

Microsoft Dynamics® AX 2012

Developing Solutions for Microsoft Dynamics AX in a Shared AOS Development Environment

White Paper

This document provides guidance for developing solutions when multiple development clients are connected to a single shared AOS.

Date: July 2011

Author: Robert Badawy

Send suggestions and comments about this document to adocs@microsoft.com. Please include the title with your feedback.



Table of Contents

Introduction	3
General Guidelines	4
Refresh metadata in development scenarios by restarting the development client.....	5
Refresh metadata in system administration scenarios by restarting the AOS	5
Compile elements on one developer client at a time	5
Synchronize the database on one developer client at a time	5
Generate CIL on one developer client at a time.....	5
Deploy elements on one developer client at a time	5
Use MorphX VCS.....	5
Import XPOs without restarting the AOS	6
Do not modify elements in the layer below the highest customized layer	6
Consider Visual Studio to be a separate development client	6
Use a separate AOS to debug IL code.....	6
Do not add an element to version control in a layer above the layer you are in.....	6
Do not expect version control history to show the history of client operations.....	6
TFS Guidelines	6
Enable TFS version control	7
Edit the TFS workspace	7
Open the workspace edit utility by using the tf.exe tool	7
Open the workspace edit utility by using Team Explorer	7
Known Issues	8
Compiling an element creates a new version of the element.....	8
Changes to temporary tables require you to clear the metadata cache in the report server	8
Hotswapping required on CIL runtime so that new sessions pick up the latest changes	8
Data is lost when inserting data into a new table column	8
Only one developer at a time can deploy static reports with assemblies.....	8

Introduction

Developing for Microsoft Dynamics® AX is best done in an environment where each developer has a full Microsoft Dynamics AX installation that runs on a single box. The developer artifacts are synchronized by using one of the supported version control systems in Microsoft Dynamics AX, and solutions are produced by running a build of the sources in the version control system (VCS).

Figure 1 shows an example of a single Application Object Server (AOS) topology with Microsoft Visual Studio® Team Foundation Server (TFS) used as VCS backend.

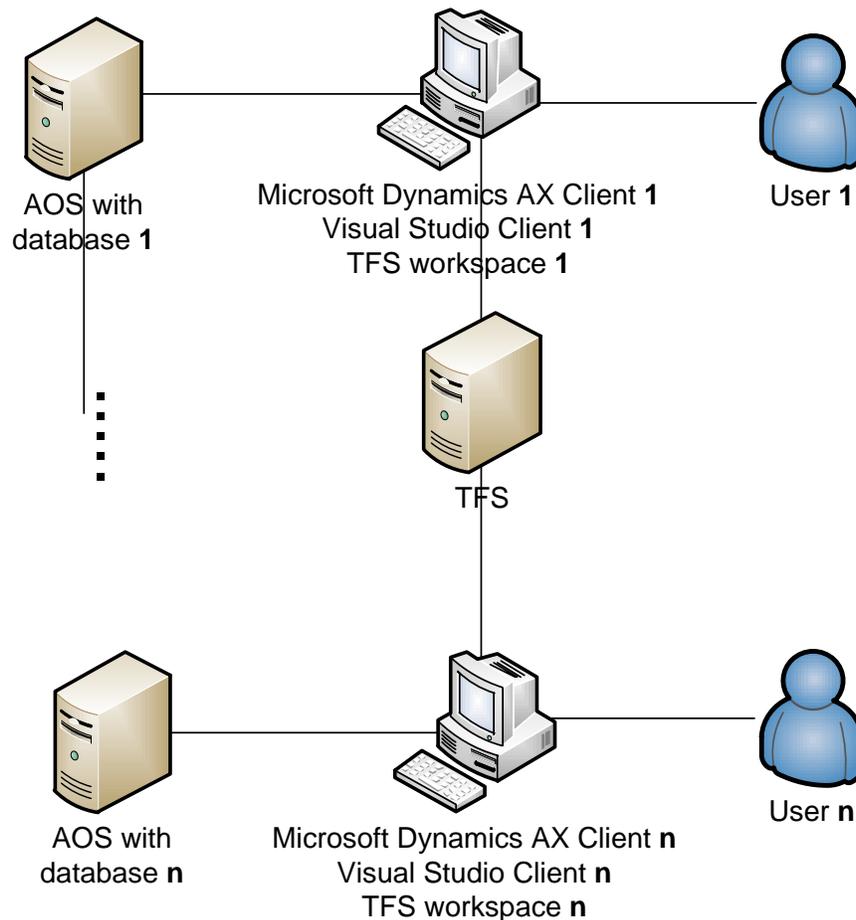


Figure 1: Single AOS development topology

If you do not follow the recommended single box development topology, you may encounter unpredictable behavior. To minimize issues, follow the guidelines in this document.

Figure 2 shows an example of a shared AOS development topology that uses TFS as a version control backend and a TFS public workspace as a version control frontend.

