



Microsoft Dynamics AX 2009 Upgrade Guide

Microsoft Corporation

Published: November 2009

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 for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED, OR STATUTORY, AS TO THE INFORMATION 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.

© 2009 Microsoft Corporation. All rights reserved.

Microsoft, Microsoft Dynamics, the Microsoft Dynamics Logo, SQL Server, Active Directory, SharePoint, Visual C++, Windows, Windows Server, Navision, and Axapta are either registered trademarks or trademarks of the Microsoft group of companies.

All other trademarks are property of their respective owners.

Table of Contents

Check for updated upgrade information	5
Introduction to upgrade.....	6
Supported upgrade paths.....	7
System requirements	8
Quickstart: Upgrade from Microsoft Dynamics AX 3.0.....	8
Quickstart: Upgrade from Microsoft Dynamics AX 4.0.....	9
Improvements to the upgrade process	10
Best practices for upgrading	12
Before you upgrade	14
Upgrade tasks for supported upgrade paths.....	15
Set the appropriate user permissions	16
Back up your data	18
Clean up your data	19
Copy your existing application files	21
Prepare Axapta 3.0 data for upgrade to Dynamics AX 2009	22
Upgrade to 64-bit record ID fields.....	23
Left-justify database columns	24
Create a new database.....	25
The Microsoft Dynamics AX DB Upgrade Preparation tool	26
Upgrade Team Server.....	29
Upgrading to Microsoft Dynamics AX 2009.....	30
Install Microsoft Dynamics AX before you upgrade	31
The Upgrade checklist	34
The Data Upgrade Cockpit.....	35
Prepare to upgrade	39
Compile the application (Upgrade)	40
Provide license information.....	41
Time zones and upgrade	43
Set time zone for upgraded data	43
Set customer feedback options	44
Detect customization conflicts	44
The Compare tool	47
How to: Compare Two Application Objects.....	48
How to: Compare Versions of an Object.....	49
How to: Compare Application Object Layers	50
Test the system after code upgrade	52
Create virtual companies for global address books.....	53

Microsoft Dynamics AX

Upgrade your data.....	55
Improving data upgrade performance.....	56
Presynchronize your database	56
Synchronize your database	57
Postsynchronize your database.....	57
Test the system after data upgrade	58
Finalize upgrade.....	58
Configure system accounts	59
Upgrade Enterprise Portal	61
Specify Role Center Web site.....	63
Upgrading AIF.....	64
Additional upgrade tasks.....	65
Application upgrade notes	67
Upgrade notes affecting all modules.....	67
Upgrade notes affecting specific modules	74
Upgrade notes affecting specific locales.....	90
Upgrade troubleshooting	94
Troubleshooting the Microsoft Dynamics AX DB Upgrade Preparation tool.....	95
Troubleshooting AOS or Microsoft Dynamics AX client startup during upgrade	98
Compiling Microsoft Dynamics AX troubleshooting	100
Troubleshooting the Data upgrade cockpit	101
Troubleshooting the Detect code upgrade conflicts tool	109
Error messages during upgrade.....	110
Troubleshooting the Compare tool.....	112
Upgrade Guide appendix.....	113
Development access licenses.....	114
Walkthrough: upgrading date and time table field pairs into UtcDateTime.....	116
Prerequisite software installed by Setup	128
AIF upgrade process	129
Code upgrade: Examples.....	142
Additional code upgrade information.....	146
Additional data upgrade information	150

Check for updated upgrade information

The information contained in this document was current as of November 2009. The documentation may be updated as new information becomes available. For the most current documentation for system administrators, check the [TechNet Library](#). For the most current documentation for developers, check the [MSDN Library](#).

Introduction to upgrade

The Upgrade Guide contains information that will help you be successful in upgrading to Microsoft Dynamics AX 2009. This section contains the following topics.

- [Supported upgrade paths](#)
- [System requirements](#)
- [Quickstart: Upgrade from Microsoft Dynamics AX 3.0](#)
- [Quickstart: Upgrade from Microsoft Dynamics AX 4.0](#)
- [Improvements to the upgrade process](#)
- [Best practices for upgrading](#)

Supported upgrade paths

You can upgrade to Microsoft Dynamics AX 2009 directly from Microsoft Dynamics AX 4.0 and from Microsoft Dynamics AX 3.0 (formerly known as Microsoft Business Solutions—Axapta 3.0) with any supported service pack for any country except Brazil. Any earlier versions require that you upgrade incrementally to either Microsoft Dynamics AX 3.0 with a supported service pack or Microsoft Dynamics AX 4.0 before you can upgrade to Microsoft Dynamics AX 2009.

 **Note:**

Data loss might occur if you upgrade from an unsupported version of Microsoft Dynamics AX that has country specific DIS/DIP layers and if the ID for a table and/or field has changed in any version between your unsupported version and Microsoft Dynamics AX 2009. Identify cases where this situation exists and create an upgrade script to handle the situation automatically, or copy the data manually from the old version database to the Microsoft Dynamics AX 2009 database after the upgrade.

Direct upgrade for Brazil is only available from Microsoft Dynamics AX 4.0 SP2 to Microsoft Dynamics AX 2009.

 **Note:**

The individual layer for Brazil is included in the Microsoft Dynamics AX 2009 consolidated GLS layer. Before you can install a consolidated GLS layer, you must install Service Pack 1 for Microsoft Dynamics AX 2009. For information about upgrading the GLS layer for a specific country, see the [Service Pack 1 page](#) on CustomerSource.

System requirements

For up-to-date hardware and software requirements for Microsoft Dynamics AX, see the system requirements Web page (<http://go.microsoft.com/fwlink/?LinkId=107378>).

Some software prerequisites will be installed automatically by Setup. For more information, see [Prerequisite software installed by Setup](#).

Quickstart: Upgrade from Microsoft Dynamics AX 3.0

This topic lists the high-level steps involved in upgrading from Microsoft Dynamics AX 3.0 to Microsoft Dynamics AX 2009.

1. Back up your existing database and application files.
2. Run the Pre-Upgrade Duplicate Detection form, available on CustomerSource: <http://go.microsoft.com/fwlink/?LinkId=137799&clcid=0x409>
3. Import two .xpo files from the installation media to assist with data upgrade.
 - UpgradeColumnList.xpo, for 32-bit to 64-bit ReclId field conversion.
 - LeftJustified.xpo, for removing any trailing spaces from fields.
4. (Optional) To help improve performance, remove all user data and logs of Microsoft Dynamics AX 3.0. For example, clean up the SysDatabaseLog table.
5. Create an empty database for Microsoft Dynamics AX 2009 in SQL Server 2005.
6. (Optional) To help improve performance, set initial data and log file sizes so that they don't grow excessively large during the data upgrade process.
7. (Optional) To help improve performance, set the recovery model to **Simple** for the Microsoft Dynamics AX 2009 database.
8. Run AXDBUpgrade.exe (the Microsoft Dynamics AX DB Upgrade Preparation tool).

 **Note:**

To help improve performance, you can apply the LeftJustified.xpo to the database created in step 5. Do this after you've used the Microsoft Dynamics AX DB Upgrade Preparation tool as described in step 8 but before you start the Microsoft Dynamics AX 2009 AOS.

 **Note:**

To help improve performance, you can run this tool in multithreaded mode. For example, to run this tool with 10 threads, enter **AxDbUpgrade.exe P/10** at a command prompt.

9. (Optional) Apply the LeftJustify file imported in step 2 to the Microsoft Dynamics AX 2009 database created in step 5.
10. Back up your Microsoft Dynamics AX database. The database is ready to be upgraded.
11. Run the Microsoft Dynamics AX 2009 Setup file from the installation media. During installation, select the database that you created in step 5. For step-by-step instructions, see [Install Microsoft Dynamics AX before you upgrade](#).
12. Install Microsoft Dynamics AX 2009 Service Pack 1. To download and install the service pack, go to <http://go.microsoft.com/fwlink/?LinkId=130877>.
13. Copy your upgraded customized file into the correct application directory.
14. Start the AOS.
15. Start the Microsoft Dynamics AX 2009 client. The upgrade checklist is displayed automatically.
16. Complete the steps in the upgrade checklist to finish upgrading.

Quickstart: Upgrade from Microsoft Dynamics AX 4.0

The steps below provide a high-level overview of the tasks that you must complete to upgrade from Microsoft Dynamics AX 4.0 to Microsoft Dynamics AX 2009.

1. Back up your existing database and application files.
2. (Optional) To help improve performance, remove all user data and logs of Microsoft Dynamics AX 4.0. For example, clean up the SysDatabaseLog table.
3. (Optional) To help improve performance, set initial data and log file sizes so that they don't increase while you perform the data upgrade process.
4. (Optional) To help improve performance, set the recovery model to **Simple** for the Microsoft Dynamics AX 2009 Database.
5. Back up your Microsoft Dynamics AX database. Your database is ready to be upgraded.
6. Run the Microsoft Dynamics AX 2009 Setup file from the installation media. During installation, select your existing Microsoft Dynamics AX database. For step-by-step instructions, see [Install Microsoft Dynamics AX before you upgrade](#).
7. Install Microsoft Dynamics AX 2009 Service Pack 1. To download and install the service pack, go to <http://go.microsoft.com/fwlink/?LinkId=130877>.
8. Start the Microsoft Dynamics AX 2009 client. The Upgrade checklist is displayed automatically.
9. Complete the steps in the Upgrade checklist to finish upgrading.

Improvements to the upgrade process

Additional features are included in Microsoft Dynamics AX 2009 to help you upgrade from previous versions. Each feature is described below.

Batch jobs using the batch framework

The upgrade process for Microsoft Dynamics AX 2009 has been modified to take advantage of the updated batch framework. Upgrade is now run as a set of batch jobs, one for each upgrade step (Presynchronize, Postsynchronize, and Upgrade additional features), on an Application Object Server (AOS) instance. Each upgrade job consists of multiple upgrade tasks, one for each upgrade script. Dependencies between the upgrade scripts are created using batch framework constraints. You can monitor the dependencies between upgrade jobs by using the batch dependencies interface.

Additional tasks in the Upgrade checklist

Previous tasks have been rearranged or renamed and new tasks have been added to the upgrade checklist.

If you have upgraded from a previous version of Microsoft Dynamics AX, the upgrade checklist is displayed the first time you start the Microsoft Dynamics AX 2009 client. The upgrade checklist contains a list of upgrade tasks and the order in which you must complete them. For more information about the Upgrade checklist, see [The Upgrade checklist](#).

Enhanced Data Upgrade Cockpit

The data upgrade cockpit has been redesigned to display task status and error information more clearly. Also, error reporting and SQL statements now are displayed in the Infolog and you can double-click on a failed script to display the script's code.

A progress bar also has been added to the form.

Detecting upgrade conflicts

For any type of upgrade conflict, you now can use the code upgrade toolset. In case of upgrade conflicts, you are still able to delete obsolete elements. A new option is to "Auto-resolve property conflicts." This automatically resolves conflicts where one property is changed in your revision of an AOT node, and a different property is changed in the newest Microsoft revision on the same AOT node. If the same property is changed in both your version and Microsoft's version, it is still marked as a conflict.

