Microsoft Dynamics® AX 2012

Code Upgrade Overview for Microsoft Dynamics AX 2012

White Paper

This document describes changes in the code upgrade process for Microsoft Dynamics AX 2012

Date: April 2011

Author: Andrew Jewsbury, Senior Program Manager

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

Microsoft Dynamics®

Table of Contents

Overview

A Microsoft Dynamics® AX 2012 upgrade contains several steps, as shown in the following illustration.

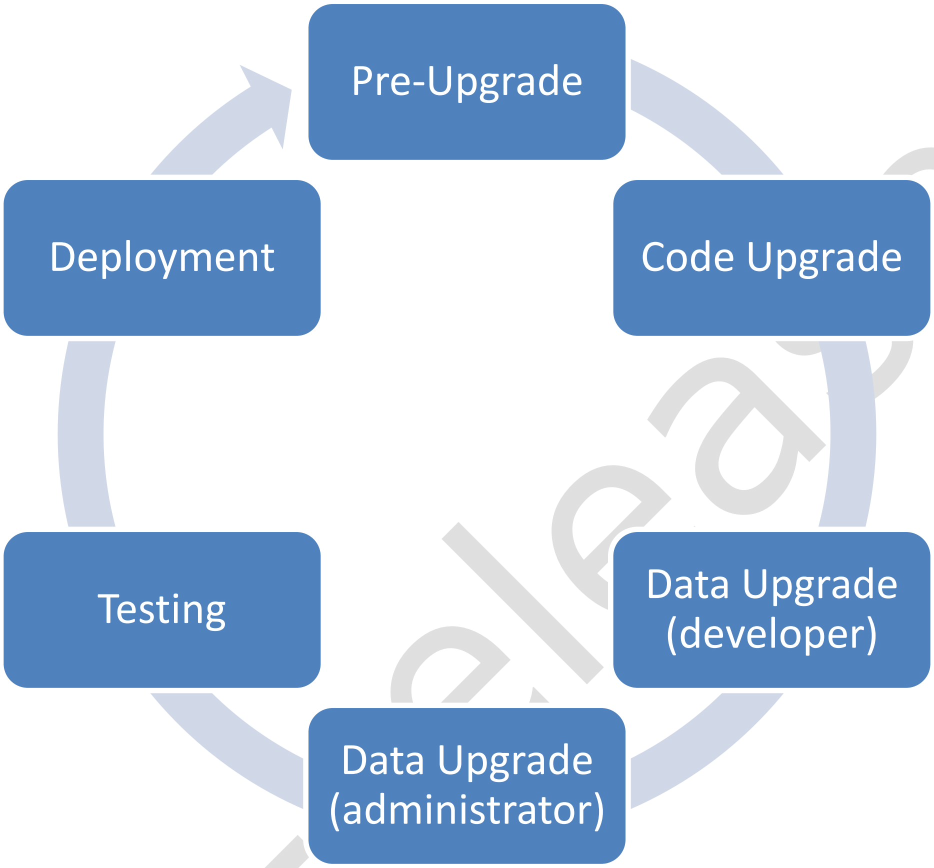


Figure 1: The upgrade process

This white paper describes the first two steps in the process: pre-upgrade and code upgrade. Data upgrade is described in the Microsoft Dynamics AX 2012 TechNet library, in the section Upgrade to Microsoft Dynamics AX 2012.

Document purpose

The purpose of this document is to enable partners to successfully plan for and implement a code upgrade to Microsoft Dynamics AX 2012.

Planning for a code upgrade

Each code upgrade is different. There are three key considerations for your code upgrade:

1. **New features that replace existing customizations.**

   In some cases new Microsoft Dynamics AX 2012 features will replace your customizations. When planning a code upgrade, you must identify these customizations and choose not to upgrade them because they have been replaced by Microsoft Dynamics AX 2012 features.

2. **Microsoft Dynamics AX 2012 refactoring that impacts your customizations.**

   In some cases in which Microsoft Dynamics AX 2012 features have changed, the changes will impact your customizations. You must identify which changes impact your customization. There are two categories here:

   - Required changes
   - Optional changes

   Further guidelines are provided below.

3. **Upgrading of standalone partner code.**

   In some cases, partners have added code (tables, classes, reports, or forms) that do not customize Microsoft code. In this case, there are no direct dependencies on Microsoft. The work here is to ensure that the metadata/code runs and compiles on Microsoft Dynamics AX 2012. Partners can optionally uptake Microsoft Dynamics AX 2012 features in their metadata.

Code upgrade tasks

When planning your code upgrade, you will need to consider which Microsoft Dynamics AX 2012 changes are required, and which are recommended.

The Microsoft Dynamics AX 2012 components are shown in the following figure:

