

Microsoft Dynamics® AX

Integrating a third party Version Control System into Microsoft Dynamics® AX 2009

White Paper

This white paper describes a case study of how to add support for a third party version control system to Microsoft Dynamics® AX 2009 by using Team Foundation Server.

Date: February 2, 2010



Table of Contents

Introduction	3
Solution	4
Class Diagram of VCS in Microsoft Dynamics AX 2009	4
VCS Integration	4
Framework Interop Extension.....	4
Configuration	6
Additional Information	8
SystemVersionControlFilebased and SystemVersionControlFilebasedBackEnd Framework.....	8

Introduction

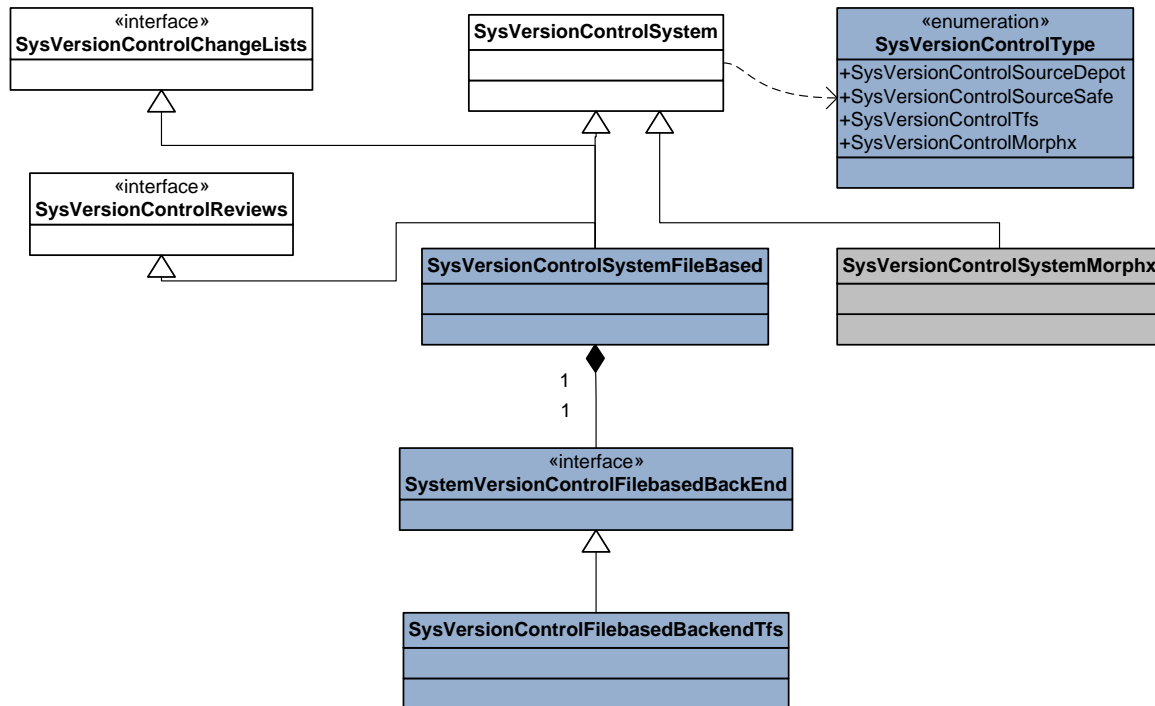
The MorphX® Integrated Development Environment (IDE) can integrate various Version Control Systems (VCS). Team Foundation Server (TFS) is a newly integrated VCS with the MorphX IDE in Microsoft Dynamics® AX 2009. This whitepaper provides detailed step-by-step instructions to integrate a VCS with the MorphX IDE, by using how TFS was included in Microsoft Dynamics AX 2009 as an example.

The Version Control Extensibility Framework was refactored in Microsoft Dynamics AX 2009, which is now used by the integration with TFS. The refactoring work was focused primarily on supporting file-based VCS.