Enhanced visual indicators

The code upgrade process includes new visual indicators. Each code conflict and resolved code conflict is flagged with an icon on the affected node. This makes it easier to get an overview of where conflicts are. As you work your way through the conflicts, you can mark a conflict as resolved, which causes the conflict icon to change from a red alert to a green check mark.

Enhancements to the Compare tool

Microsoft Dynamics AX 2009 brings some very helpful improvements to the Compare tool. First, the Compare tool can now be started by a keyboard shortcut (CTRL-G), and it has been promoted to the top level of the context menu. Second, your preferences are now stored, so the next time you open the Compare tool, you don't have to reselect what you want to compare. Third, more information is now available to assist you in making the right decisions more easily. In the Comparison form, the original value is now included, so you now can see three values—your modifications, Microsoft's modifications, and the original value—all in the same form.

Time estimation report for conflict resolution

You now can print a fully configurable report that contains time estimates for solving upgrade conflicts detected using the upgrade checklist. The estimates are broken down to the same granularity as the detected conflicts, which is on a per node level.

Best practices for upgrading

This section contains information to help you have the best upgrade experience possible. Read through the topics below before you begin the upgrade process.

Upgrade in a test environment first

Upgrade to Microsoft Dynamics AX 2009 in a test environment before you upgrade in your production environment.

Test whether your installation is functioning properly in the test environment before you upgrade in your production environment.

To test whether the upgrade to Microsoft Dynamics AX 2009 has succeeded, perform at least these minimum steps:

- Restart all Application Object Server (AOS) instances.
- Launch a client that is connected to each running AOS instance.
- Test that users can execute the most common tasks for your system.
- Test that the administrator can run the system maintenance tasks.
- Test that users can execute the most vital tasks for your system.
- Test that users can read the reports that are generated.
- Test that the permission settings are working right by checking that users have access to the data they had access to prior to upgrade.
- Validate that your data is consistent with that prior to upgrade.

Disk space and log size

Verify that you have the proper amount of space on your hard drive and in your transaction log. See [Improving data upgrade performance](#) for more information.

Database server configuration

Ensure that Microsoft SQL Server is configured for the highest possible performance and throughput. See [Improving data upgrade performance](#) for more information.

The upgrade process should take place on a dedicated server. If there are other applications running on the database server, they will compete with the upgrade process for resources.

Review the performance benchmark of the existing database server and determine if you need to add more resources. These might include moving the database server to a 64-bit computer or upgrading server components such as processor(s), memory, or storage. After you upgrade or change server components, you should test the server for optimum performance before you start the Microsoft Dynamics AX upgrade process.

Backing up your data

It is recommended that you back up your database after successful completion of each of the following upgrade steps so that you can easily restart at any point if necessary.

Upgrade step	Timing
Left-justify database columns	Back up the database that you perform this step on after this step.
The Microsoft Dynamics AX DB Upgrade Preparation tool	Back up the pre-upgraded Microsoft Dynamics AX 2009 database after this step.
Synchronize your database	Back up Microsoft Dynamics AX 2009 database after this step.
Postsynchronize your database	Back up Microsoft Dynamics AX 2009 database after this step.
Additional upgrade tasks	Back up Microsoft Dynamics AX 2009 database after this step.

Delete obsolete application objects

After you upgrade your system, you should disable the **Keep update objects 4.0 (SysDeletedObjects40)** and **Keep update objects 4.1 (SysDeletedObjects41)** configuration keys. Disabling these keys will cause deletion of obsolete application objects that have the prefix DEL_.

 **Note:**

Do not disable this configuration key until your data upgrade is complete.

1. Click **Administration > Setup > System > Configuration**. The **Configuration** form is displayed.
2. In the left pane, expand **Administration**.
3. Unmark **Keep update objects 4.0** and **Keep update objects 4.1**, Click **OK**.

Before you upgrade

You can upgrade to Microsoft Dynamics AX 2009 directly from Microsoft Dynamics AX 4.0 or from Microsoft Dynamics AX 3.0 with a supported service pack. This section contains information about tasks that you need to complete before you can install Microsoft Dynamics AX 2009. This section contains the following topics.

- [Upgrade tasks for supported upgrade paths](#)
- [Set the appropriate user permissions](#)
- [Back up your data](#)
- [Clean up your data](#)
- [Copy your existing application files](#)
- [Prepare Axapta 3.0 data for upgrade to Dynamics AX 2009](#)
- [Upgrade Team Server](#)

Upgrade tasks for supported upgrade paths

The procedures for upgrading data from Microsoft Dynamics AX 3.0 and from Microsoft Dynamics AX 4.0 differ primarily in the amount of preparation required. Upgrading from Microsoft Dynamics AX 3.0 requires additional steps. The following table lists the upgrade tasks for each version.

Upgrade Tasks for Microsoft Dynamics AX 3.0	Upgrade Tasks for Microsoft Dynamics AX 4.0
Set the appropriate user permissions.	Set the appropriate user permissions.
Back up your data , including application files and databases.	Back up your data , including application files and databases.
Copy your existing application files.	Copy your existing application files.
Clean up your data.	Clean up your data.
Install Microsoft Dynamics AX before you upgrade.	Install Microsoft Dynamics AX before you upgrade.
Run the Pre-Upgrade Duplicate Detection form .	
Left-justify database columns on your Microsoft Axapta 3.0 source database.	
Run the The Microsoft Dynamics AX DB Upgrade Preparation tool , AxDbUpgrade.exe.	
Copy your existing application files to Microsoft Dynamics AX 2009 and delete the index file from the standard directory.	Copy your existing application files to Microsoft Dynamics AX 2009 and delete the index file from the standard directory.
Start the Application Object Server (AOS).	Start the Application Object Server (AOS).
Start the Microsoft Dynamics AX 2009 client.	Start the Microsoft Dynamics AX 2009 client.
Complete the tasks in the Upgrade checklist. See The Upgrade checklist .	Complete the tasks in the Upgrade checklist. See The Upgrade checklist .



Note:

We recommend that you read through the [Best practices for upgrading](#) topic before you complete these tasks.

Set the appropriate user permissions

Before you begin the Microsoft Dynamics AX 2009 installation process, work with a system administrator to ensure that the account with which you log onto each server has appropriate permissions.

In all cases, you must be a member of the Administrators group on the local computer where you are upgrading a component.

The table below describes the permissions you need to set before you begin your upgrade. The permissions listed are implemented using the principle of least privilege.

Action	Permissions required for account
Install Application Object Server (AOS).	Member of the securityadmin role on the Microsoft SQL Server computer you want to connect to and full permission to the application
Create a Microsoft SQL Server database.	Member of the dbcreator role on the SQL Server instance
Connect the AOS to a SQL Server database.	Member of Database Security Administrators role on the SQL Server instance
Create an Oracle database.	Member of SYSDBA on Oracle Database Server
Connect the AOS to an Oracle database.	Member of SYSDBA on Oracle Database Server
Install the application file server.	Member of Administrators group on the target computer
Install the Microsoft Dynamics AX client.	Member of Administrators group on the local computer
Install Role Centers and Enterprise Portal framework	Member of the Administrators group in Microsoft Dynamics AX and a member of the dbcreator role on the SQL Server instance being used for Windows SharePoint Services (if WSS is installed through Microsoft Dynamics AX Setup)
Install Workflow	Member of the Administrators group in Microsoft Dynamics AX
Start the Reporting extensions	Member of the Administrators group in Microsoft Dynamics AX

Microsoft Dynamics AX

Action	Permissions required for account
Start the Synchronization service	Member of the Administrators group in Microsoft Dynamics AX
Start the Synchronization proxy	Member of the dbowner database role on the SQL Server database for Microsoft Office Project server, and an administrator on the computer running Office Project Server
Install AIF Web services	Member of the Administrators group in Microsoft Dynamics AX
Install BizTalk adapter	Member of the administrators group in BizTalk Server
Complete Developer installation	All rights that are required to install the database, application files, AOS, client, and .NET Business Connector
Run the Microsoft Dynamics AX Dynamics AX DB Upgrade Preparation Tool.	<p>If you are using SQL Server: The user running the tool must be a member of the ddl_admin, db_datareader, and db_datawriter roles on the source and target databases</p> <p>If you are using Oracle: The user running the tool must be a member of the dba role on the source and target Oracle schemas</p>
Run the upgrade checklist.	Member of the Microsoft Dynamics AX Administrator group

Back up your data

Before upgrade, you should back up the following information.

- Your data. See your SQL Server information at [Microsoft SQL Server documentation](#) or consult your Oracle documentation.
- Your existing application files (*.aod, *.add, *.ald, *.ahd, and *.khd files) from all layers above the DIS layer (LOS, BUS, VAR, CUS, USR) from your existing system. For example, on your existing Microsoft Dynamics AX 4.0 system, copy the application files from \Microsoft Dynamics\4.0\Application\ApplStandard.



Note:

The layers listed here exist only if customizations have been made.

Layers in Microsoft Dynamics AX 2009 have been renamed. Before you upgrade data, you need to change the names of the layers. See [Copy your existing application files](#) for the list of renamed layers.

Clean up your data

Remove data in the Microsoft Dynamics AX 3.0 database that is not needed for audit trails, that slows down performance, or that makes the database unnecessarily large. Before you upgrade your database, select the type of data that you want to clean up by using the standard filtering mechanism.

1. Start Microsoft Dynamics AX 3.0 in your test environment (**Start > All programs > Navision Axapta**), and log on.
2. Open the Application Object Tree (AOT).
3. Under **Menu items > Action**, open and run the following forms to clean up the respective types of data:
 - SysUserLogCleanUp
 - SysDatabaseLogCleanUp
 - ProdJournalCleanup
 - InventJournalCleanUp
 - ProdTableCleanUp
 - SalesParmCleanUp
 - PurchParmCleanUp
 - LedgerJournalCleanUp

Database accent-sensitivity consideration

The Microsoft Dynamics AX 2009 setup program requires that your database be accent-sensitive. If the Microsoft Dynamics AX database you are upgrading is accent-insensitive, you must take additional actions before you upgrade.

1. Evaluate your data for potential issues with accented characters. For example, in an accent-insensitive database, *café* and *cafe* are considered identical values. In an accent-sensitive database, they are different values.

If it matters to you that words with accented characters will be treated differently after you convert to an accent-sensitive collation sequence, then you will need to convert your data to account for the differences. In this example, you may want to convert *café* to be *cafe* so that the values are identical when you choose an accent-sensitive database collation.

2. Choose a SQL collation sequence that supports accent-sensitive data.

For more information, see the SQL Server documentation or [The Impact of Changing Collations and of Changing Data Types from Non-Unicode to Unicode](#) white paper.

Pre-Upgrade Duplicate Detection form

The upgrade scripts for Microsoft Dynamics AX 4.0 and Microsoft Dynamics AX 2009 require that duplicate record IDs within a company do not exist. Duplicate record IDs within a company add unique index constraints into some of the tables in the Microsoft Dynamics AX 4.0 and Microsoft Dynamics AX 2009 databases. Duplicate record IDs may exist in Microsoft Dynamics AX 3.0 databases. To avoid running into problems during upgrade, the Pre- Upgrade Duplicate Detection form should be run on the Microsoft Dynamics AX 3.0 database before you run any upgrade scripts. The Pre-Upgrade Duplicate Detection form identifies any duplicate record IDs that exist in a Microsoft Dynamics AX 3.0 database.