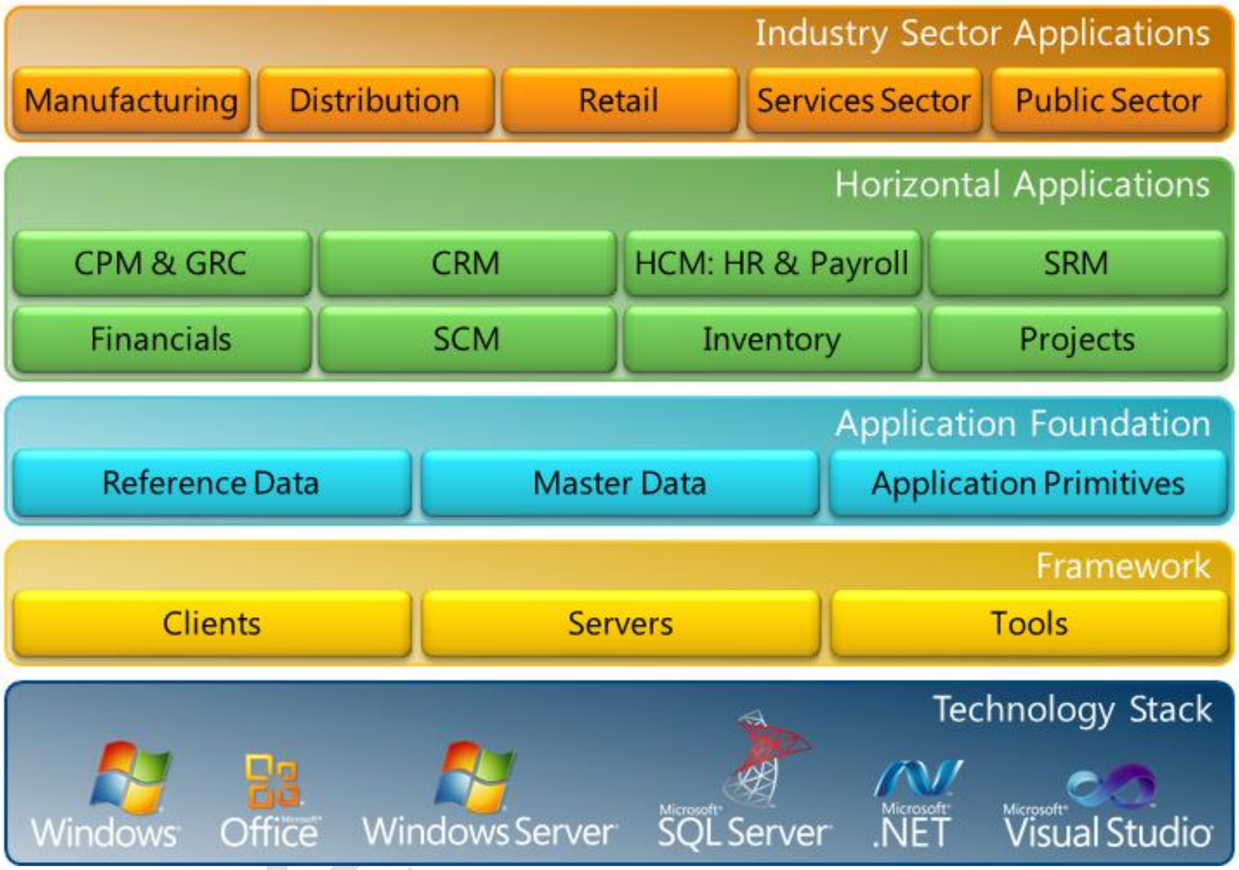


Figure 2: Microsoft Dynamics AX 2012 components

The table below lists the code upgrade tasks for Microsoft Dynamics AX 2012. A code upgrade project may or may not require these tasks. It is important that partners analyze which tasks impact their upgrade. Details and resources are provided for each task later in this document.

Key code upgrade tasks	Component	Importance
Number sequences	Application Foundation	Required
Address book framework	Horizontal Applications	Required
Account and financial fimensions	Horizontal Applications	Required
Human resource framework	Horizontal Applications	Required
Item-Product data management framework	Horizontal Applications	Required
Budget control framework	Horizontal Applications	Required
Table modifications (Required)	Framework	Required
Code changes (Required)	Framework	Required
Help server	Framework	Required
Table modifications (Recommended)	Framework	Recommended
Code changes (Recommended)	Framework	Recommended
SQL Server Reporting Services (SSRS) Reporting	Framework	Recommended
User Experience (UX) patterns	Framework	Recommended
EDT relations	Framework	Recommended

Setup and configuration

This section describes the steps to take during setup and configuration.

Before you begin

- You must have a valid developer license installed on the system in order to view the code upgrade checklist.
- Before you start to upgrade your code to Microsoft Dynamics AX 2012, remember to create and keep a backup of the code you plan to upgrade (for example, your Microsoft Dynamics AX 4.0 or Microsoft Dynamics AX 2009 system).

Baseline database

The first step, after planning for a code upgrade, is to configure a new installation for a code upgrade. The goal here is to create a copy of the Microsoft Dynamics AX 4.0 or Microsoft Dynamics AX 2009 code in a read-only "baseline" database. This baseline database is used for reference purposes during your code upgrade. The imported layers are the code that will be upgraded to Microsoft Dynamics AX 2012. The first step is to create a baseline database during setup.