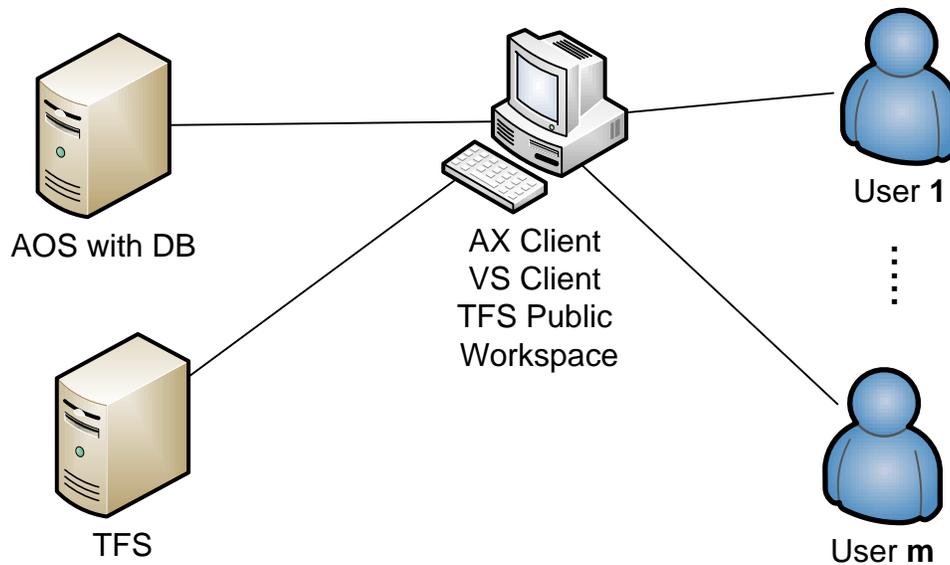


Figure 2: Shared AOS development topology

General Guidelines

The following guidelines will help you to have the most predictable experience if you develop in a shared AOS environment. These guidelines will help you to:

- Ensure that metadata is up to date in all development clients.
- Have predictable behavior throughout the development cycle.
- Have clarity on how to install new solutions and solution updates.
- Debug element types without disrupting other developers.
- Maximize the uptime of the shared AOS.

If you do not follow the guidelines, you may encounter:

- Stale metadata on developer clients.
- Unpredictable behavior of development clients that connect to the shared AOS.
- Trouble installing new solutions and upgrading existing solutions, and in some cases corrupt business data.
- An unresponsive AOS while debugging the server.
- The need to frequently restart the AOS, disrupting the development team.

Refresh metadata in development scenarios by restarting the development client

You do not need to restart the AOS to get the latest metadata on development clients. Select **Refresh** or **Restore** in the developer client to get the latest metadata in the AOT. If this does not work, restart the development client.

Refresh metadata in system administration scenarios by restarting the AOS

Administration scenarios require restarting the AOS and recompiling the AOT. For example, restart the AOS and recompile the AOT when you:

- Install a new model or upgrade an existing model.
- Install updates.
- Upgrade from one version to another.

Compile elements on one developer client at a time

If you compile the same element from different development clients, the first developer to compile the element will succeed, and the other developers will get an error.

Synchronize the database on one developer client at a time

If you synchronize the same element from different development clients simultaneously, the first developer to synchronize the element will succeed, and the other developers will get an error.

Generate CIL on one developer client at a time

If you perform Common Intermediate Language (CIL) generation for the same element from different development clients simultaneously, the first developer to perform the CIL generation will succeed, and the other developers will get an error.

Deploy elements on one developer client at a time

If you deploy the same element to Enterprise Portal or the report server from different development clients simultaneously, the first developer to deploy the element will succeed, and the other developers will get an error.

Use MorphX VCS

MorphX VCS is a version control system that comes with Microsoft Dynamics AX and allows you to check in and check out elements. MorphX VCS allows developers to share an AOS. It enables you to lock elements and to avoid write conflicts when multiple developers attempt to modify the same element.

To enable MorphX VCS:

1. In the developer workspace in Microsoft Dynamics AX, click **Version control > Version Control Parameters**.
2. In **Version control status** field, select **Enable**.
3. In the **Version control system** field, select **MorphX VCS**, and then click **OK**.

For more information about setting up version control, see the [Microsoft Dynamics AX version control documentation](#) on MSDN.

Import XPOs without restarting the AOS

When you import an XPO, you do not need to restart the AOS because an XPO is considered to be a developer artifact. Do not import XPOs for administration scenarios or for shipping solutions. Use .axmodel files for administration scenarios and to ship solutions.

Do not modify elements in the layer below the highest customized layer

If an element is modified in a layer, but a customization exists in a layer above it, the element modification will be stored in the highest layer if permissions are sufficient. This should be avoided unless absolutely necessary because the results may be unpredictable.

Consider Visual Studio to be a separate development client

Visual Studio is used in many development scenarios. If you develop under a shared AOS environment, the Visual Studio client should be considered as another developer client. All rules that apply to developer clients also apply to Visual Studio clients.

Use a separate AOS to debug IL code

When you debug code that runs on Intermediate Language (IL), typically batch jobs and services, you should start a separate AOS instance to use for debugging. Otherwise, the AOS may be blocked during debugging and other developers will not be able to work while one developer is debugging IL code.