Solution

Class Diagram of VCS in Microsoft Dynamics AX 2009

The following class diagram shows the new structure and relationships of VCS classes in Microsoft Dynamics AX 2009.



VCS Integration

The integration process is divided into two parts:

- Framework Interop Extension
- Configuration

Framework Interop Extension

To integrate TFS as a VCS, the **SysVersionControlFilebasedBackEndTfs** class was added. Follow these steps to implement a similar class:

1. Add a new class that represents the VCS and implements the **SysVersionControlFilebasedBackEnd** interface.

```
class SysVersionControlFilebasedBackEndTfs
implements SysVersionControlFilebasedBackEnd
{
}
```

2. Add a new base enumeration value to the **SysVersionControlControlType** base enumeration type. For TFS, the **SysVersionControlTfs** base enumeration value was added.

3. Add new class construction logic to the newType method of the SysVersionControlSystem class.

```
public static SysVersionControlSystem newType(SysVersionControlType _type)
{
    switch (_type)
    {
        case SysVersionControlType::SysVersionControlTfs:
            return
SysVersionControlSystemFileBased::newBackEnd(SysVersionControlFilebasedBackEndTfs::construct());
        }
    return null;
}
```

4. Implement the VCS setup methods. These methods are the bootstrap methods that are responsible for initializing the backend instance. Therefore, they must be implemented for each VCS.

```
public static SysVersionControlFilebasedBackEndTfs construct()
public SysVersionControlSystemStatus init(SysVersionControlParameters _parameters)
public void interpretXMLPluginParameters(XmlElement _topNode)
```

5. Implement the basic VCS operation methods in the new class such as the fileGetVersion, fileSetCheckIn, fileCheckout, fileAddToVersionControl, and fileDelete methods. Full documentation is available in the public API at <http://go.microsoft.com/fwlink/?LinkId=180373>.

```
public void createRepositoryEnd(SysVersionControlItemComment _comment)
public boolean fileAddToVersionControl(Filename _filename)
public str fileCheckedOutTo(Filename _filename)
public boolean fileCheckout(Filename _filename)
public boolean fileDelete(Filename _filename)
public boolean fileExists(Filename _filename)
public Filename fileGetVersion(Filename _filename, SysVersionControlItemVersion _version)
public boolean fileLock(Filename _filename)
public boolean fileRename(Filename _oldFilename, Filename _newFilename)
public boolean fileSetCheckin(Map _setOfFiles, SysVersionControlItemComment _comment)
public boolean fileSetResolve(Map _setOfFiles)
public Map fileSynchronize(
    Filename _filename,
    boolean _force,
    boolean _deleteLocalObjects,
    boolean _silentOnSuccess = false,
    SysVersionControlItemVersion _version = 0)
public container fileUndoCheckout(Filename _filename)
public boolean fileUnlock(Filename _filename)
public Map folderSetSynchronize(
    Set _folderSet,
    boolean _force,
    boolean _deleteLocalObjects)
```

6. Use the advanced VCS methods to implement additional VCS features. The framework also defines a list of runtime surrogate methods to let the VCS notify the framework whether these features

are supported by the VCS. These features are optional. Advanced features are defined in the `SysVersionControlChangeLists` and `SysVersionControlReviews` interfaces. TFS implements the `supportMultipleCheckOut` and `supportSubmittedChangeLists` methods.

```
public SysVersionControlTmpItem changeListContents(SysVersionControlChangeNumber _number)
public SysVersionControlPendingChangeList changeListsPending()
public FilenameFilter reviewFilenameLookupFilter()
public void reviewFileSet(
    Set _setOfFiles,
    SysVersionControlItemComment _comment,
    FilenameSave _fileNameOfReviewPackage,
    SysVersionControlRepositoryFolder _folder)
public boolean supportChangeVersion()
public boolean supportCheckInWizard()
public boolean supportMultipleCheckout()
public boolean supportPendingChangeLists()
public boolean supportReviews()
public boolean supportSubmittedChangeLists()
```

7. Implement the plumbing methods that are required by the framework.

```
public str actionText(SysVersionControlSystemFileAction _action)
public str actionTextForCheckedOutFile(Filename _filename)
public ClassDescription description()
public void showInfo()
```

As an example of how to implement these steps see the implementation of the `SysVersionControlFilebasedBackendTfs` class.

