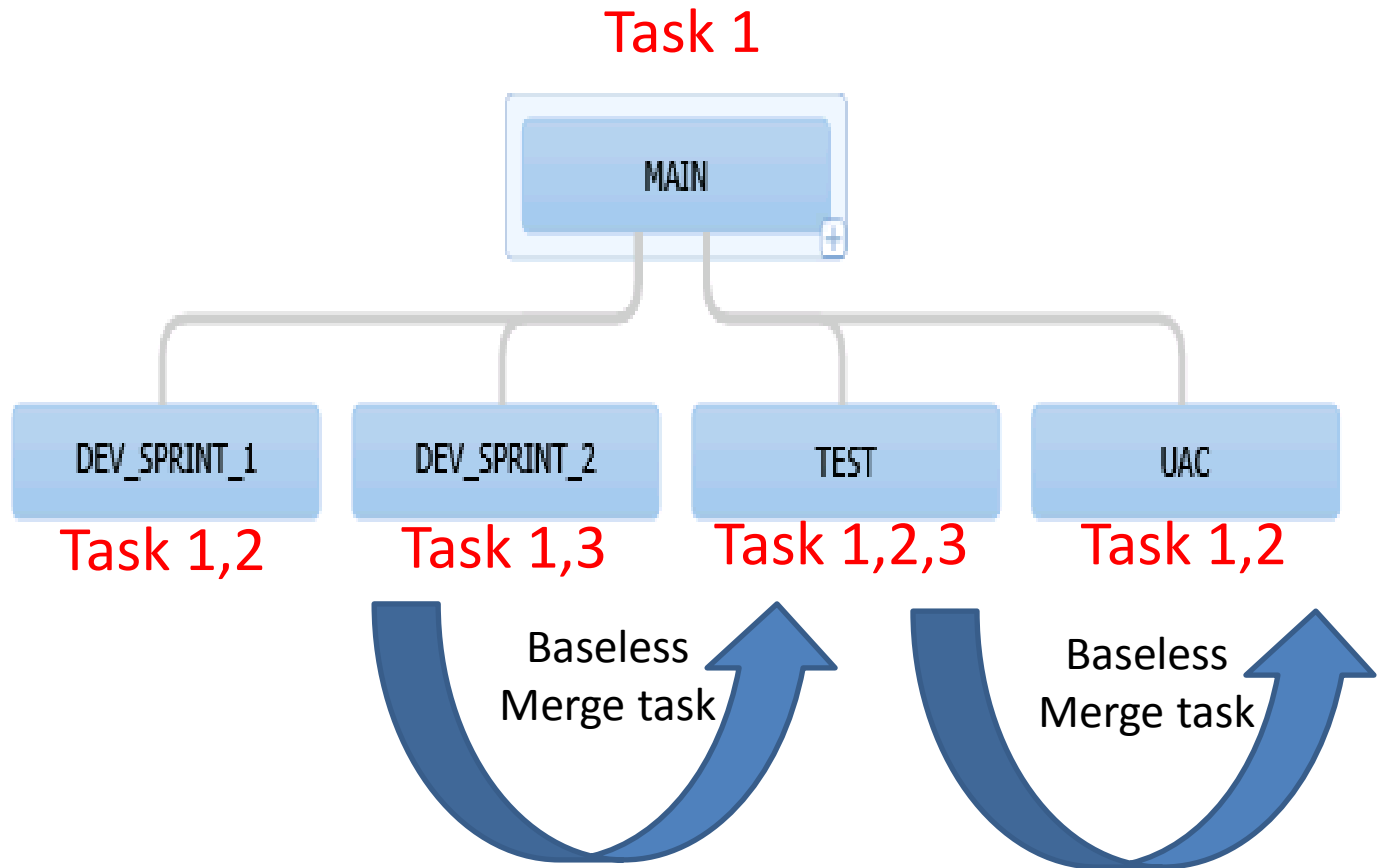
Tracking of Issues + Code Changes

Is that code already released to Test?