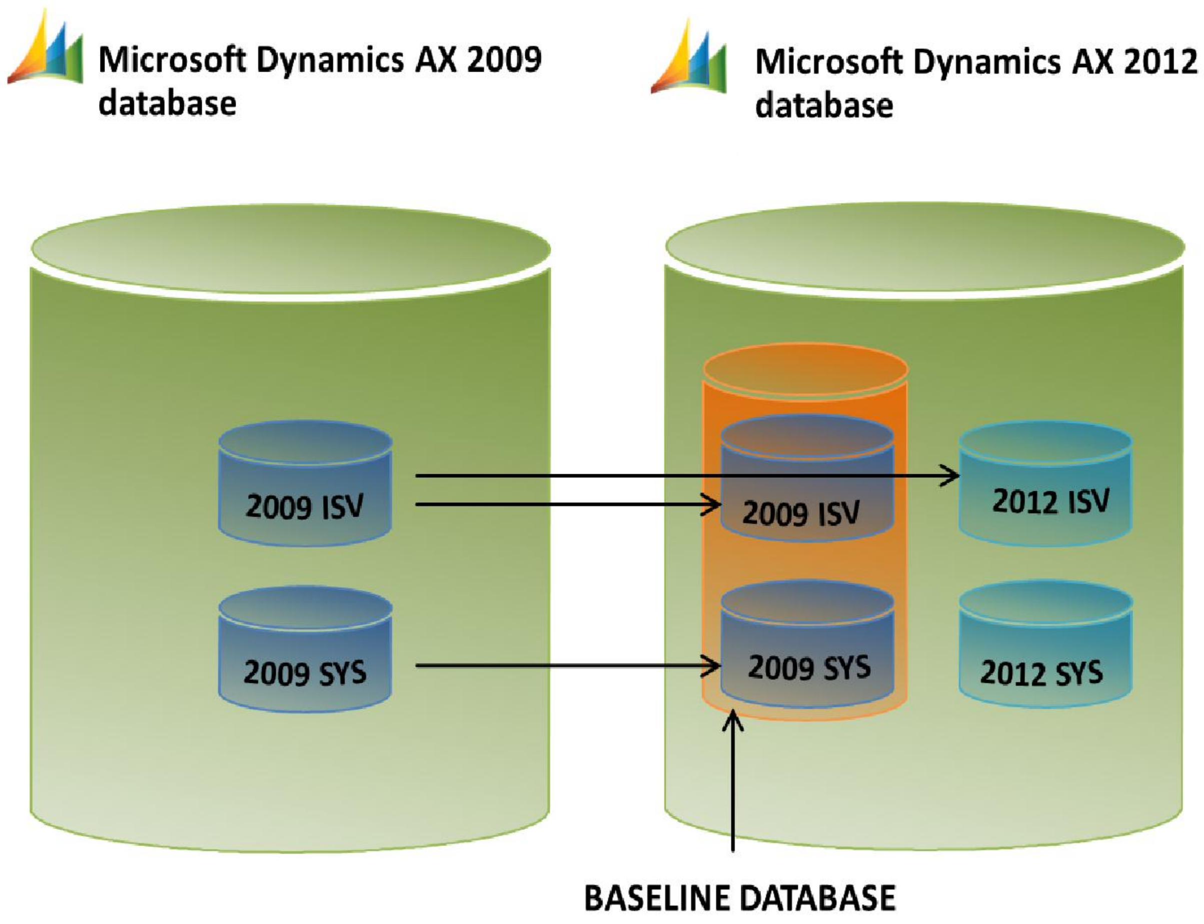
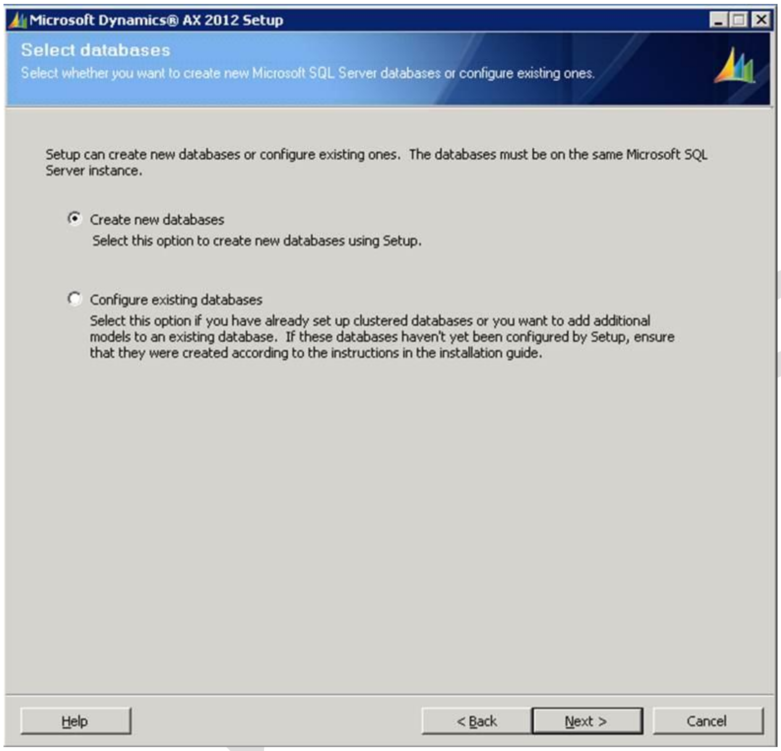

Microsoft Dynamics AX 2009 database
Microsoft Dynamics AX 2012 database
2009 ISV
2009 SYS
2009 ISV
2009 SYS
2012 ISV
2012 SYS
BASELINE DATABASE

Figure 3: Creating a baseline database

Create baseline database during setup

If you are planning to upgrade code from a previous release, create a baseline database during setup.

1. In the Setup dialog box, select **Create new databases** and click **Next**.



Select databases

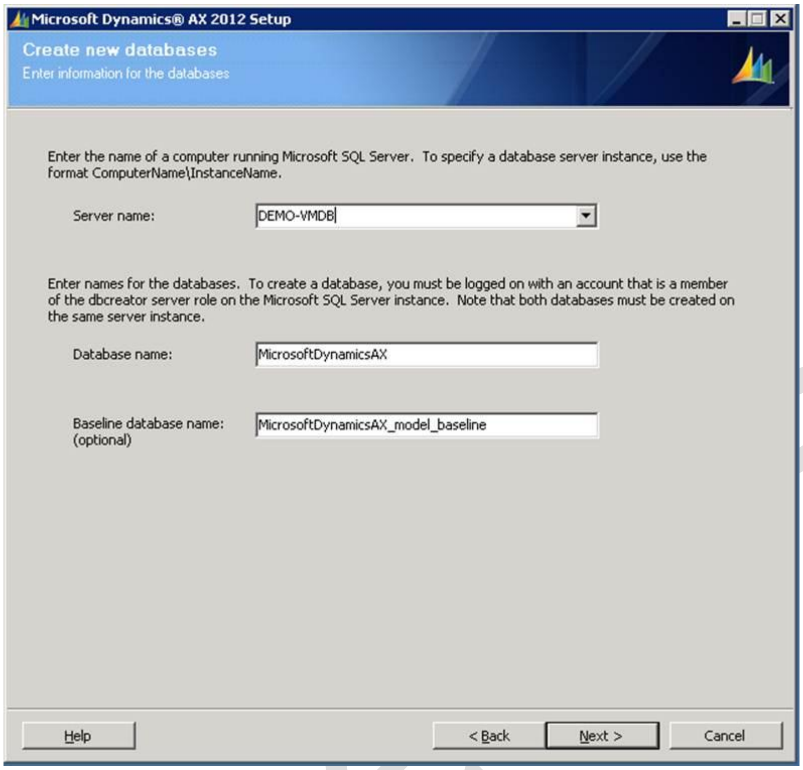
Select whether you want to create new Microsoft SQL Server databases or configure existing ones.