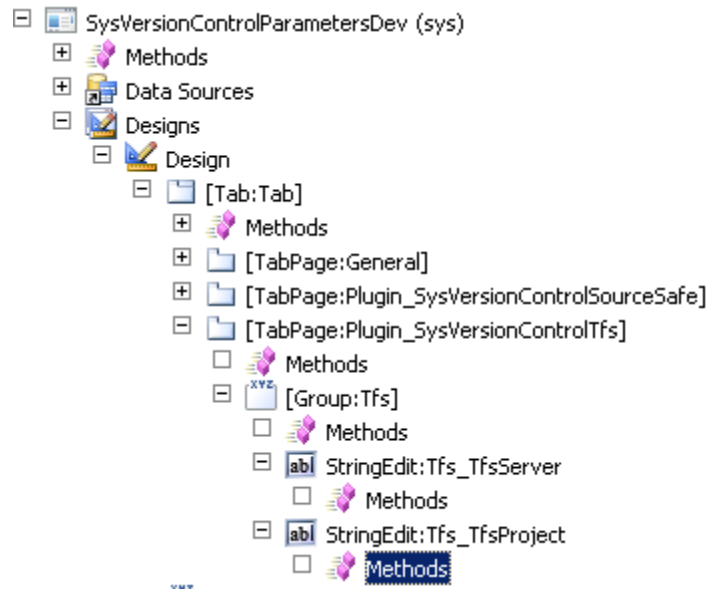
Configuration

The configuration collects the setup data that is required to connect to the VCS from the VCS setup form. This is the `SysVersionControlParametersDev` form. Each additional VCS can have a plug-in tab page on this form. The `SysVersionControlParameters` table should be extended to persist the backend-specific setup data.

For TFS, two new string fields, each with an extended data type, were added: the `TfsServer` and `TfsProject` fields. The `Tfs` field group was added to compose these two fields. The `Tfs` field group was bound to a new tab page with the name `Plugin_SysVersionControlTfs`. The tab page should be named `Plugin_` followed by the name of the enumeration value added in step 2 of the Framework Interop Extension section. This is because this name is used by the VCS setup form to activate the tab page.

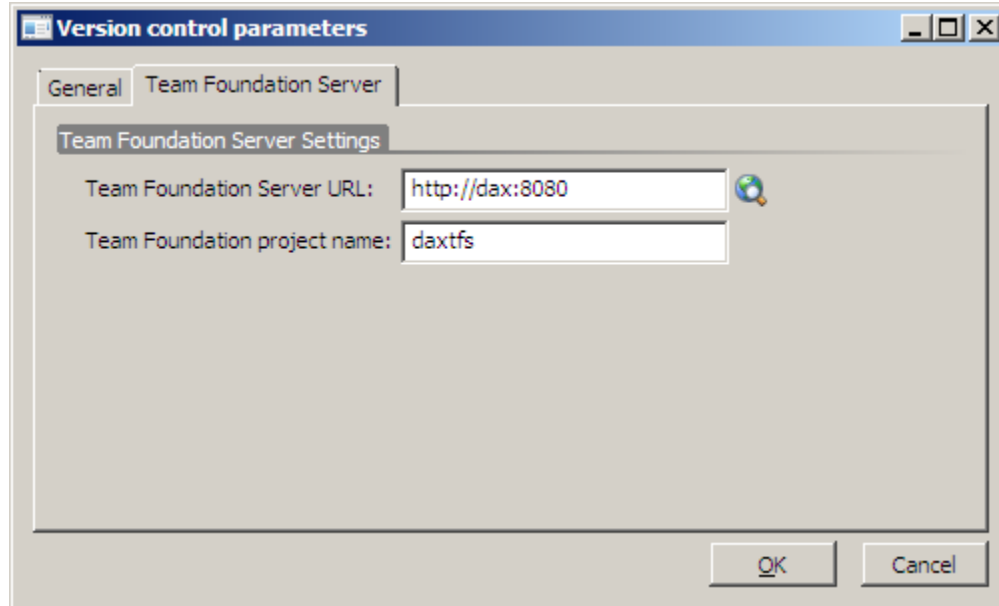
Layout of the SysVersionControlParametersDev Form