It works in Test but not in UAT, is there a difference in code (no more layer compare)

What do I need to Test (now Test is related to user story, and my tasks also)

Tracking of Issues + Code Changes



```
tf merge /baseless "c:\TFS_Merge\AX2012\DEV\DEV_SPRINT_1" "c:\TFS_Merge\AX2012\TEST\TEST" /recursive
```

Tracking of Issues + Code Changes

Details for Work Item 1

Title: Task 1

Status:

Assigned To: Administrator

State: Active

Reason: New

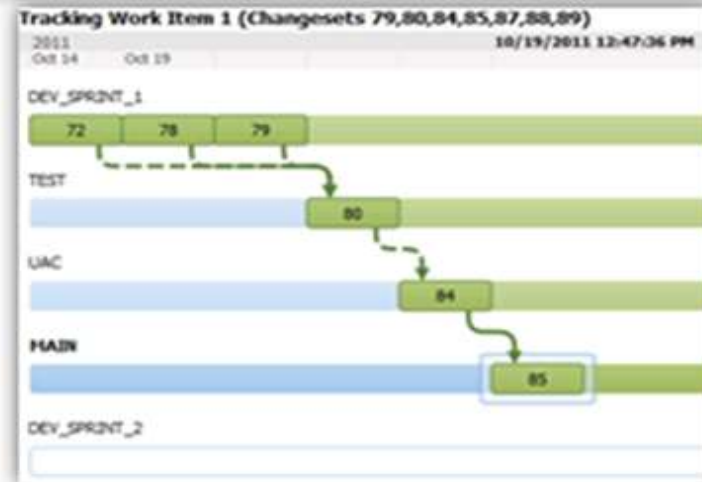
Planning

Stack Rank: Priority: 2

Details | Implementation | All Links | Attachments

Link to:

ID	Work Item...	Title
Changeset (7 items)		
		Changeset 80
		Changeset 85
		Changeset 87
		Changeset 88
		Changeset 79
		Changeset 89
		Changeset 84



Tracking of Issues + Code Changes

Task 2: Task 2

Title: Task 2

Status:

Assigned To: Administrator

State: Active

Reason: New

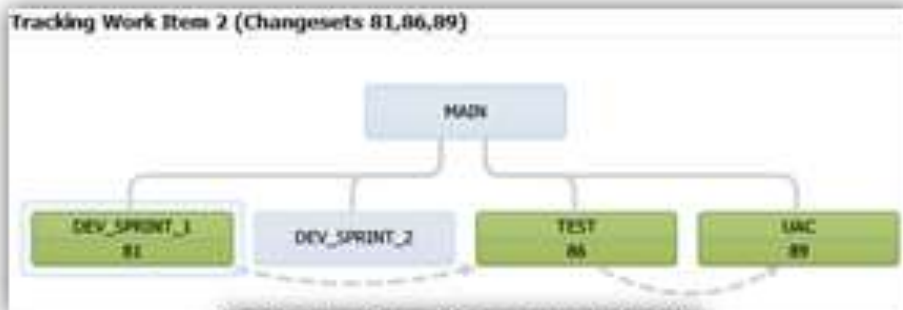
Planning:

Stack Rank: Prior

Details | Implementation | All Links | Attachments

New | Go Link to

ID	Work Item...	Title
Changeset (3 items)		
	Changeset 81	
	Changeset 86	
	Changeset 89	



Tracking of Issues + Code Changes

Task 3 : Task 3

Title: Task 3

Status:

Assigned To: Administrator

State: Active

Reason: New

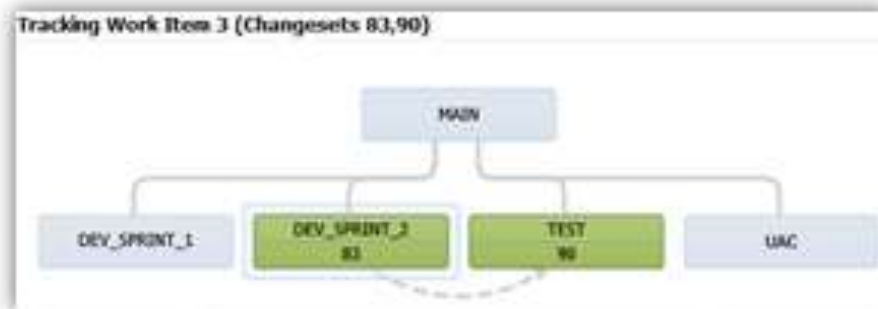
Planning

Stack Rank: Priority:

Details | Implementation | All Links | Attachments

new Link to

ID	Work Item...	Title
Changeset (2 items)		
		Changeset 90
		Changeset 83



Tracking Work Item 3 (Changesets 83,90)

10/19/2011 12:46:11 PM
Oct 19

DEV_SPRINT_2

83

TEST

90

DEV_SPRINT_1

MAIN

UAC

Tracking of Issues + Code Changes

- Release Documentation,
- Reports?
 - Iteration/Sprints,
 - Tasks,
 - Branch,
 - Changed object.

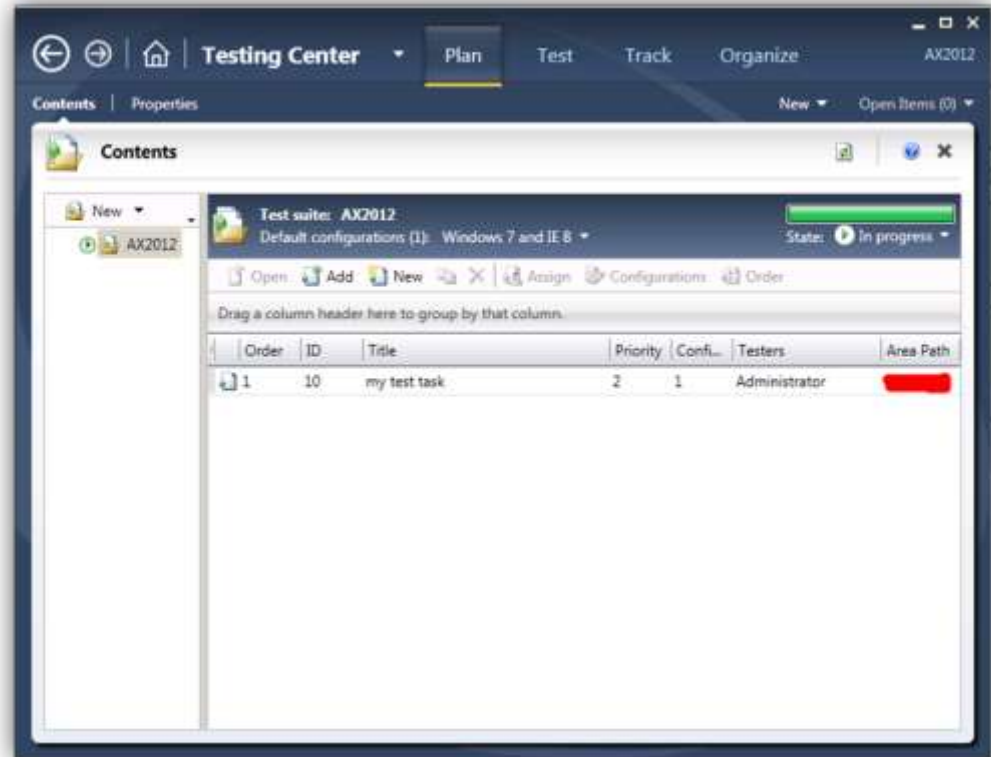
Building a Release

- No content.

Automated Testing

Microsoft Test Manager.

A bug need to be repaired in the code and in the test case.



Automated Testing

Test requirements, not tasks.