Download the .xpo file from the location below, and import it into your Microsoft Dynamics AX 3.0 system. Run the form to identify duplicates. You must resolve these duplicates before running any of the upgrade scripts.

<http://go.microsoft.com/fwlink/?LinkId=137799>

Copy your existing application files

To retain your modifications to existing application objects, you must copy your application files to a directory where they can be compared with the updated versions.

1. Replicate your application from your production environment to a test environment.
2. Copy the application files (*.aod, *.ahd, *.ald, *.add, *.khd) from layers above the **DIS** layer (**BUS, VAR, CUS, USR**). Copy the application files from the production environment to the test environment, placing them in analogous folders.

Do not copy the following files:

- axSYS.*
- axSYP.*
- axGLS.*
- axGLP.*
- axDIS.*
- axDIP.*

 **Note:**

Not all of the above files may be present.

 **Note:**

Copying the wrong layers can halt the upgrade.

For example, in a Microsoft Dynamics AX 4.0 system, copy all the application files from the application folder of the production environment (**\\Microsoft Dynamics AX\40\Application\App\Standard**) to a folder of the instance name and location in the test environment (**\\Microsoft Dynamics AX\50\Application\App\<instance name>**).

In Microsoft Dynamics AX 2009 the names of some layers have been changed. The following table shows the old names and the corresponding new names. Be sure to rename the old layer files to the corresponding new names before you upgrade.

Old name	New name
DIS	HFX
DIP	SL1
LOS	SL2
LOP	SL3

The words **SL1**, **SL2**, and **SL3** are reserved words. If you have a label file that contains these words in its file name, you should rename the label file before upgrading.

In order to compare the prior version application code with your modifications to the current application code, you must create a folder named "Old" in the current application folder of your test environment. (For example: **Microsoft Dynamics AX\50\Application\Appl\<instance name>\Old**) Copy all previous version application files (*.aod, *.ahd, *.ald, *.add, *.khd) into the "Old" folder.

3. If you are using the Microsoft Dynamics AX Payroll module or any third-party applications, install the new versions of those solutions that are certified for Microsoft Dynamics AX 2009. For more information on upgrading the Payroll module, see [Upgrade notes affecting specific modules](#).

Prepare Axapta 3.0 data for upgrade to Dynamics AX 2009

To upgrade from Microsoft Dynamics AX 3.0 with a supported service pack to Microsoft Dynamics AX 2009 you must first prepare your data for use in the new version.

 **Note:**

We recommend that you read through the [Best practices for upgrading](#) topic before you complete these tasks.

This section contains information about the following topics.

- [Upgrade to 64-bit record ID fields](#)
- [Left-justify database columns](#)
- [Create a new database](#)
- [The Microsoft Dynamics AX DB Upgrade Preparation tool](#)

Upgrade to 64-bit record ID fields

Before a database created for Microsoft Dynamics AX 3.0 can be upgraded for use with Microsoft Dynamics AX 2009, all 32-bit (data type **int**) identification references must be converted to 64 bits (data type **int64**). This includes 32-bit record identification references and transaction identification references.

Use the file `PrivateProject_UpgradeColumnList.xpo` to create a list of columns containing 32-bit values requiring conversion. The Microsoft Dynamics AX DB Upgrade Preparation tool performs the actual conversion at a later time.

 **Note:**

We recommend that you back up your data before you perform this task.

1. Start Microsoft Dynamics AX 3.0.
2. On the Microsoft Dynamics AX 3.0 client toolbar, click the Projects icon to open the Projects form.
3. Click the Import icon. The Import form is displayed.
4. Enter the location of the `PrivateProject_UpgradeColumnList.xpo` file in the **File name** field. This file is located on the installation media in the `DatabaseUpgrade` folder.
5. Click **OK**. The project now is listed in the Private folder within the Projects form.
6. Expand the private folder.
7. Right-click on `UpgradeColumnList` and select **Open**.
8. The Project `UpgradeColumnList` form is displayed.
9. Right-click **UpgradeColumnList**, and then click **Open**.
10. On the **Columns to be upgraded** form, click **Generate**.

The script generates a list of the columns that contain record ID fields that need to be converted from 32-bit to 64-bit values.

Left-justify database columns

Database columns in Microsoft Dynamics AX 2009 are left-justified. Therefore, database columns in Microsoft Dynamics AX 3.0 that are not left-justified must be converted to left-justification. Use the file `LeftJustified.xpo` to perform this conversion.

**Note:**

We recommend that you back up your data before performing this task.

1. Start Microsoft Dynamics AX 3.0.
2. On the Microsoft Dynamics AX 3.0 client toolbar, click the Projects icon to open the **Projects** form.
3. Click the Import icon. The **Import** form is displayed.
4. Enter the location of the `PrivateProject_LeftJustified.xpo` file in the **File name** field. This file is located on the installation media in the `DatabaseUpgrade` folder.
5. Click **OK**. The project now is listed in the **Private** folder within the **Projects** form.
6. Expand the **Private** folder.
7. Right-click **LeftJustified** and select **Open**. The **Project LeftJustified** form is displayed.
8. Right-click **MainJob(usr)**, and then click **Open**.

A progress bar displays the process of scanning the columns. When the scan is complete, the **Left Justify** form opens.

9. Click **Left justify all**.

As an alternative, you can click **Display commands** for better performance. Refer to the Run the SQL commands in SQL Server procedure below for more information.

Run the SQL commands in SQL Server

If you are working with a large database, you can improve performance by clicking **Display commands** instead of **Left justify all** in step 9 of the procedure above.

If you clicked **Display commands** in step 9, continue with the following steps:

1. After you click **Display commands**, an InfoLog form is displayed with a list of SQL commands. Copy the commands to an SQL text file and set it aside until after you have run the Microsoft Dynamics AX DB Upgrade Preparation tool.
2. Open Microsoft SQL Server Management Studio.
3. Choose the database used by Microsoft Dynamics AX 2009.

4. Click **New Query**.
5. Load the SQL commands from your text file and run them.

 **Note:**

For faster performance you can divide the SQL commands among several text files and run the commands in parallel.

If you are using an Oracle database, use SQLPlus (or equivalent) to run the SQL commands.

Create a new database

Before you run the Microsoft Dynamics AX DB Upgrade Preparation tool, you must create an empty database. This will be your target database when you run the Microsoft Dynamics AX DB Upgrade Preparation tool. This database also will be the database that you upgrade to Microsoft Dynamics AX 2009.

1. Open the SQL Server Management Studio (**Start > Programs > Microsoft SQL Server 2005 > SQL Server Management Studio**)
2. Enter the information necessary to connect to your server and click **Connect**.
3. In the Object Explorer pane, right-click on the **Databases** node and choose **New Database**. The **New Database** window is displayed.
4. Enter a name for the database.
5. (Optional) To help data upgrade performance later on, on the **Options** page, set the **Recovery model** to **Simple**. See [Improving data upgrade performance](#) for more information.
6. Click **OK**.

The Microsoft Dynamics AX DB Upgrade Preparation tool

The Microsoft Dynamics AX DB Upgrade Preparation tool is a separate application that prepares a new database (target) that you created in the previous procedure for the automatic data upgrade that is performed by Microsoft Dynamics AX 2009. It is a best practice to install the Microsoft Dynamics AX DB Upgrade Preparation tool on the database server that contains Microsoft Dynamics AX 2009. The tool must be run after the database server has been installed but before any Microsoft Dynamics AX 2009 components have been started.

 **Note:**

The tool, once it has finished running, cannot be run again against the same target database.

Changes that the tool makes to the Microsoft Dynamics AX 3.0 database

Database preparation is necessary because fundamental and significant changes were implemented in the Microsoft Dynamics AX 4.0 database. These changes must be made in the Microsoft Dynamics AX 3.0 database to prepare it for the setup wizard in Microsoft Dynamics AX 2009.

- Unicode enabling
- Record identification extension
- Optimistic concurrency control (OCC) implementation.

The new database (target) database will be ready to upgrade to Microsoft Dynamics AX 2009 after you run the Microsoft Dynamics AX DB Upgrade Preparation tool.

In addition to those changes in the database, the record identification references and the transaction identification reference fields are converted from 32-bit to 64-bit data types (**int64** instead of **int**). For relevant tables, OCC is enabled.

Actions performed by the tool

The Microsoft Dynamics AX DB Upgrade Preparation tool performs the following actions:

- Creates the initial set of objects in the Microsoft Dynamics AX 2009 (target) database. These objects will reflect additional table and data type changes that are part of the upgrade.
- Exports data, using bulk copy export, from the Microsoft Dynamics AX 3.0 database (source) to a temporary location on a local hard drive.
- The Microsoft Dynamics AX DB Upgrade Preparation tool reads a Microsoft Dynamics AX 3.0 database (source) and populates a new database (target) with Unicode data types (**nchar**, **nvarchar**, and **ntext**) in place of the multi-byte character set (MBCS) data types.

- Imports data into the target Microsoft Dynamics AX 2009 database. During this import, light data transformation converts data for the new data types incorporated into the Dynamics 4.0 database schema. This import is performed by the Bulk Copy Export tool and is logged in the database.

 **Note:**

The Microsoft Dynamics AX DB Upgrade Preparation tool will fail if you use FRx. For more information, see [Troubleshooting the Microsoft Dynamics AX DB Upgrade Preparation tool](#).

Non-AOT-based tables

The Microsoft Dynamics AX DB Upgrade Preparation tool is designed to work only with the AOT-based tables that shipped with Microsoft Dynamics AX 3.0. Its behavior with tables added by ISVs or customers will be unpredictable. Non-AOT-based tables should be handled according to the following steps.

1. To identify non-AOT-based tables, run either DBPrep_nonAOTtable_SQL.sql (for a system based on Microsoft SQL Server) or DBPrep_nonAOTtable_Ora.sql (for a system based on Oracle Database Server) on your source database (Microsoft Dynamics AX 3.0).
2. Back up the tables identified by the scripts in step 1 and then drop them from the Microsoft Dynamics AX 3.0 database schema.
3. Run the Microsoft Dynamics AX DB Upgrade Preparation tool to copy the Microsoft Dynamics AX 3.0 database to the target database.
4. Restore the non-AOT-based tables to your Microsoft Dynamics AX 2009 database.

Optimizing tool performance

For optimization purposes, it is a best practice to run the Microsoft Dynamics AX DB Upgrade Preparation tool in parallel on a system that supports this. Running the tool in parallel requires that you start it from the command line with the command `Axdbupgrade.exe /pn`, where **n** is the number of parallel threads (minimum 1, maximum 64, default 8 threads).

Performance and capacity planning for the temporary folder must also allow for concurrent files to be written to this location. It is a best practice, if possible, to specify the temporary folder for the Microsoft Dynamics AX DB Upgrade Preparation tool on a volume that is separate from the volume containing the database files and has sufficient storage capacity.