Setup can create new databases or configure existing ones. The databases must be on the same Microsoft SQL Server instance.

● Create new databases
Select this option to create new databases using Setup.

○ Configure existing databases
Select this option if you have already set up clustered databases or you want to add additional models to an existing database. If these databases haven't yet been configured by Setup, ensure that they were created according to the instructions in the installation guide.

2. In the **Server name** box, select the server, enter the database name, and then enter a name for the baseline database. Click **Next** to continue.



Microsoft Dynamics® AX 2012 Setup
Create new databases
Enter information for the databases

Enter the name of a computer running Microsoft SQL Server. To specify a database server instance, use the format ComputerName\InstanceName.

Server name: DEMO-VMDB

Enter names for the databases. To create a database, you must be logged on with an account that is a member of the dbcreator server role on the Microsoft SQL Server instance. Note that both databases must be created on the same server instance.

Database name: MicrosoftDynamicsAX

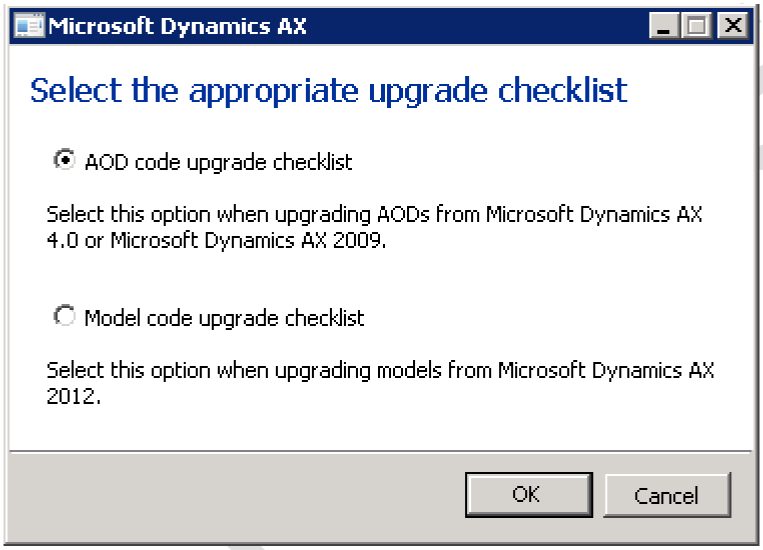
Baseline database name: (optional) MicrosoftDynamicsAX_model_baseline

Choose an upgrade checklist

When you install Microsoft Dynamics AX, select "Register database for Upgrade Mode." When you open Microsoft Dynamics AX 2012, you can choose which type of upgrade to perform. You may choose one of these options:

- AOD code upgrade
- Model code upgrade

The AOD code upgrade checklist is for code upgrades from pre-Microsoft Dynamics AX 2012 to Microsoft Dynamics AX 2012. The Model code upgrade checklist is for upgrades from Microsoft Dynamics AX 2012 to later Microsoft Dynamics versions.



Microsoft Dynamics AX

Select the appropriate upgrade checklist

● AOD code upgrade checklist

Select this option when upgrading AODs from Microsoft Dynamics AX 4.0 or Microsoft Dynamics AX 2009.

○ Model code upgrade checklist

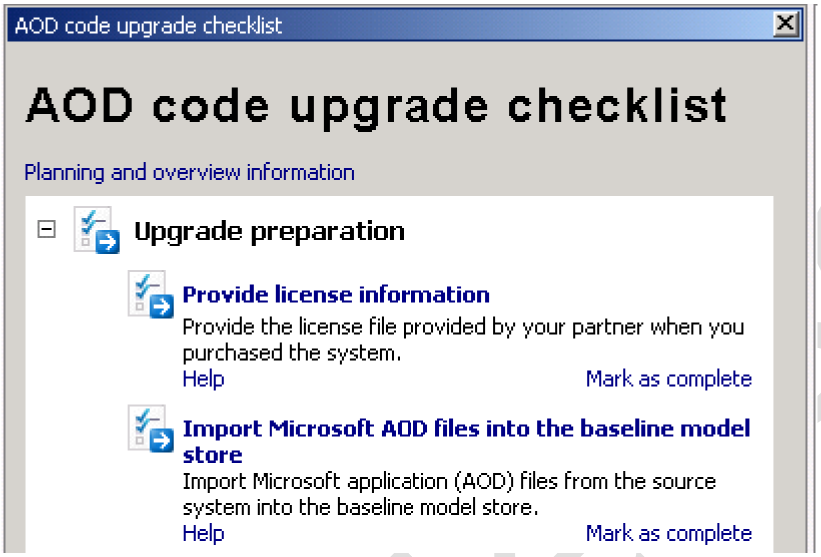
Select this option when upgrading models from Microsoft Dynamics AX 2012.