The screenshot displays a software development tool interface. The main window is titled "User Story 5 : custform". It contains several sections:

- Title:** custform
- Status:** Assigned To: Administrator, State: Active, Reason: New
- Classification:** Area: , Iteration:
- Planning:** Stack Rank:
- Navigation:** Details | Implementation | **Test Cases** | All Links | Attachments
- Parent and Child Stories and Tasks:** A table with columns ID, Work Item..., and Title. It shows three child items: ID 6 (Task: change tales), ID 7 (Task: change form), and ID 8 (Task: create autorisation).
- Test Cases testing this Story:** A sub-window with a navigation bar (New, Link to, etc.) and a table with columns ID, Work Item..., and Title. It shows one tested item: ID 12 (Test Case: Test Form).

Automated Testing

The screenshot displays the Microsoft Test Center interface. The top navigation bar includes 'Testing Center' with sub-tabs for 'Plan', 'Test', 'Track', and 'Organize'. Below this, there are links for 'Test Plan Manager', 'Test Configuration Manager', and 'Test Case Manager'. The main area is divided into two panels. The left panel, titled 'Shared Steps Manager', contains a table with the following data:

ID	Priority	Title
11	2	shared Step start Ax client

The right panel shows a detailed view of a step, with tabs for 'Steps', 'Summary', 'Tested User Stories', 'All Links', and 'Attachments'. The 'Steps' tab is active, showing a list of steps:

1. shared Step start Ax client
2. Open Cust Form
3. Edit test
4. Change name CustomerName @Promentum
5. Change name back CustomerName @Crimsonwing

Below the list, there is a 'Parameter Values:' section with a table:

Parameter	Value
Promentum	Crimsonwing
Pro	CWP

Automated Testing

The image displays two overlapping windows from Microsoft Dynamics AX. The left window is titled "Test 1 of 1: Iteration 1 of 1" and shows a test task named "10*: my test task" running on "Windows 7 and IE 8". The test steps are:

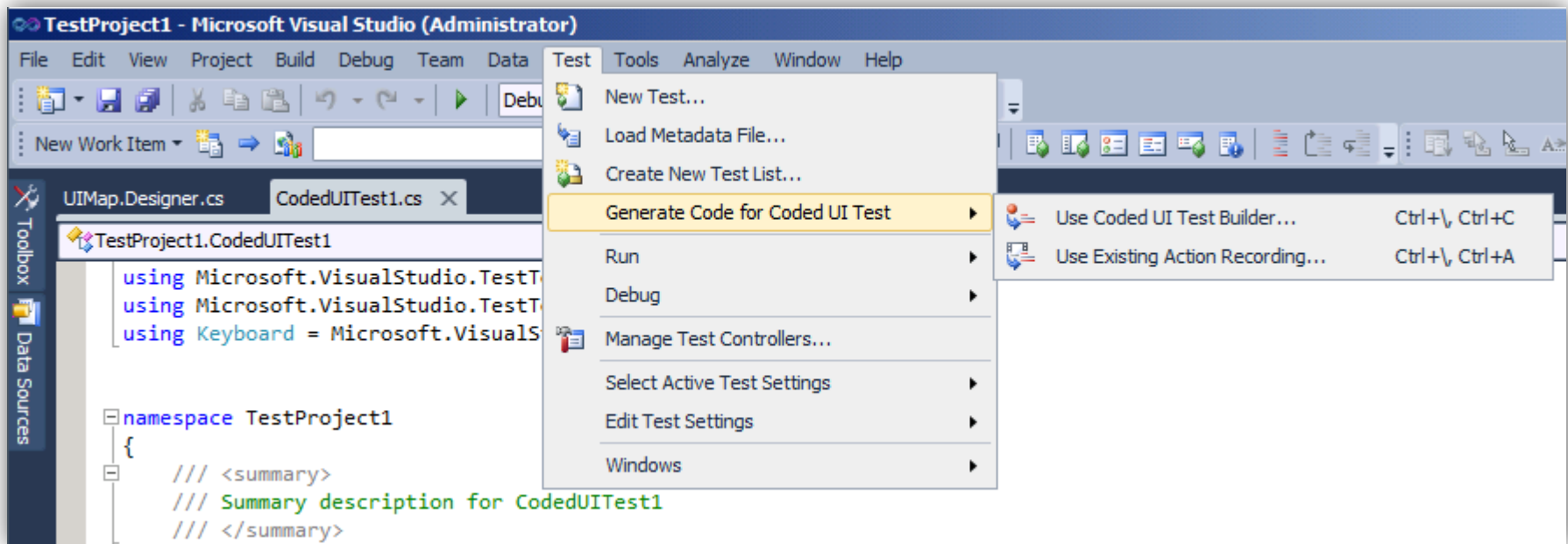
1. shared Step start Ax client (Status: Success)
2. Open Cust Form (Status: Success)
3. Edit test (Status: Success)
4. Change name CustomerName @Promentum (Status: Success)
 Promentum = Pro
5. Change name back CustomerName @Crimsonwing (Status: Success)