To estimate the space requirements for the temporary folder, consider the size of the largest tables and the degree of parallelism expected. For example, if you plan to use a parallelism degree of four for the Microsoft Dynamics AX DB Upgrade Preparation tool, assume that your four largest tables will be exported or imported in parallel. The size of your temporary folder must equal the sum of the size of your four largest tables.

Run the Microsoft Dynamics DB Upgrade Preparation tool

When you run the Microsoft Dynamics AX DB Upgrade Preparation Tool, remember these two important points:

- The tool can be run only once against a specific target database.
- You must have read access to the source database and write access to the target database in order to use the Microsoft Dynamics AX DB Upgrade Preparation tool. The target database requires read/write access to the temporary folder.

Note:

We recommend that you back up your data after you run the Microsoft Dynamics AX DB Upgrade Preparation tool.

Running the tool on SQL Server databases

1. On the computer on which the Application Object Server (AOS) is being upgraded or installed, double-click `AxDBUpgrade.exe` on the installation media.
2. Mark the **Trusted Connection** check box.
Enter the SQL Server instance and database names for your source and target databases. The target database is the database that you created in the [Create a new database](#) procedure.
3. Enter the path to the log folder. The target server must have access to this directory.
4. Enter the path to the temporary folder. The target server must have access to this directory.
5. Click **Start**. You can stop the process at any time.

Running the tool on Oracle databases

1. On the computer on which the Application Object Server (AOS) is being upgraded or installed, double-click `AxDBUpgrade.exe` on the installation media.
2. Click the **Oracle** tab.
3. Enter the source and target database connection strings. The connection string should contain the server computer name and the Oracle TNSNAMES in the following format.
SERVER_NAME/TNSNAMES
4. Enter the source and target schema. The source schema is the Microsoft Axapta 3.0 schema and the target schema is the one you created during the Microsoft Dynamics AX 2009 installation and it must be empty.
5. Enter the path to the log folder. The target server must have access to this directory.
6. Enter the path to the temporary folder. The target server must have access to this directory.
7. Click **Start**. You can stop the process at any time.

Upgrade Team Server

If you use a version control system and you are upgrading from Microsoft Dynamics AX 4.0, you must upgrade Team Server before you upgrade to Microsoft Dynamics AX 2009. For more information, refer to your Microsoft Dynamics AX 2009 installation guide.

Upgrading to Microsoft Dynamics AX 2009

This section provides information about upgrading Microsoft Dynamics AX 3.0 with a supported service pack to Microsoft Dynamics AX 2009. While the Microsoft Dynamics AX application provides you with tools to complete your upgrade, careful planning ensures a successful upgrade, including minimal downtime to your Microsoft Dynamics AX system during data upgrade. This section contains the following topics.

- [Install Microsoft Dynamics AX before you upgrade](#)
- [The Upgrade checklist](#)
- [The Data Upgrade Cockpit](#)
- [Prepare to upgrade](#)
- [Upgrade your data](#)
- [Finalize upgrade](#)
- [Additional upgrade tasks](#)

Install Microsoft Dynamics AX before you upgrade

After you have completed the [Before you upgrade](#) steps, you are ready to install Microsoft Dynamics AX 2009. For more specific information about installing, refer to the Microsoft Dynamics AX 2009 Installation Guide.

 **Note:**

For the supported upgrade paths, see [Supported upgrade paths](#).

Install Microsoft Dynamics AX 2009

1. Run Setup.exe from the installation media. The Microsoft Dynamics AX Setup page is displayed.
2. Select the language in which you want to view Setup and click OK. The **Welcome to Microsoft Dynamics AX Setup** page is displayed.
3. Click **Next**. The **License terms** page appears.
4. Mark the **I accept the license terms** check box, and then click **Next**. The **Select installation type** page is displayed.
5. Select **Developer installation**, and then click **Next**. The **Previous version detected** page is displayed.
6. Complete the appropriate step below for your upgrade path.

Upgrade Path	Step
Using Microsoft SQL Server and upgrading from Microsoft Dynamics AX 3.0 to Microsoft Dynamics AX 2009	Select Connect to an existing SQL Server database , and then click Next .
Using Microsoft SQL Server and upgrading from Microsoft Dynamics AX 4.0 to Microsoft Dynamics AX 2009	Select Connect to an existing SQL Server database , and then click Next .
Using Oracle	Connect to an Oracle database server schema under which the Microsoft Dynamics AX 2009 objects will be created in the database.

7. If the **Install files page** is displayed, skip to step 9. Otherwise, if prerequisite software must be installed, the **Install prerequisites** page is displayed. Click **Install prerequisite software**.

 **Note:**

A new installation on Windows Server 2003 will require installation of the Microsoft Visual C++ Redistributable package and the Visual C++ 2008 Redistributable package.

8. After the prerequisite software has been installed, click **Next**. The **Install files page** is displayed.
9. Accept the default file location or enter a different location.
10. Click **Next**. One of the following pages is displayed depending on the selection that you made in step 6.

Page name	Steps
AOS: Connect to a SQL Server Database	<ol style="list-style-type: none"> 1. Enter your server name. 2. Enter the name of your Microsoft Dynamics AX 4.0 database or your target database that you used with the Microsoft Dynamics AX DB Upgrade Preparation tool. For more information, see The Microsoft Dynamics AX DB Upgrade Preparation tool.
AOS: Connect to an Oracle database server	<p>Select the connection type you plan to use.</p> <ul style="list-style-type: none"> • If you select Use a net service (TNS) to connect, type the service name. • If you select Use custom settings to connect, type in the host name, database service name, and TCP/IP port to connect to. • In the Schema (user) name box, specify a name for the schema under which the Microsoft Dynamics AX objects will be created in the database. You can accept the default value of DBO. If you enter a schema name that does not already exist, Setup creates it for you. In the Password box, specify a password for the schema.

11. Click **Next**. The **Applications files: Set an instance name** page is displayed.
12. Click **Next**. The **Application files: Select a country or region** page is displayed.

13. Click **Next**. The **AOS: Create an instance page** is displayed.
14. Click **Next**. The **AOS: Select an account page** is displayed.
15. Enter network service account or domain account information, and then click **Next**. The **Client: Select a display language page** is displayed.
16. Click **Next**. The **Client: Select Help languages page** is displayed.
17. Click **Next**. The **Ready to install page** is displayed.
18. Clear the **Start the AOS instance after installation is completed** check box.
19. Click **Install**. After installation, the **Setup was completed successfully** page is displayed.
20. Click **Finish**.

Install Microsoft Dynamics AX 2009 Service Pack 1

To upgrade to Microsoft Dynamics AX 2009 Service Pack 1, install that service pack at this time. To download and install the service pack, go to <http://go.microsoft.com/fwlink/?LinkId=130877>.

Next steps

After the installation has completed, follow the steps below to continue your upgrade.

1. If you haven't done so already, copy your existing application files. For more information, see [Copy your existing application files](#).
2. Start the Microsoft Dynamics AX 2009 AOS. From the Start menu, click **Run**, type `Services.msc`, and click **OK**. On the **Services window**, right-click **Dynamics AX Object Server 5.0\$[DATABASE NAME]** and choose **Start**.

 **Note:**

- The size and complexity of your database will affect how long it takes to start the AOS. It may take a long time to start.
3. Start the Microsoft Dynamics AX client. The Upgrade checklist is displayed. Complete the steps in the Upgrade checklist to continue upgrading. See [The Upgrade checklist](#) for more information.

The Upgrade checklist

The Upgrade checklist helps you to upgrade your Microsoft Dynamics AX system correctly and efficiently by guiding you through the steps of the upgrade process.

The Upgrade checklist displays the required and optional tasks involved in upgrading successfully. As each task is completed, the task is checked by the system. Most of these tasks are one-time setup options.

Icons are used to mark the status of the tasks involved. Some tasks are mandatory, and some tasks depend on other tasks being completed first. Tasks must be completed from the top down.

Important:

Do not run the upgrade checklist if you are upgrading from Microsoft Dynamics AX 3.0 or 4.0 with GLS layers for individual countries. In Microsoft Dynamics AX 2009, individual GLS layers are being consolidated. Before you can install a consolidated GLS layer, you must install Service Pack 1 for Microsoft Dynamics AX 2009. For information about upgrading the GLS layer for a specific country, see the [Service Pack 1 page](#) on CustomerSource.

Displaying the Upgrade checklist

The Upgrade checklist is displayed automatically when you start the Microsoft Dynamics AX 2009 application for the first time or if you've used a previous version of Microsoft Dynamics AX on your system but haven't completed all of the Upgrade checklist tasks. You also can access the Upgrade checklist by clicking **Administration > Setup > System > Checklists > Upgrade checklist**.

Prevent the Upgrade checklist from starting

(For developer use.) The Upgrade checklist will be displayed every time you start the Microsoft Dynamics AX 2009 client unless you've completed all of the steps in the checklist. To prevent the Upgrade checklist from starting every time you start the Microsoft Dynamics AX 2009 client, select the **Prevent startup of Upgrade Checklist** checkbox at the bottom of the Upgrade checklist. The **Prevent startup of the checklist** check box only is displayed if you've completed all of the required tasks in the Upgrade preparation section of the Upgrade Checklist.

The Data Upgrade Cockpit

Use the Data Upgrade Cockpit during the Presynchronize, Postsynchronize, and Additional upgrade stages.

 **Note:**

You must complete all the tasks in the Upgrade preparation section of the Upgrade checklist before you can work with the Data Upgrade Cockpit.

The Data Upgrade Cockpit creates an upgrade job that maintains the correct dependency relationships among data upgrade tasks and ensures that the tasks are started in the correct order. The Data Upgrade Cockpit also helps you to interact with the upgrade job and associated tasks, and to view the status of the upgrade job.

Just above the Data Upgrade Cockpit grid is a box that contains information about data logging.

 **Note:**

Because the data upgrade process is performance-intensive, it is a best practice to change the database logging to minimal during the upgrade. Be sure to return your database logging to its previous setting when you complete the upgrade process.

Each row in the Data Upgrade Cockpit grid represents an individual task in the upgrade job.

Display the Data Upgrade Cockpit

Click **Presynchronize**, **Postsynchronize**, or **Additional upgrade** in the Upgrade checklist to display the Data Upgrade Cockpit.

Start the upgrade job

Click **Run** to start an upgrade job and submit it to the batch server. Job tasks are sorted into the correct order for completion based on the upgrade script dependency tree.

Task status

When you run a job, the Data Upgrade Cockpit displays the status of each task in the job.

Status	Description
Ready	The upgrade script is ready to be executed.
Waiting	The upgrade script is waiting for another job to finish or it is waiting because the job was restarted.
Finished	The upgrade script completed successfully.
Error	An error occurred during execution of the upgrade script.
Executing	The upgrade script is being executed.

Pause the upgrade job

Click **Pause** to stop, but not cancel, the upgrade job. The tasks in the job that have started will not be cancelled but tasks that are in a ready state will be placed in a waiting state and no other tasks will be submitted to the batch until you run the job again.