Figure 4: Select the appropriate upgrade checklist dialog box

Once you have chosen the AOD code upgrade checklist, you must complete the "Upgrade preparation" and the "AOD code upgrade" sections.

Upgrade preparation

Figure 5 shows the Upgrade preparation section of the AOD code upgrade checklist:



AOD code upgrade checklist

AOD code upgrade checklist

Planning and overview information

⊟ Upgrade preparation

Provide license information

Provide the license file provided by your partner when you purchased the system.

Help   Mark as complete

Import Microsoft AOD files into the baseline model store

Import Microsoft application (AOD) files from the source system into the baseline model store.

Help   Mark as complete

Figure 5: Upgrade preparation steps

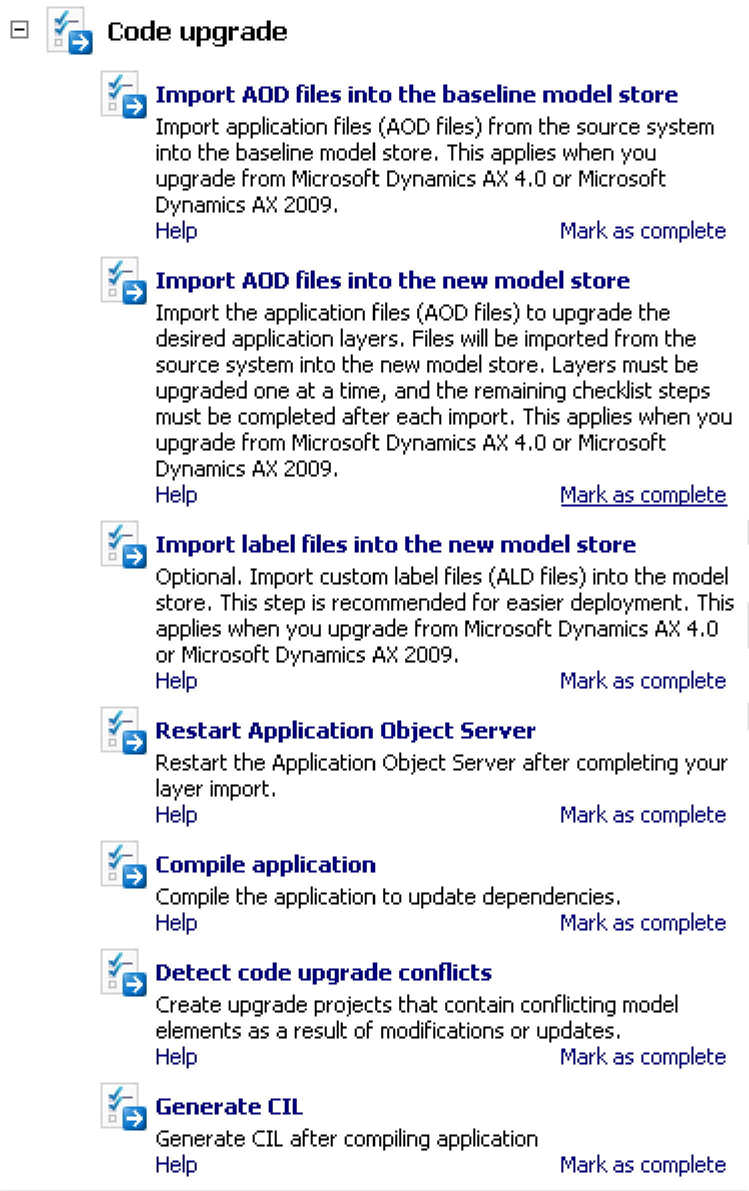
In this step you must complete two tasks:

1. Provide relevant license information.
2. Import all Microsoft AOD files into the baseline model store. In this step, you should import all Microsoft Dynamics AX 4.0 or Microsoft Dynamics AX 2009 Microsoft AODs.

   You should import layers from the lowest to the highest, for example SYS first, then SYP, and so on, until you reach the last Microsoft AOD file.

AOD code upgrade

Figure 6 shows the AOD code upgrade section of the checklist.



Code upgrade

Import AOD files into the baseline model store
Import application files (AOD files) from the source system into the baseline model store. This applies when you upgrade from Microsoft Dynamics AX 4.0 or Microsoft Dynamics AX 2009.
Help Mark as complete

Import AOD files into the new model store
Import the application files (AOD files) to upgrade the desired application layers. Files will be imported from the source system into the new model store. Layers must be upgraded one at a time, and the remaining checklist steps must be completed after each import. This applies when you upgrade from Microsoft Dynamics AX 4.0 or Microsoft Dynamics AX 2009.
Help Mark as complete

Import label files into the new model store
Optional. Import custom label files (ALD files) into the model store. This step is recommended for easier deployment. This applies when you upgrade from Microsoft Dynamics AX 4.0 or Microsoft Dynamics AX 2009.
Help Mark as complete

Restart Application Object Server
Restart the Application Object Server after completing your layer import.
Help Mark as complete

Compile application
Compile the application to update dependencies.
Help Mark as complete

Detect code upgrade conflicts
Create upgrade projects that contain conflicting model elements as a result of modifications or updates.
Help Mark as complete

Generate CIL
Generate CIL after compiling application
Help Mark as complete

Figure 6: Code upgrade steps

You will complete this checklist multiple times, once for each layer that you are upgrading. It is important that you start at the lowest layer (for example, ISV). After the lowest layer is complete, start on the next layer up. Perform this task sequentially until all layers are upgraded.