The following diagram describes the layout of the VCS setup form. It also shows the new fields that were added for TFS integration.



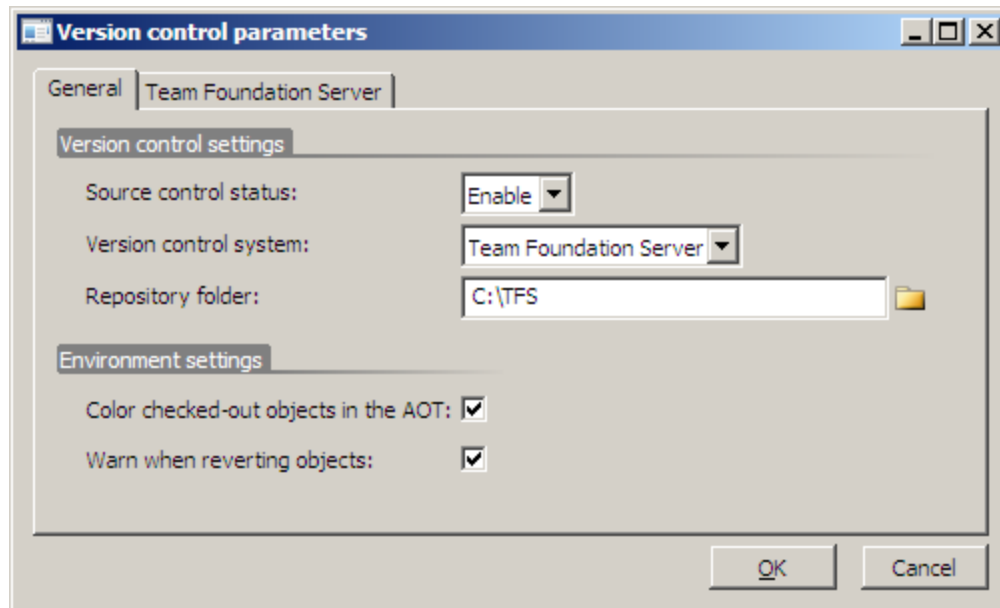
The Team Foundation Server Tab Page

The following image shows the tab page specific to TFS from the VCS setup form. This tab becomes visible when Team Foundation Server is selected as the Version control system on the General tab.



The General Tab Page

The following image shows the general tab of the VCS setup form. The items listed in the Version control system combo box are from the SysVersionControlControlType base enumeration values.



Setting up TFS required the repository folder field from the General tab and the Team Foundation Server URL and Team Foundation project name fields from the Team Foundation Server tab. This information is used in the init method of the SysVersionControlFilebasedBackendTfs class to connect to a Team Foundation Server.

```
public SysVersionControlSystemStatus init(SysVersionControlParameters _parameters)
{
    SysVersionControlTfsServer tfsServer = _parameters.TfsServer;
    SysVersionControlTfsProject tfsProject = _parameters.TfsProject;
    repositoryFolder = _parameters.RepositoryFolder;
}
```

The init method will be called when Microsoft Dynamics AX 2009 starts and when the user clicks the OK button on the setup form. See the init method of the VersionControl class for a detailed workflow.