Do not add an element to version control in a layer above the layer you are in

Each element exists in a layer. If you modify an element in the VAR layer and log into the ISV layer to add the element to version control, you will get the following error:

```
Error: <Element path> can't be added to version control as it does not exists in isv layer
```

Do not expect version control history to show the history of client operations

Version control history in the developer client shows the history of operations against the version control system. It does not show the history of operations on the client. For example, it will not show checked out operations because a checkout does not impact the version control database.

TFS Guidelines

It is possible to set up your version control system to use TFS in a shared AOS development environment by using public workspaces in Visual Studio 2010. This configuration has the following limitations:

- All users share the same local repository folder.
- All open changes are shared across all the developers. This means any developer that has access to the AOT can change an element that is checked out by another developer.
- Any file that is checked out appears to be checked out to the owner of the workspace. If a developer checks out a file in a workspace that another developer created, the file appears to be checked out to the developer who created the workspace.
- Any file that is checked out is effectively checked out to the workspace. A developer can check in a file that was checked out by another developer. This is the case even if the Enable multiple check-

out flag is set. This flag locks a file, but it is locked to a workspace and not to a user of the workspace.

To use a TFS public workspace for version control, enable TFS version control in Microsoft Dynamics AX. This creates a TFS workspace. Then edit the workspace.

Enable TFS version control

1. In the developer workspace in Microsoft Dynamics AX, click **Version control > Version Control Parameters**.
2. In **Version control status**, select **Enable**.
3. In **Version control system**, select **Team Foundation Server**.
4. In **Repository folder**, enter the path of the repository folder.
5. Click the **Team Foundation Server** tab.
6. In the **Team Foundation Server URL** field, enter the URL of the server that has the TFS project.
7. In the **Team Foundation project name** field, enter the name of the TFS project.

For more information about setting up version control, see the [Microsoft Dynamics AX version control documentation](#) on MSDN.

Edit the TFS workspace

Open the workspace edit utility. You can do this by using the tf.exe tool or Team Explorer.

Open the workspace edit utility by using the tf.exe tool

1. Open tf.exe.
2. Execute the following command:

```
tf workspace /server:<server URI> <workspace name>
```

For example,

```
tf workspace /server:http://MyServer.domain.com:8080/ UserWorskpace
```

Open the workspace edit utility by using Team Explorer

1. Open Source Control Explorer.
2. In the **Workspace** field, select **Workspaces**.
3. Select the workspace that you created when you enabled version control in Microsoft Dynamics AX.
4. Click **Edit**.
5. In **Permissions**, select **Public workspace**.
6. Click **OK** and then close the Manage Workspaces window.

When you use TFS with MorphX, a TFS workspace is used as the local repository for the XPO files that reflect the elements in the AOT.

Visual Studio 2010 has a public workspace feature. A public workspace allows multiple users to work against a single workspace. Multiple developers can check out, change, and check-in elements, and the change is attributed to the developer who makes the change.

Known Issues

Be aware of the following known issues.

Compiling an element creates a new version of the element

Even if a developer does not edit an element, compiling it will result in a new version of the element. This means that if you open an element and edit it, and then someone else opens the same element and compiles it, you will get the following error message:

```
Unable to save since the version of the <element type> on the server is newer
```

Changes to temporary tables require you to clear the metadata cache in the report server

In some cases, changes to temporary tables used to cache report data sets created using Report Data Provider classes will require you to clear the metadata cache in the report server. This issue is surfaced in the Visual Studio compiler by the following error message when building the report:

```
An error has occurred during report processing. (rsProcessingAborted)  
Query execution failed for dataset '[DataSet]'. (rsErrorExecutingCommand)
```

You must restart the report server, rebuild, and redeploy the report.

Hotswapping required on CIL runtime so that new sessions pick up the latest changes

When CIL is generated on a particular AOS, the generated managed assemblies are stored in the database. The CLR (common language runtime) requires that the assemblies are present in the file system, so they must be copied to the file system of the machine that the AOS is running on. This happens when the AOS process is started. This means that each AOS must be restarted to reflect the changes.

Data is lost when inserting data into a new table column

Modifying table metadata after it has been used to create, read, update, or delete data may lead to unpredictable behavior and data loss. Whenever a table is used to perform CRUD operations, the structure of the table is cached on the AOS to improve runtime performance. If the table metadata is modified after these CRUD actions are performed, the structure of the cached table gets out of sync with the structure of the table in the database. Creating new fields and deleting or renaming existing fields are examples of specific actions that cause the structure of the table to change. If these actions are performed, the AOS should be restarted to clear the cached statements.

Only one developer at a time can deploy static reports with assemblies.

If two developers try to deploy static reports with assemblies at the same time, one developer will get a Microsoft SQL Server® Reporting Services connection error.

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.

© 2011 Microsoft Corporation. All rights reserved.

Microsoft, Microsoft Dynamics, the Microsoft Dynamics logo, SQL Server and Visual Studio are trademarks of the Microsoft group of companies.

All other trademarks are property of their respective owners.

Microsoft