Before you upgrade

Before you upgrade your code to Microsoft Dynamics AX 2012, be aware of the following:

- If you have customized any of the hybrid classes (\Classes\Session, \Classes\Application, \Classes\Company, \Classes\Info, \Classes\ClassFactory, or \Classes\VersionControl), *remove the customizations* before upgrading to Microsoft Dynamics AX 2012. If you do not do this, this is done automatically for you on AOD import into the current model store. Failure to do this may destabilize Microsoft Dynamics AX 2012 during upgrade. To manually import these classes, first export them on your Microsoft Dynamics AX 4.0 or Microsoft Dynamics AX 2009 system to an XPO file. Then, import this XPO file into the Microsoft Dynamics AX 2012 system by clicking **Import** from the AOT toolbar.

- If a table supports inheritance in Microsoft Dynamics AX 2012 and has been customized in your earlier installation, it will not be imported. To manually import this table, first export it on your Microsoft Dynamics AX 4.0 or Microsoft Dynamics AX 2009 system to an XPO file (include the export option to export with ID and layer). Then, import this XPO file into the Microsoft Dynamics AX 2012 system. However, it may be necessary to modify the design of the table.
- You cannot import a model file from an earlier build than the Microsoft Dynamics AX 2012 January CTP (build 788.35) release. To manually import a model from the January CTP, create a project by using **Create project from model tool**. Then, import the project into the Microsoft Dynamics AX 2012 system by clicking **Import** from the toolbar of the **Projects** window.
- You cannot import deprecated metadata into Microsoft Dynamics AX 2012. Deprecated metadata will be removed at AOD import.
- During AOD import, IDs that are in conflict are rejected. During XPO import, the IDs of application objects might change. Do not attempt to change the IDs back to the values they had in a previous release. To resolve the ID change, set the legacyID property to the old ID before it is imported. This step is only required if you have packed business data in containers.

AOD upgrade process

This section describes how to import AOD files to the baseline model store and the new model store.

To import AOD files to the baseline model store

1. Create a folder named **Old** on the Microsoft Dynamics AX 2012 server in the following location:

   %ProgramFiles%\Microsoft Dynamics AX\60\Server\<AOS instance name>\bin\Application\Appl\Standard

   Standard is the default value. This path is the same as the Server Config value.

2. Copy the AOD files from the source system to the folder created in Step 1.

   Note: You must copy all AOD files, even for the Microsoft layers.

3. Some layers have been renamed with a new prefix in Microsoft Dynamics AX 2012. Rename any existing AOD files to the name of the new layer as follows:

Microsoft Dynamics AX 4.0 Layer	Microsoft Dynamics AX 2009 Layer	Microsoft Dynamics AX 2012 Layer (rename to this)
axbup.aod	axbup.aod	axisp.aod
axbus.aod	axbus.aod	axisv.aod
axlop.aod	axsl3.aod	axslp.aod
axlos.aod	axsl2.aod	axsln.aod
axdip.aod	axsl1.aod	axfpp.aod
axdis.aod	axhfx.aod	axfpk.aod

4. From the **Import Microsoft AOD files into the baseline model store** dialog box, select the name of the layer file to import. You must import the Microsoft layers at the bottom of the list first, and then the higher-level layers. Each layer must be imported one at a time.

Repeat this step until all Microsoft layers have been imported.

5. From the **Import AOD file into the baseline model store** dialog box, select the lowest-level layer file that you want to upgrade.
6. Click **OK** to import the AOD file.

To import AOD files to the new model store

Important: Import only the non-Microsoft layers into the new model store.

1. Copy the layer file that you want to upgrade from the source system to this folder:

   %ProgramFiles%\Microsoft Dynamics AX\60\Server\<*AOS_instance_name*>\bin\Application\Appl\Standard

   Do not copy the AOD files for the Microsoft layers into this folder.
2. Some layers have been renamed in Microsoft Dynamics AX 2012. Rename any AOD files to the name of the new layer as follows:

Microsoft Dynamics AX 4.0 Layer	Microsoft Dynamics AX 2009 Layer	Microsoft Dynamics AX 2012 Layer (rename to this)
axbup.aod	axbup.aod	axisp.aod
axbus.aod	axbus.aod	axisv.aod
axlop.aod	axsl3.aod	axslp.aod
axlos.aod	axsl2.aod	axsln.aod
axdip.aod	axsl1.aod	axfpp.aod
axdis.aod	axhfx.aod	axfpk.aod

3. From the **Import AOD files into the new model store** dialog box, select the name of the layer file to import.

   You must start with the lowest layer first.

   Important: Do not import the Microsoft layers into the new model store.
4. Select the model to import the AOD file into.
5. Click **OK** to import the AOD file.

   An Infolog message may appear if there are items in the AOD file that cannot be imported. See the log file in the message for more information. The most common reasons for an application object failing to import are as follows:

   - A method was added to a class that no longer exists in the AOT.
   - There is an ID conflict between two elements that have the same name and type but with different IDs.
   - You have customized a hybrid class or a table that supports inheritance.

   To upgrade these conflicts, first export the application objects from your Microsoft Dynamics AX 4.0 or Microsoft Dynamics AX 2009 system that failed to import to an XPO file. Then, import this XPO file into the Microsoft Dynamics AX 2012 system by clicking **Import** from the AOT toolbar. You can perform this step before or after completing upgrade.
6. Restart the AOS.

7. Continue to the next steps in the code upgrade checklist for the layer file that you have just imported.
8. After you have completed the code upgrade checklist for one layer, repeat this procedure for the next layer file. Remember that lower-level layer files must be imported before higher-level layer files.