The right window shows the "Customer" form for account "1101 : Forest Wholesales". The form fields are:

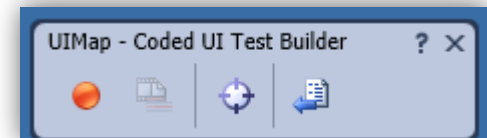
Field	Value
Account:	1101
Record type:	Organization
Name:	Forest Wholesales
Search name:	CWP
Customer group:	10
Classification group:	

Automated Testing

Visual studio Premium or Ultimate



Control will appear for recording or creating checks



Automated Testing

Check if field is Read-only.

The screenshot displays a software application interface with a 'General' tab. The 'Name' field is highlighted with a blue border. An overlaid 'Coded UI Test Builder' window shows a table of properties for the selected field, with the 'ReadOnly' property set to 'True'.

Property	Value
InsertionIndexLineRelative	0
SelectionStart	0
SelectionEnd	0
SelectionText	
IsPassword	False
ReadOnly	True
CurrentLine	0
Text	Banana Conference Ce
CopyPastedText	
Generic	
ClassName	AxEdit
FriendlyName	Name
HasFocus	False

Automated Testing

Check if field is Read-only.

```
public void CodedUITestMethod1()
{
    // To generate code for this test, select "Generate Code for Coded UI Test" from the shortcut menu and se
    // For more information on generated code, see http://go.microsoft.com/fwlink/?LinkId=179463
    this.UIMap.sharedStepstartAxclient();
    this.UIMap.OpenCustForm();
    this.UIMap.Edittest();
    this.UIMap.ChangenameCustom
    this.UIMap.ChangenameCustom {
    this.UIMap.ChangenamebackCu
    this.UIMap.ChangenamebackCu
    this.UIMap.NameReadOnly();
}

public void NameReadOnly()
{
    Variable Declarations

    // Verify that 'Name' text box's property 'ReadOnly' equals 'True'
    Assert.AreEqual(this.NameReadOnlyExpectedValues.UINameEditReadOnly, uINameEdit.ReadOnly);
}

[GeneratedCode("Coded UI Test Builder", "10.0.40219.1")]
public class NameReadOnlyExpectedValues
{
    #region Fields
    /// <summary>
    /// Verify that 'Name' text box's property 'ReadOnly' equals 'True'
    /// </summary>
    public bool UINameEditReadOnly = true;
    #endregion
}
```

Whitepapers

TFS

<http://branchingguidance.codeplex.com/>
<http://tfsbranchingguideii.codeplex.com/>

Dynamics AX

- Testing best practices for Microsoft Dynamics AX 2012.
- Developing Solutions for Microsoft Dynamics AX in a Shared AOS Development Environment AX2012.
- Deploying Customizations Across Microsoft Dynamics AX 2012 Environments .