Rerun scripts

Select a task and click **Rerun script** to resubmit the task to the batch, and then click **Run** to start the task. Perform this action when you want to restart one failed script at a time.

Note:

A task that has run successfully will not be rerun. The upgrade framework will not start a task if an entry for it is present in the syssetupcompanylog table.

Rerun all failed scripts

If you encounter multiple upgrade-task failures, click **Rerun all failed scripts**.

Reload scripts

Click **Reload scripts** to recreate the data upgrade job from scratch. This action will recreate the job by re-traversing the dependency tree for all upgrade scripts, and recreating the upgrade job constraints based on dependency. This option can be useful after you have modified one or more dependencies within the upgrade scripts and you want to rerun the data upgrade job based on a new dependency tree.

View job errors

When the status of a job is **Error**, you can detect the source of the error by doing the following:

1. Select the task with the error, and click **View job errors**.
An Infolog box that indicates the source of the error is displayed.
2. After you investigate the source of the error and fix it, click **Rerun task** to run the task again.

Note:

If you have a developer license, you can double-click on a task to access the script for that task and edit it. For information about creating upgrade scripts, see the [How to Write Data Upgrade Scripts white paper](#) (<http://go.microsoft.com/fwlink/?LinkId=115169&clcid=0x409>). For more information about troubleshooting, see [Troubleshooting the Data upgrade cockpit](#).

View job predecessors

Tasks that have a status of **Waiting** are dependent on other tasks that have not completed yet. You can see which tasks any one task is dependent on by doing the following:

1. Select a task that has a status of **Waiting**, and click **View script predecessors**.
2. A new dialog box is displayed which lists the tasks that must be completed before the selected task can be started.

View batch task

Select a batch task and click **View batch task** to display details about the task.

View batch job

Select **View batch job** to view details about the batch job. A form is displayed showing how all of the batch tasks are related to each other.

Cancel

Click **Cancel** to prevent the start of any upgrade jobs that have a status of waiting. Jobs that have started before you click **Cancel** will continue until they are completed.

Note:

Perform this action with caution. After you click **Cancel**, the database might not be in a consistent state. It is recommended that you work from a database backup when you resume the upgrade.

Trial run

Click **Trial run** to scan through the upgrade tasks and identify possible data upgrade issues. The data upgrade doesn't stop on failures as it would during the **Run** operation. The main purpose of this action is to estimate the effort that it will take to fix any data upgrade issues that you might encounter during the data upgrade process.

During a trial run, false failures can occur in child tasks where the parent upgrade task fails for a legitimate reason. The failure occurs because the data dependency requirement is not met. You would not need to repair the task in this case.

Prepare to upgrade

This section contains information about the tasks that you must complete before you upgrade your data. See the following topics for more information.

- [Compile the application \(Upgrade\)](#)
- [Provide license information](#)
- [Time zones and upgrade](#)
- [Set time zone for upgraded data](#)
- [Set customer feedback options](#)
- [Detect customization conflicts](#)
- [The Compare Tool](#)
- [Test the system after code upgrade](#)
- [Create virtual companies for global address books](#)

Compile the application (Upgrade)

Compile the application to ensure that all object references are updated and the application is ready to use. This process must be completed.

 **Note:**

To run the Compile application task, the .NET Business Connector must be installed on the computer where you are running the checklist.

The application must also be compiled during installation of a service pack because the object references from the standard application will require updating.

When the application is compiled, its source code files are translated into binary object files that can be interpreted by the Microsoft Dynamics AX kernel. These object files have the .aod extension.

 **Note:**

Depending on your hardware, compilation can take an hour or more. It is critical that compilation be allowed to run until it is finished.

Provide license information

To be able to use Microsoft Dynamics AX, the administrator must enter license information. By entering license codes, you enable the general functionality covered by the license. Then, you can enable or disable access to more specific features by changing configuration keys.

 **Note:**

If you change your current license settings because license keys are updated, the new functionality will not be available until the client is restarted.

In addition to license codes for Microsoft Dynamics AX functionality, there are four licenses for access to the Microsoft Dynamics AX development environment. For more information, see the Developer Help, available from the Microsoft Dynamics AX Help menu.

If you do not have a license, you can set up Microsoft Dynamics AX in demonstration mode. Demonstration mode provides all the functionality of Microsoft Dynamics AX, and enables all configuration keys by default.

Language-specific licenses

If your license includes specific languages, you must restart the AOS after importing the license file or entering license information. Restarting the AOS ensures that the correct languages are listed in the **Options** form (**Microsoft Dynamics AX > Tools > Options...**).

If you do not restart the AOS, end users will be able to select unlicensed languages, which would prevent the Microsoft Dynamics AX client from starting.

Import license information

1. Open the **License information** form from the initialization checklist or the upgrade checklist, or click **Administration > Setup > System > License information**.

2. Click **Load license file** to import the license codes from a file.

The **Load license file** dialog box appears.

3. Click the folder icon and browse for your license file.

 **Note:**

We recommend that you store your license file in a secure location that is known only to Microsoft Dynamics AX administrators.

4. Click **OK**. A message appears, asking whether you want to synchronize the database.
5. Click **Yes**.
6. Close the **License information** window.

Enter license information

As an alternative to importing the license information, you can enter the license information manually.

1. Open the **License information** form from the initialization checklist or the upgrade checklist, or click **Administration > Setup > System > License information**.
2. Enter the name of the license holder, the system's serial number, and the expiration date. The information is in your license document.
3. On the **System** tab, enter the license code and verify that the **Status** field displays the expected text.

The license code indicates whether you have a standard, professional, or enterprise solution.

 **Important:**

The first four entries (name of the license holder, the system's serial number, expiration date, and license code) determine what appears in the remaining codes.

Therefore, they must be entered correctly.

4. Enter the remaining codes.

For each, review the **Status** field to make sure that the code is accepted.

Time zones and upgrade

The **utcDateTime** data type is the preferred method of storing date and time data in Microsoft Dynamics AX 2009. The data upgrade automatically updates all system date and time fields to use the **utcDateTime** data type. During data upgrade, all tables are scanned to find the system fields in each table, whether the table ships with Microsoft Dynamics AX 2009 or has been added by a partner.

In order for your data to be upgraded correctly, you must provide the time zone you want to use through the Upgrade checklist. This time zone is used to determine which offsets to apply to the date/time fields when they are converted into Coordinated Universal Time (UTC) values.

 **Note:**

If custom **Date** or **Time** fields have been added to Microsoft Dynamics AX, you need to decide whether those fields should be merged into new **UtcDateTime** fields.

UtcDateTime fields store date/time data in UTC. See [Walkthrough: upgrading date and time table field pairs into UtcDateTime](#) for more information.

For more information about date/time data and time zones, see the **Applications and Business Processes** help in Microsoft Dynamics AX.

Set time zone for upgraded data

During the data upgrade, all system date and time fields are updated to use the **datetime** data type, and are converted into Coordinated Universal Time values (UTC). All tables are scanned for system date and time fields, including those added by a partner.

You can also upgrade dates and times that are not system fields.

Set time zone for upgraded data

1. On the Upgrade Checklist, click **Set current time zone**.
The **Current time zone** form appears.
2. In the **Time zone** list, select the time zone to use. The time zone is used to convert datetime fields into UTC values.

Set customer feedback options

Join the Customer Experience Improvement Program to help improve the quality, reliability, and performance of Microsoft software and services.

Open the **Microsoft Dynamics Ax Customer Feedback Options** form from the initialization checklist or the upgrade checklist, or click **Administration > Setup > System > Customer feedback options**.

The program collects information about computer hardware and how you use Microsoft Dynamics AX, without interrupting you. This helps Microsoft identify which Microsoft Dynamics AX features to improve. No information collected is used to identify or contact you.

For more information and a complete privacy statement for the Customer Experience Improvement Program, visit the [Customer Experience Improvement Program Web site](#).

Detect customization conflicts

The code upgrade part of the upgrade process consists of examining the code in your Microsoft Dynamics AX 3.0 or 4.0 installation and comparing it with the Microsoft Dynamics AX 2009 code. An upgrade conflict exists when code in a previously installed version differs from code in the new version. The following scenarios can result in code conflicts.

- You developed functionality in your previous installation that now exists in Microsoft Dynamics AX 2009.
- You have deprecated or renamed objects that existed in the previous installation, creating incompatibilities with Microsoft Dynamics AX 2009.
- You have modified existing objects in your previous installation, which means that you will need to move or update those modifications (fields in tables, class modifications, form/report modifications) to work with Microsoft Dynamics AX 2009.

Several tools are available to help you detect upgrade conflicts that are caused by customizations that you've made to your Microsoft Dynamics AX system.

Detect code upgrade conflicts

Use the **Detect code upgrade conflicts** tool to analyze your customizations for code upgrade conflicts and create upgrade projects.

1. From the **Upgrade checklist**, click **Detect code upgrade conflicts**. The **Detect code upgrade conflicts** form is displayed.
2. Click **OK**. One or more upgrade projects will be created. The following table describes some types of upgrade projects.

Project name ends with...	Description
LayerConflicts	This type of project compares layers and lists the objects that differ.
ReclDConflicts	This type of project contains elements that violate certain best practices. For information on how to solve these errors, see the Writing Secure X++ Code white paper.
TwCConflicts	This type of project contains elements that violate certain best practices. For information on how to solve these errors, see the Microsoft Dynamics AX documentation on MSDN .

Create an Estimation report

Now that an upgrade project has been created, you can use the **Estimation report** tool to help you determine the amount of time it will take you to upgrade your customizations.

1. With the main Microsoft Dynamics AX form displayed, type Ctrl+Shift+P to display the **Projects** form.
2. Right-click on an upgrade project that was created in the **Detect code upgrade conflicts** procedure above and select **Open**. The Upgrade project form is displayed.
3. Right-click on the upgrade project and select **Parameters**. The **Upgrade parameters** form is displayed with default time estimates already entered for you.
4. To create a new entry in the **Overridden estimates** grid, click in the grid and type Ctrl+N.
5. Select an **Application object type** and enter a time estimate for how long that type of conflict will take you to fix.
6. Click the **Additional estimates** tab to review and modify the default estimates.
7. Close the **Upgrade parameters** form when you have finished entering your time estimates.

8. On the **Upgrade project** form, right-click the upgrade project and select **Estimation report**. The **Upgrade estimate** form is displayed. You can modify the report settings before you generate a report.
9. Click **OK** to create the report.

Compare layers tool

As an alternative to the **Detect code upgrade conflicts** tool, you can use the **Compare layers** tool (from the Microsoft Dynamics AX menu, select **Tools > Development tools > Code upgrade > Compare layers**) to compare any two layers and create a project with the objects that differ.

The **Compare layers** tool can provide an overview of modifications made in a certain layer. For more information, see [The Compare Tool](#).

 **Note:**

If you have modifications in more than one layer, upgrade one layer at a time. The lowest layer that you have access to should be upgraded first. For example, if you have a VAR layer and a USR layer, upgrade the VAR layer first and then the USR layer.