To import label files to the new model store

1. On the Microsoft Dynamics AX 4.0 or Microsoft Dynamics AX 2009 source system, use the **Label file Wizard** to create a label file for the layer that you are working with.
2. Copy the label file to the Microsoft Dynamics AX 2012 system.
3. From the **Select file** dialog box, select the name of the label file to import. You should select the label file that corresponds to the AOD file that you imported in the code upgrade checklist.
4. Click **OK** to import the label file.
5. Continue to the next step in the code upgrade checklist for the label file that you have just imported.
6. After you have completed the code upgrade checklist for one layer, repeat this procedure for the next layer's label file, after you have imported that layer.
   Remember that lower-level label files must be imported before higher-level label files.

Model code upgrade process

To complete a model code upgrade, the steps are very similar to those for upgrading from an AOD file. The steps include:

- Set up the new version of Microsoft Dynamics AX that you are upgrading to.
- Import the code, as models, into a baseline model store and new model store.
- Run the upgrade tools.

To install the Microsoft code into the baseline database, the partner will need to take the signed Microsoft model files (from the version they are upgrading from) and import them into the baseline model store. Partners cannot export the Microsoft code and import it into the baseline model store. That's because, to install the code into the Microsoft layers, it is necessary that the model be signed by Microsoft.

To install the partner code into the baseline and new version, developers will need to do the following:

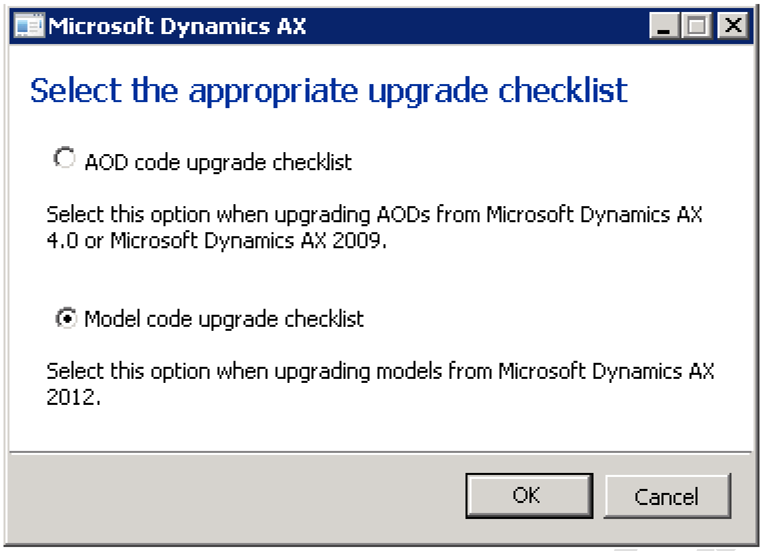
- On the earlier install, **Export** the code as one or more models per layer via AxUtil, for example:

  axutil export /model:ISV /file:*filename*

- On the later install, **Import** the model or models per layer via AxUtil, for example:

  axutil import /file:*filename*

Developers should follow the steps in the Model Upgrade Checklist selected at the following dialog box.



Select the appropriate upgrade checklist

○ AOD code upgrade checklist

Select this option when upgrading AODs from Microsoft Dynamics AX 4.0 or Microsoft Dynamics AX 2009.

● Model code upgrade checklist

Select this option when upgrading models from Microsoft Dynamics AX 2012.

Figure 7: Select the appropriate upgrade checklist dialog box

If import fails

If the model import fails, it could be because some child nodes have missing parents.

The /createparents switch is used with the import command to create parents. If a child-element is imported, but its parent element is not in the model store, a fake parent is created in the model store.

The reason for this is because, when you upgrade to a new version, it is possible that the parent has been removed in a lower layer. If that is the case, you need a fake parent to import the model.

In this case, the developer should import from the AxUtil command line utility by using this command:

axutil import /file:*filename* /createparents

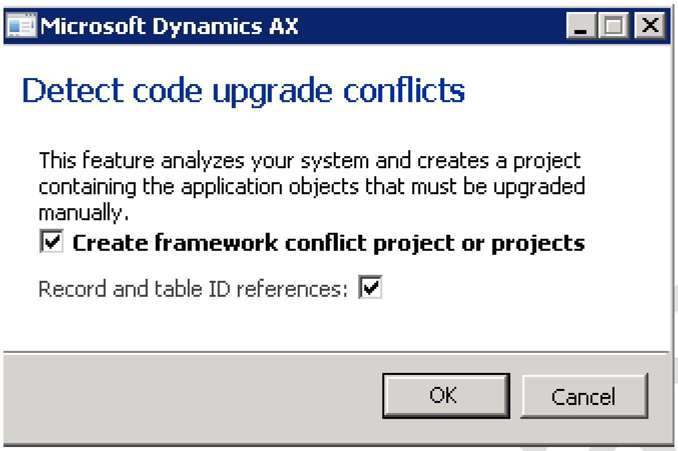
Detect code upgrade conflicts

Microsoft Dynamics AX 2012 includes a **Detect code upgrade conflicts** tool that analyzes your installation, optionally deletes duplicate objects, and creates a project that contains the application objects with conflicts. This tool is used when you upgrade from Microsoft Dynamics AX 4.0 or from Microsoft Dynamics AX 2009.

The project contains the objects that have been changed both in the new release and in your Application Object Layer.

To create an upgrade project by using the Detect code upgrade conflicts tool

1. From the Microsoft Dynamics AX client menu, click **Tools > Code upgrade > Detect code upgrade conflicts**.



Detect code upgrade conflicts

This feature analyzes your system and creates a project containing the application objects that must be upgraded manually.

☑ **Create framework conflict project or projects**

Record and table ID references: ☑

2. Optionally select **Create framework conflict project or projects**, and then select the **Record and table ID references** check boxes if you want to create separate projects for those conflicts.