Additional Information

SystemVersionControlFilebased and SystemVersionControlFilebasedBackend Framework

This subsolution provides a file-based integration framework, which should be used by any integration work of a third party file-based VCS.

With this framework we see the following integration:

1. A distinction in terms of artifacts abstraction between files and AOT elements. Any conversion from an AOT tree node to an XPO file is delivered by the framework and does not require any further integration.
2. The SysVersionControlFilebasedBackend interface only applies to files. Therefore, it more appropriately defines an API to be used by the third party VCS.

The interface can be logically grouped into four categories. The first three categories are required but the last is optional for third party VCS integration. The interface is fully documented at <http://go.microsoft.com/fwlink/?LinkId=180373>.

Setup Logic

```
public static SysVersionControlFilebasedBackEndTfs construct()  
public SysVersionControlSystemStatus init(SysVersionControlParameters _parameters)  
public void interpretXMLPluginParameters(XmlElement _topNode)
```

Basic VCS Operations

```
public void createRepositoryEnd(SysVersionControlItemComment _comment)  
public boolean fileAddToVersionControl(Filename _filename)  
public str fileCheckedOutTo(Filename _filename)  
public boolean fileCheckout(Filename _filename)  
public boolean fileDelete(Filename _filename)  
public boolean fileExists(Filename _filename)  
public Filename fileGetVersion(Filename _filename, SysVersionControlItemVersion _version)  
public boolean fileLock(Filename _filename)  
public boolean fileRename(Filename _oldFilename, Filename _newFilename)  
public boolean fileSetCheckin(Map _setOfFiles, SysVersionControlItemComment _comment)  
public boolean fileSetResolve(Map _setOfFiles)  
public Map fileSynchronize(  
    Filename _filename,  
    boolean _force,  
    boolean _deleteLocalObjects,  
    boolean _silentOnSuccess = false,  
    SysVersionControlItemVersion _version = 0)  
public container fileUndoCheckout(Filename _filename)  
public boolean fileUnlock(Filename _filename)  
public Map folderSetSynchronize(  
    Set _folderSet,  
    boolean _force,  
    boolean _deleteLocalObjects)
```

Informational

```
public ClassDescription description()  
public void showInfo()  
public str actionText(SysVersionControlSystemFileAction _action)  
public str actionTextForCheckedOutFile(Filename _filename)  
public SysVersionControlTmpItem fileHistory(Filename _filename)  
public SysVersionControlTmpChange folderChanges(FilePath _folder)  
public SysVersionControlTmpItem filesPending(  
    SysVersionControlChangeListId _changeListId,  
    SysVersionControlRepositoryFolder _folder,  
    boolean _silent = false)
```

Advanced VCS Features (optional)

```
public SysVersionControlTmpItem changeListContents(SysVersionControlChangeNumber _number)  
public SysVersionControlPendingChangeList changeListsPending()
```

```
public FilenameFilter reviewFilenameLookupFilter()
public void reviewFileSet(
Set _setOfFiles,
SysVersionControlItemComment _comment,
FilenameSave _fileNameOfReviewPackage,
    SysVersionControlRepositoryFolder _folder)
public boolean supportChangeVersion()
public boolean supportCheckInWizard()
public boolean supportMultipleCheckout()
public boolean supportPendingChangeLists()
public boolean supportReviews()
public boolean supportSubmittedChangeLists()
```

Microsoft Dynamics is a line of integrated, adaptable business management solutions that enables you and your people to make business decisions with greater confidence. Microsoft Dynamics works like and with familiar Microsoft software, automating and streamlining financial, customer relationship and supply chain processes in a way that helps you drive business success.

U.S. and Canada Toll Free 1-888-477-7989
Worldwide +1-701-281-6500
www.microsoft.com/dynamics

This document is provided "as-is." Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it.

Some examples depicted herein are provided for illustration only and are fictitious. No real association or connection is intended or should be inferred.

This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes. You may modify this document for your internal, reference purposes.

© 2010 Microsoft Corporation. All rights reserved.

Microsoft