Project filter tool

If you have developer rights, you can use the **Project filter** tool in the upgrade process as an alternative to either the **Detect code upgrade conflicts** tool or **Compare layers** tools. **Project filter** lets you create a project based on a criterion supplied in query form. For example, the criterion might be all objects from a relevant layer, all objects with a specific prefix, or all objects created by a specific user. Refer to the **Developer Help** documentation under the Microsoft Dynamics AX Help menu for more information about the Project filter tool.

The Compare tool

In Microsoft Dynamics AX, you can compare application objects to find differences by using the Compare tool. After you identify the differences, you can insert, delete, or move the differences between the compared application objects. This topic describes how to interpret the results of the comparison.

The Compare tool enables you to compare:

- Two different application objects.
- The same application object in two different layers.
- Two versions of the same application object, if you are using version control.
- Two versions of the same application object and the original version of the application object.

For more information about the Compare tool, see the [Compare Tool topic](#) in the Microsoft Dynamics AX Developers documentation.

How to: Compare Two Application Objects

In Microsoft Dynamics AX, you can compare two application objects in the same or different layers by using the **Compare** tool. For example, you can compare a table in the CUS layer with the same table in the SYS layer. The **Compare** tool identifies the differences between the two application objects and provides action buttons to copy, move, or delete the differences. This topic describes how to compare two application objects in the Application Object Tree (AOT).

To compare two application objects

1. In the AOT, select the application objects that you want to compare.
2. Right-click the selected application objects, and then click **Compare**. The **Comparison** form opens.
3. Click the **Advanced** tab, and then set the following options as needed.

Option	Description
Show differences only	Select this option to view only the differences. When this option is cleared, matching results are displayed as black check marks.
Suppress whitespace	Select this option to show differences in spaces in X++ code.
Case sensitive	Select this option to view upper and lowercase differences. By using this option, MyName and Myname would be displayed as a difference.
Show line numbers	Select this option to enable line numbers in X++ code.

4. Click **Compare** to begin the comparison of the two application objects.

The differences are displayed in the results window on the **Comparison** form. For more information about the results, see Compare Tool.

How to: Compare Versions of an Object

In Microsoft Dynamics AX, you can compare two versions of the same application object by using the **Compare** tool. For example, you can compare a table with an earlier version of the same table provided that version control is enabled. Each version of the application object stored by the version control system is numbered. The **Compare** tool identifies the differences between two versions and provides action buttons to copy, move, or delete the differences. This topic describes how to compare two different versions of the same application object.

 **Note:**

You can also open the **Compare** tool form on the **History** form.

To compare two versions of an application object

1. In the Application Object Tree, right-click the application object that you want to compare, and then click **Compare**. The **Comparison** form opens.
2. On the **Name & Location** tab, select the two versions that you want to compare.
3. Click the **Advanced** tab, and then set the following options as needed.

Option	Value
Show differences only	Select this option to view only the differences. When this option is cleared, matching results are displayed as black check marks.
Suppress whitespace	Select this option to view differences in spaces in X++ code.
Case sensitive	Select this option to view upper and lowercase differences. By using this option, MyName and Myname would be displayed as a difference.
Show line numbers	Select this option to enable line numbers in X++ code.

4. Click **Compare** to begin the version comparison.

The differences between the versions are displayed in the results window on the **Comparison** form. For more information about the results, see the [Compare Tool topic](#) in the Microsoft Dynamics AX Developers documentation.

How to: Compare Application Object Layers

This topic describes how to use the **Compare layers** tool in Microsoft Dynamics AX. You can use the **Compare layers** tool to compare any two application object layers, for example, the SYP layer and the SYS layer.

The **Compare layers** tool compares a source layer to a reference layer. The compare process searches for application objects that have changed or exist only in the reference layer.

If VAR is the source layer that you want to compare to the standard application that has service packs installed, select SYP as the reference layer. The compare layers process will then compare the VAR layer with the SYP and SYS layers. The result of the comparison is a project that holds objects that do not occur in the reference layer, or occur in both layers, but are different provided that the option to **Only include elements present in both layers** is not selected.

When the reference layer is not an **old** layer, such as **old sys**, the reference layer automatically includes all lower layers. For example, if the USR layer is the source layer and it is compared to the CUS layers as a reference layer, then the reference layer includes the CUS layer and all layers underneath it.

1. On the **Microsoft Dynamics AX** menu, point to **Tools**, point to **Development tools**, point to **Code Upgrade**, and then click **Compare layers**. The **Compare layers** tool opens.
2. In the **Compare layers** tool, set the following properties as required.

Option	Description
Project name	The name of the project to contain the results of the compare layers tool process.
Source layer	The layer used as the base layer for the comparison. The compare tool results only display differences for application objects that are contained in the source layer.
Reference layer	The layer that you want to compare to the source layer.  Note: You can select old sys as the reference layer. This is useful if you want to learn what application objects have changed.
Only include elements present in both layers	Select to detect conflicts between layers. When selected, elements found in only one of the layers are ignored.

3. Click **OK** to begin the application object layer comparison.
4. A project holding application objects that differ is now created. The objects are presented in a tree structure and the regular Application Object Tree (AOT) shortcut menu commands are available.

 **Note:**

In a compare situation, it is useful to see the layer to which each object belongs. For more information, see the How to: Set the Layer View for Application Objects topic in your Microsoft Dynamics AX for Developers documentation.

Test the system after code upgrade

After upgrading your code and before proceeding to upgrade your data, test whether your installation is functioning properly.

To test whether the code upgrade to the Microsoft Dynamics AX system has succeeded, at a minimum you should perform these steps:

- Restart all Application Object Server (AOS) instances.
- Launch a client that is connected to each running AOS instance.
- Test that users can execute the most common tasks for your system.
- Test that the administrator can run the system maintenance tasks.
- Test that users can execute the most vital tasks for your system.
- Test that the permission settings are working right by checking that users have access to the data they had access to prior to upgrade.

Create virtual companies for global address books

You can share address book information between companies by using the global address book feature. For each group of companies that will share an address book, you must create a virtual company.

Important:

You must set up virtual companies for address books before you create customer, vendor, employee, competitor, or contact records for a company.

For more information about company accounts, virtual company accounts, and global address books, see the **System and Application Setup** Help, available on the Help menu in Microsoft Dynamics AX.

Example

Fabrikam Inc. has five companies set up in Microsoft Dynamics AX:

- Company 1 and Company 2 are in related manufacturing and distribution businesses.
- Company 3 and Company 4 are in the professional services industry.
- Company 5 is in the retail industry.

Fabrikam Inc. requires that related businesses share address book information. Therefore, the administrator sets up the following virtual companies:

Virtual company	Companies it contains
Virtual Company A	Company 1 Company 2
Virtual Company B	Company 3 Company 4
N/A	Company 5

Set up virtual companies for address books

1. Open the **Virtual company accounts** form from the initialization checklist or the upgrade checklist, or click **Administration > Setup > Virtual company accounts**.
2. Press CTRL+N to create a new record.
3. In the **Company accounts** field, enter the company identification.

4. In the **Name of company accounts** field, enter the name of the virtual company.
5. Click the **Company accounts** tab.
6. Select the company accounts to participate in the virtual company.
 - To add a company account, select the company name under **Remaining company accounts** and then click the left arrow (<) to move it to the **Selected company accounts** list.
 - To remove a company account, select the company name under **Selected company accounts** and then click right arrow (>) to move it to the **Remaining company accounts**.
7. Click the **Table collections** tab.
8. Select the **DirPartCollection** table and then click the left arrow (<) to move it to the **Selected table collections** list.
9. If you opened the **Virtual company accounts** form from the initialization checklist or upgrade checklist, you can check for address book errors related to virtual company setup by clicking **Validate address books**.

Upgrade your data

As part of the upgrade process to a new version or to a service pack, data must be upgraded. This is necessary when changes have been made to the existing data dictionary, where tables, fields, or indexes have been renamed, deleted or added.

This section contains information about improving data upgrade performance and about the data upgrade tasks listed in the Upgrade checklist. See the following topics for more information.

- [Improving data upgrade performance](#)
- [Presynchronize your database](#)
- [Synchronize your database](#)
- [Postsynchronize your database](#)
- [Test the system after data upgrade](#)

Improving data upgrade performance

Read through the Optimizing Data Upgrade whitepaper (<http://go.microsoft.com/fwlink/?LinkId=119309>) to help improve the performance of your system during the data upgrade process.

Presynchronize your database

Use the Data Upgrade Cockpit to prepare your database for synchronization. For more information, see [The Data Upgrade Cockpit](#).

This process prepares earlier versions of the Microsoft Dynamics AX database schema for synchronization.

For example, the presynchronization process deletes duplicates and disables unique indexes such as `ReleaseUpdateDB_V30toV40::allowDupPurchParmTable`.

The presynchronization process also prepares any metadata changes with an "undo" in the postsynchronization phase.

 **Note:**

You must complete all preceding tasks in the Upgrade checklist to work with the Data Upgrade Cockpit.

1. In the Upgrade checklist, click **Presynchronize**. The **Data Upgrade Cockpit** form is displayed and lists the upgrade tasks to complete.
2. Click **Run**. The upgrade job starts and the icon next to each task in the job displays the status of that task.
3. Address any errors that occur and close the Data Upgrade Cockpit when you are finished. For more information about the actions that are available to you, see [The Data Upgrade Cockpit](#) for more information.

Synchronize your database

Synchronize your application files with their corresponding current versions on the database server. When you perform these steps, you are synchronizing tables and indexes with changes that have been made in the Application Object Tree (AOT). Synchronization is mandatory when you upgrade and when view and schema changes in the AOT are populated to the database.

1. In the Upgrade checklist, click **Synchronize database**. The **Synchronize database** form is displayed.

The **Overview** tab displays a summary of errors, warnings, and Infolog messages that the synchronization process created.

2. Click the **Errors**, **Warnings**, or **Info** tab to display a list of tables that caused messages and to view the text of the messages.
3. Click **Continue**.

Postsynchronize your database

Use the Data Upgrade Cockpit tool to complete the Postsynchronize process. For more information about the Data Upgrade Cockpit, see [The Data Upgrade Cockpit](#).

Most of the data upgrade occurs during postsynchronization. This process also reverses the metadata changes that occurred during presynchronization.

 **Note:**

You must complete all preceding tasks in the Upgrade checklist before you can begin postsynchronization.

1. In the Upgrade checklist, click **Postsynchronize**. The Data Upgrade Cockpit is displayed.
2. Click **Run** to submit the postsynchronization tasks to the upgrade job batch server for execution.

The status of a task is either **Ready**, **Waiting**, **Executing**, **Finished** or **Error**. For more information about task status, see [The Data Upgrade Cockpit](#).

After you complete postsynchronization, you are ready to begin the first task in the Finalize upgrade section of the Upgrade checklist.

Test the system after data upgrade

After upgrading your data, test whether your Microsoft Dynamics AX 2009 installation is functioning properly before proceeding to upgrade in the production environment. This is a good time to test any documentation and training you have created.