3. Click **OK.**

   A new upgrade project or projects is created.

   Important: For the Microsoft Dynamics AX 2012 Convergence 2012 CTP build, no forms will appear in the project even if there is a conflict.

To manually apply changes to forms

In many cases, if a change between the original form and the new form is large, you may need to manually apply changes to a form to upgrade it.

1. Delete the form from the layer that you have customized it in.

2. Open the form from the baseline model store. From the AOT, right-click the form, select **Add-Ins > Open new window > Open old layer**, and then select the layer that contains the form you customized.

3. The form from the baseline model store opens.

4. Manually apply the changes from the form in the baseline model store to the new form.

Choosing alternatives to the Detect code upgrade conflicts tool

You may not want to use the Detect code upgrade conflicts tool when you upgrade. Alternatives include:

- The **Compare layers** tool.
- The **Project filter** tool.

Compare layers tool

As an alternative to the **Detect code upgrade conflicts** tool, you can use the **Compare layers** tool to compare any two layers and create a project with the objects that differ.

Unlike the **Detect code upgrade conflicts** tool, you cannot choose to delete duplicates. Consequently the **Compare layers** tool may also be useful as a general tool (for example, to give an overview of modifications made in a certain layer).

Project filter tool

The **Project filter** tool lets you create a project that is based on criteria supplied in the query form. For example, such criteria could be all objects from a relevant layer, all objects that have a specific prefix, or all objects created by a specific user.

Code upgrade FAQ

Question: In what sequence should I approach my code upgrade?

Answer: The code upgrade sequence will vary per project. Typically, however, it is best to focus on upgrading the lowest-level concept first and then build up. Therefore, start with the data model (the tables/schema), then the business logic, and finally the presentation layer. Completing an upgrade on the lowest-layer concepts first ensures that work does not have to be redone.

Additionally, it is important to sequence the Microsoft Dynamics AX components that you upgrade.

Question: Can I delete my layer in my baseline database?

Answer: Yes. AxUtil is command line utility that allows you to delete layers. To delete a layer from the baseline database the command is:

axutil delete /layer:isv /db: <*baseline_model_store_name*>

Question: Do I upgrade my security artifacts?

Answer: Prior versions of Microsoft Dynamics AX did not include any security settings out of the box. The Microsoft Dynamics AX 2012 release is the first release in which Microsoft is shipping security definitions as per the Roles, Duties, Privileges, and securable controls. Therefore, you should not expect to migrate security definitions to Microsoft Dynamics AX 2012.

You must start over by provisioning users with standard Microsoft roles and then implement any additional changes. This step is necessary to account for deviations from these Microsoft standards to match the company security policies.

Question: What does an ISV or partner need to do to get their Microsoft Dynamics AX 2009 code ready for use in Microsoft Dynamics AX 2012 non-admin mode?

Answer: Outside of accounting for a few deprecated APIs (for example, hassecuritykeyaccess), there should be no need for other changes. ISVs or partners do, however, need to create and customize role definitions.

Question: Is security optional or required?

Answer: Microsoft Dynamics AX 2012 has an out-of-the-box security model. We expect users to account for deviations from the standard roles provided.

Question: Forms have undergone a major change (both user experience and data model). How expensive will it be to upgrade my forms?

Answer: The new user experience (UX) patterns will greatly improve business productivity. To facilitate upgrading to the new UX patterns, we have provided detailed documentation on each new pattern. In addition, we have provided utilities that help validate the forms pattern.

Question: Is upgrading my forms to the new UX patterns required or optional?

Answer: It is optional. However you need to consider your business scenarios. End users often benefit from interacting with business forms that are consistent and provide a consistent UX pattern. This is up to each implementation to decide.

Question: What tooling is there to help me upgrade my forms?
Answer: The **Forms Style Checker** tool validates that forms meet the new UX guidelines in the product. In addition, it allows you to create new forms from a template.

Additional resources

Microsoft has provided extensive documentation on changes that impact an upgrade.

Much of the documentation can be found in the Microsoft Download Center, on the Code Upgrade white papers landing page. Additional papers will be added over time.

Additionally, some videos are available on the Microsoft Connect site.

The following white papers are available on the Code Upgrade white papers landing page, or through the direct links below.

Financials

- Implementing the Account and Financial Dimensions Framework Microsoft Dynamics AX 2012
- Implementing the Budget Control Framework for Microsoft Dynamics AX 2012
- Implementing Budgeting for Microsoft Dynamics AX 2012

Supply chain management

- Implementing the Operations Resource Model for Microsoft Dynamics AX 2012
- Implementing InventTrans refactoring for Microsoft Dynamics AX 2012
- Implementing Item-Product data management framework for Microsoft Dynamics AX 2012

HCM: HR and payroll

- Implementing the Address Book Framework for Microsoft Dynamics AX 2012
- Implementing and Updating the Human Resources Framework in Microsoft Dynamics AX 2012

Application primitives

- Using the Enhanced Number Sequence Framework in Microsoft Dynamics AX 2012
- Video: Using the Enhanced Number Sequence Framework: http://connect.microsoft.com/DynamicsAX6TAP/Downloads/DownloadDetails.aspx?DownloadID=33660

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 supports a preliminary release of a software product that may be changed substantially prior to final commercial release. This document is provided for informational purposes only and Microsoft makes no warranties, either express or implied, in this document. Information in this document, including URL and other Internet Web site references, is subject to change without notice. The entire risk of the use or the results from the use of this document remains with the user. Unless otherwise noted, the companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted in examples herein are fictitious. No association with any real company, organization, product, domain name, e-mail address, logo, person, place, or event is intended or should be inferred. Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.