To test whether the data upgrade has succeeded, at a minimum you should perform these steps:

- Restart all Application Object Server (AOS) instances.
- Launch a client that is connected to each AOS instance that is running.
- Test that users can execute the most common tasks for your system.
- Test that the administrator can run the system maintenance tasks.
- Test that users can execute the most vital tasks for your system.
- Test that users can read the reports that are generated.
- Test that the permission settings are working right by checking that users have access to the data they had access to prior to upgrade.
- Validate that your data is consistent with that prior to upgrade.

Finalize upgrade

This section contains information about the tasks that you must complete after you upgrade your data. See the following topics for more information.

 **Note:**

The [Upgrading AIF](#) task is not applicable to systems upgrading from Microsoft Dynamics AX 3.0 with any supported service pack because AIF was not present in the Microsoft Dynamics AX 3.0 release.

- [Configure system accounts](#)
- [Upgrade Enterprise Portal](#)
- [Specify Role Center Web site](#)
- [Upgrading AIF](#)

Configure system accounts

Configure the accounts that the system uses. This includes the Business Connector proxy account, the Workflow system account, the Workflow execution account, the synchronization service account, and the Virtual Earth account.

Open the **System service accounts** form from the initialization checklist or the upgrade checklist, or click **Administration > Setup > Security > System service accounts**.

Configure the Business Connector proxy account

In the **Alias** and **Network domain** fields, enter the user name and domain for the Business Connector proxy account.

The Business Connector proxy account is used for communication between Microsoft Dynamics AX and applications such as Role Centers and Enterprise Portal Framework, Workflow, the synchronization service, and AIF Web services. You must configure the Business Connector proxy account before you can install these applications.

For more information about the requirements for the Business Connector proxy account, see the [Server and Database Administration Guide](#).

Configure the Workflow system account

The Workflow system account is used for communication between Microsoft Dynamics AX and the Workflow Web server. You can use one of the following accounts.

- Enter a domain account in the **Alias** and **Network domain** fields to use a domain user for the Workflow system account.
- Enter a Microsoft Dynamics AX user for the Workflow service to communicate with. You can associate the service with a new or existing Microsoft Dynamics AX user.

For more information about the requirements for the Workflow system account, see the [Microsoft Dynamics AX Installation Guide](#).

Configure the Workflow execution account

The Workflow execution account is used for running application business logic and accessing Microsoft Dynamics AX data. You can use one of the following accounts.

- Enter a domain account in the **Alias** and **Network domain** fields to use a domain user for the Workflow execution account.
- Enter a new or existing Microsoft Dynamics AX user to access the database.

For more information about the requirements for the Workflow execution account, see the [Microsoft Dynamics AX Installation Guide](#).

Configure the synchronization service account

The synchronization service account is used for communication between Microsoft Dynamics AX, Windows Message Queuing, and Microsoft Office Project Server 2007. You can use one of the following accounts.

- Enter a domain account in the **Alias** and **Network domain** fields to use a domain user for the synchronization service account.
- Enter a Microsoft Dynamics AX user for the synchronization service to communicate with. You can associate the service with a new or existing Microsoft Dynamics AX user.

For more information about the requirements for the synchronization service account, see the [Microsoft Dynamics AX Installation Guide](#).

Configure the Virtual Earth account

The Virtual Earth account is used to access the online Virtual Earth map when working in Enterprise Portal.

You can find the Virtual Earth account name and password on [CustomerSource](#).

By selecting the check box and entering the information, you agree to be bound by the Microsoft Virtual Earth Map Control and MapPoint Web Service [End User Terms of Use](#).

Upgrade Enterprise Portal

You must verify all prerequisites and complete the procedures in this topic to upgrade Enterprise Portal. Before you upgrade Enterprise Portal, read the "Enterprise Portal and Role Centers" section of the [Microsoft Dynamics AX 2009 Installation Guide](#) to understand the installation process.

The Microsoft Dynamics AX 2009 installation process has changed since earlier versions. We recommend that you read the "What's new in Enterprise Portal" topic in the **Enterprise Portal Administration** help, which is available from the **Help** menu in the Microsoft Dynamics AX 2009 Windows client.

Before you upgrade, note the following:

- If you created new Enterprise Portal pages for Microsoft Dynamics AX 4.0 SP1, those pages will continue to function in the Microsoft Dynamics AX 2009 version of Enterprise Portal as they did in 4.01.
- Enterprise Portal data is upgraded when you run the Microsoft Dynamics AX Upgrade checklist.

Prerequisites

- If you are running Enterprise Portal on Microsoft Dynamics AX 4.01, you can upgrade directly to the version in Microsoft Dynamics AX 2009. If you are currently running Enterprise Portal on Microsoft Dynamics AX 4.0, you must upgrade to Microsoft Dynamics AX 4.01 before you can upgrade to the latest version of Enterprise Portal.
- The latest version of Enterprise Portal requires either Microsoft Windows SharePoint Services 3.0 with Service Pack 1 (or later) or Microsoft Office SharePoint Server 2007 with Service Pack 1 (or later). You must upgrade earlier versions of SharePoint to one of these versions before you can upgrade to the latest version of Enterprise Portal.
- You must complete the Upgrade checklist for Microsoft Dynamics AX 2009 before you upgrade Enterprise Portal. This means you must select the **Finalize Enterprise Portal upgrade** option in the Upgrade checklist before you start the procedures in this topic.

Upgrading Enterprise Portal

1. Start Microsoft Dynamics AX 2009 Setup.
2. When prompted, select **Add or modify components**, and then click **Next**.
3. On the **Add or modify components** page, select **Role Centers and Enterprise Portal**. Click **Next**.

4. If Setup does not detect all the prerequisite software, the **Install prerequisites** page appears. If Setup prompts you to install ASP.NET 2.0 or Windows SharePoint Services 3.0 with Service Pack 1, then click the **Install prerequisite software** button. When Setup verifies that all the prerequisite software is installed on the server, click **Next**.
5. On the **.NET Business Connector proxy account information** page, enter the domain\username and password for the .NET Business Connector proxy account. Click **Next**.
6. On the **Role Centers and Enterprise Portal framework: Configure IIS** page, use the **Web site** drop-down list to select the Microsoft Dynamics AX 4.01 Enterprise Portal Web site.
7. Select the **Configure for Windows SharePoint Services** option so that Setup can configure the application pool to run under the Business Connector proxy account and set the authentication method to Windows NTLM.
8. Clear the **Create Web site** option. If you select this option, Setup will upgrade your old Enterprise Portal site, but Setup will also create a new Enterprise Portal site.
9. On the **Ready to install** page, click **Install**. Setup might take as long as one hour to complete the upgrade and installation.

If appropriate, select the option to restart IIS when the installation has completed. Before selecting this option, consider the impact of an IIS reset on other applications on the Web server.

After you upgrade Enterprise Portal

After the installation and upgrade completes, do the following:

- If you upgraded an Enterprise Portal site that uses anonymous authentication (a public Enterprise Portal site) then you must select anonymous authentication in the IIS Manager after upgrade completes:
 - a. Click **Start > Administrative Tools > Internet Information Services (IIS) Manager**.
 - b. Expand the **Web sites** directory.
 - c. Right-click the Web site you upgraded and click **Properties**.
 - d. Click the **Directory Security** tab.
 - e. In the **Authentication and access control** section, click **Edit**.
 - f. Click **Enable anonymous access**.
- Verify permissions: The upgrade process should retain all earlier SharePoint and Microsoft Dynamics AX user and user-group permissions. You should, however, verify those permissions.
- Test the upgraded site on a staging server before moving the site into a production environment.

Specify Role Center Web site

Microsoft Dynamics AX and the Enterprise Portal framework include customizable home pages called Role Centers. Role Centers display specific data, reports, alerts, and common tasks associated with a user's role in the organization. Users can access Role Centers from the Microsoft Dynamics AX client or from an Enterprise Portal Web site.

Use this information to specify which Web site will host the Role Centers.

1. Click the option to **Specify Role Center Web site** in the upgrade checklist. The **Administration of Web sites** form opens.
2. Locate the **Web site used to display Role Centers in the Dynamics AX client** field at the bottom of this form.
3. Select a site using the lookup icon to the right of the field and then close the form. If you are not certain which site should host Role Centers, skip this step. You can select the site later using this form.
4. If no Web sites are listed in this form, Enterprise Portal has not been installed or an Enterprise Portal Web site has not been created. To learn more, see "Install Enterprise Portal and Role Centers" in the [Microsoft Dynamics AX 5.0 Installation Guide](#).

Upgrading AIF

Application Integration Framework (AIF) is the framework in Microsoft Dynamics AX that is used to exchange data with external systems. Upgrading AIF is one of the steps in the upgrade checklist.

When you upgrade AIF, there are two components to the upgrade process:

- **Code upgrade** – This process upgrades all AIF code including creation of new service classes, creation of new data classes, and creation of AOT service nodes. The code upgrade process is completed as part of the **Upgrade Aif Code** step in the upgrade checklist.
- **Data upgrade** – This process upgrades all AIF related records in the database. The data upgrade process is completed as part of the **Upgrade additional features** step in the upgrade checklist.

In Microsoft Dynamics AX 2009, there are schema changes that will impact you if you have document exchanges configured in AIF. In Microsoft Dynamics AX 2009, the document and message schemas have changed. This means that inbound messages will need to be formatted differently and validate against the new schemas. In addition, the outbound messages will be formatted differently and validate against the news schemas. You may also need to modify endpoint data policies, value mapping and XSLTs.

See [AIF upgrade process](#), for more detailed information.

Additional upgrade tasks

While you upgraded the core Microsoft Dynamics AX system in previous steps, not all features are completely functional as there may be additional features that you need to upgrade.

Upgrade additional features

Until you complete the Upgrade additional features task in the Upgrade checklist.

1. In the Upgrade checklist, click **Upgrade additional features**. The Data Upgrade Cockpit is displayed. For more information see [The Data Upgrade Cockpit](#).
2. Click **Run**.
3. After the upgrade job has completed, close the Data Upgrade Cockpit.

Add additional documentation help files

You can use the Help kit to create additional help files to use with the Microsoft Dynamics AX 2009 application. If, in a previous release, you've used the Help kit to add additional documentation files to Microsoft Dynamics AX, you should review that documentation to be sure that it is still accurate for Microsoft Dynamics AX 2009.

The Help kit contains instructions for using the kit.

[Download Help kit](#)

Upgrade SQL Server Reporting Services reports

After you upgrade, complete the following procedures to upgrade your SQL Server Reporting Services reports.

Set the Usage property for perspectives

Complete the following procedure to set the **Usage** property for perspectives.

1. Open the Microsoft Dynamics AX client.
2. Click the **Application Object Tree** icon on the tool bar.
3. In the **Application Object Tree**, expand the **Data Dictionary > Perspectives** node.
4. Right-click a perspective and click **Properties**. The **Perspective <PerspectiveName>** pane is displayed.
5. Select the **Usage** row.

6. From the **Usage** list, select one of the following:
 - **AdHocReporting** – Select this option if you want to generate a report model based on the selected perspective. For more information about generating report models, see the “Generate and publish report models” topic in the **System and Application Setup** help file. To access the **System and Application Setup** help file, click **Microsoft Dynamics AX > Help > System and Application Setup**.
 - **OLAP** – Select this option if you want to generate a business intelligence project based on the selected perspective. For more information about generating business intelligence projects, see the “Working with Business Intelligence Projects” in the **Developing for Microsoft Dynamics AX** help.
 - **Both** – Select this option if you want to generate a report model and a business intelligence project based on the selected perspective.
7. Repeat steps 4-6 for each perspective.

Regenerate your report models

Regenerate your report models by following the instructions in the “Generate and publish report models” topic in the **System and Application Setup** help file.

Generate your reports

If you have saved ad hoc reports, generate or run the reports. You can generate the reports by accessing them in Report Manager or Report Builder.

If you receive errors when generating the reports, you will need to modify the reports. Modify the reports with the tool that you used to create them. For example, if you created a report using Report Builder, use Report Builder to modify the report. Modify the reports to remove columns and fields that no longer exist in the Microsoft Dynamics AX 2009 data model.

Application upgrade notes

The following sections contain information about modules that are affected during the upgrade process.

- [Upgrade notes affecting all modules](#)
- [Upgrade notes affecting specific modules](#)

Upgrade notes affecting all modules

The issues in the following functional areas are general to Microsoft Dynamics AX 2009:

- Record-level security
- Alert rules
- Installation
- Rapid Configuration Tool (RCT)
- Task Recorder

Record-level security

Issue: Differences in record-level security between Microsoft Dynamics AX 2009 and previous versions

Versions affected

Microsoft Dynamics AX 4.0 with Service Pack 2

Upgrade note

If you set up record-level security in Microsoft Dynamics AX 4.0 with Service Pack 2, users who are members of more than one user group may have an unexpected level of access after you upgrade to Microsoft Dynamics AX 2009.

In previous releases, record-level security was additive for some scenarios, but not for others. In Microsoft Dynamics AX 2009, record-level security was made additive for all scenarios. This means that when a user belongs to multiple user groups with different permission levels, that user gets the highest level of privilege. For example, if one user group grants Read access to a table, and another user group grants Full access to the same table, a user belonging to both groups gets Full access to the table.

Consider the following examples, where User 1 belongs to both Group A and Group B, and where the table being accessed contains the records X, Y, and Z.

Example 1

In this example, neither user group has restrictions. In both versions of Microsoft Dynamics AX, this means that User 1 gets access to all records in the table.

Product version	Group A record-level security restriction	Group B record-level security restriction	Resulting access for User 1
Microsoft Dynamics AX 4.0 with Service Pack 2	None	None	XYZ
Microsoft Dynamics AX 2009	None	None	XYZ

Example 2

In this example, both user groups have restrictions, but they restrict different records in the table. In both versions of Microsoft Dynamics AX, this means that User 1 gets access to all records in the table.

Product version	Group A record-level security restriction	Group B record-level security restriction	Resulting access for User 1
Microsoft Dynamics AX 4.0 with Service Pack 2	Not Y	Not Z	XYZ
Microsoft Dynamics AX 2009	Not Y	Not Z	XYZ

Example 3

In this example, Group B restricts access to record Z, but Group A does not specify a record-level restriction. In Microsoft Dynamics AX 4.0 with Service Pack 2, this means that User 1 has restricted access because Group B has a restriction. In Microsoft Dynamics AX 2009, this means that User 1 has access to all data in the table because Group A allows access to all records.

Product version	Group A record-level security restriction	Group B record-level security restriction	Resulting access for User 1
Microsoft Dynamics AX 4.0 with Service Pack 2	None	Not Z	Not Z
Microsoft Dynamics AX 2009	None	Not Z	XYZ

Alert rules

Issue: Alert rules disabled during upgrade

Versions affected

Microsoft Dynamics AX 4.0

Upgrade note

All alert rules will be disabled when you upgrade from Microsoft Dynamics AX 4.0 to Microsoft Dynamics AX 2009. After the upgrade is complete you can enable alert rules again; however, some alert rules might fail after an upgrade because they reference forms that have changed in Microsoft Dynamics AX 2009. Alert rules that fail after an upgrade must be deleted and re-created.

Alerts that were received in Microsoft Dynamics AX 4.0 will be available after upgrade, but if the form that an alert references has changed, clicking the **Go to origin** button might not open the correct form and record by default. In these cases, you must manually browse in order to locate the form and record.

Installation

Issue: Benign error message

Versions affected

Microsoft Dynamics AX 3.0 and 4.0

Upgrade note

During the upgrade process to Microsoft Dynamics AX 2009, you may encounter the following message:

```
Object Server 01: Failed to create file old\axapd.aoi. Error code was 3
```

This message can be ignored, as it does not represent a data problem or upgrade failure. To avoid the message, before you begin the upgrade process, create a directory called "old" in the Application\App\Standard folder. For example: C:\Program Files\Microsoft Dynamics AX\5.0\Application\App\Standard\Old\.

Issue: Synchronization error during upgrade on SQL Server database

Versions affected

Microsoft Dynamics AX 4.0 SP2 (SQL Server only)

Upgrade note

During the upgrade process from Microsoft Dynamics AX 4.0 SP2 to version 5.0, an error occurs during synchronization on a SQL Server database installation. Failures are shown for four tables and the upgrade cannot proceed.

The four tables are:

- DimensionCollection
- DimensionHierarchyCombination
- DimensionSetCombinationDuplicate
- DimensionSetCombinationTemp

These tables are work tables for the Dimension Hierarchy feature and are only used during the creation of dimension hierarchy. The data is not maintained after the hierarchy is setup.

Workaround

To avoid this failure, the following SQL script must be executed on the Microsoft Dynamics AX 4.0 SP2 database before the upgrade is started:

```
--DIMENSIONCOLLECTION
UPDATE SQLDICTIONARY SET FIELDID=1 WHERE FIELDID=50002 AND TABLEID=2897;
UPDATE SQLDICTIONARY SET FIELDID=2 WHERE FIELDID=50005 AND TABLEID=2897;
UPDATE SQLDICTIONARY SET FIELDID=3 WHERE FIELDID=50007 AND TABLEID=2897;
UPDATE SQLDICTIONARY SET FIELDID=4 WHERE FIELDID=50008 AND TABLEID=2897;

--DIMENSIONHIERARCHYCOMBINATION
UPDATE SQLDICTIONARY SET FIELDID=1 WHERE FIELDID=50002 AND TABLEID=2898;
UPDATE SQLDICTIONARY SET FIELDID=2 WHERE FIELDID=50003 AND TABLEID=2898;
UPDATE SQLDICTIONARY SET FIELDID=3 WHERE FIELDID=50005 AND TABLEID=2898;
UPDATE SQLDICTIONARY SET FIELDID=4 WHERE FIELDID=50006 AND TABLEID=2898;
UPDATE SQLDICTIONARY SET FIELDID=5 WHERE FIELDID=50009 AND TABLEID=2898;

--DIMENSIONSETCOMBINATIONDUP2899
UPDATE SQLDICTIONARY SET FIELDID=1 WHERE FIELDID=50001 AND TABLEID=2899;
UPDATE SQLDICTIONARY SET FIELDID=2 WHERE FIELDID=50002 AND TABLEID=2899;
UPDATE SQLDICTIONARY SET FIELDID=3 WHERE FIELDID=50003 AND TABLEID=2899;
```

```
--DIMENSIONSETCOMBINATIONTEMP

UPDATE SQLDICTIONARY SET FIELDID=1 WHERE FIELDID=50001 AND TABLEID=2900;
UPDATE SQLDICTIONARY SET FIELDID=2 WHERE FIELDID=50002 AND TABLEID=2900;
UPDATE SQLDICTIONARY SET FIELDID=3 WHERE FIELDID=50003 AND TABLEID=2900;
UPDATE SQLDICTIONARY SET FIELDID=4 WHERE FIELDID=50004 AND TABLEID=2900;
UPDATE SQLDICTIONARY SET FIELDID=5 WHERE FIELDID=50005 AND TABLEID=2900;
UPDATE SQLDICTIONARY SET FIELDID=6 WHERE FIELDID=50006 AND TABLEID=2900;

--PROVISIONALHIERARCHY

UPDATE SQLDICTIONARY SET FIELDID=1 WHERE FIELDID=50001 AND TABLEID=2901;
UPDATE SQLDICTIONARY SET FIELDID=2 WHERE FIELDID=50002 AND TABLEID=2901;
UPDATE SQLDICTIONARY SET FIELDID=3 WHERE FIELDID=50007 AND TABLEID=2901;
```

Rapid Configuration Tool (RCT)

Issue: The RCT must be removed manually after upgrade

Versions affected

Microsoft Dynamics AX 3.0 and 4.0

Upgrade note

After you upgrade to Microsoft Dynamics AX 2009, the previous version of the RCT will remain visible in the navigation pane and code base even though it does not function with Microsoft Dynamics AX 2009.

Since there is currently no upgrade or uninstall script for the RCT, we recommend that you manually remove it from your Microsoft Dynamics AX 2009 installation. For example, if you installed Sure Step Rapid Configuration Tool 2.0 for Microsoft Dynamics AX 4.0 Service Pack 1, the shared projects **RctUserLayer** and **Class_PreImport** (in the **AOT**, navigate to **Shared > Projects**) contain the list of RCT elements that should be removed.

Code from the RCT system extended elements listed in these projects should also be evaluated and deleted manually. For example while upgrading from Sure Step Rapid Configuration Tool 2.0 for Microsoft Dynamics AX 4.0 Service Pack 1 to Microsoft Dynamics AX 2009, the following system elements, which are also extended by the RCT, must also be manually edited to remove RCT code:

Element type	Elements to be removed
Table	DocuRef
Map	AddressMap

Element type	Elements to be removed
Enum	DocuStructureType, NumberSeqModule
Class	DocuActionArchive, DocuActionCOM_Word, NumberSeqReference, SysDataExportExcel, SysDictConfigurationKey
Menu	MainMenu

Removing these now-deprecated customized RCT system elements will ensure that they do not override Microsoft Dynamics AX 2009 system classes.

At a later date, a new version of the Rapid Configuration Tool will be made available for download from [PartnerSource](#) and [CustomerSource](#).

Task Recorder

Issue: Previous version of Task Recorder must be uninstalled after upgrade

Versions affected

Microsoft Dynamics AX 4.0

Upgrade note

Microsoft Dynamics AX 2009 does not support a data upgrade from previously installed versions of Task Recorder, part of the Partner Productivity Tools 2.0 for Microsoft Dynamics AX 4.0 SP1. If a previous version is installed, the version of Task Recorder that is built into Microsoft Dynamics AX 2009 may not be available in the **Tools** menu.

To resolve this issue after upgrading to Microsoft Dynamics AX 2009:

1. Back up your Microsoft Dynamics AX 2009 data and application.
2. Press Ctrl+D to open the AOT, and then click **Projects > Shared > RecSystemOnly**.
3. Permanently delete the following customized classes from the layer where the old Task Recorder was installed:
 - Box
 - Info
 - ClassFactory

If these classes were customized for other purposes in the same layer where Task Recorder was installed, you must edit them in the AOT and manually replace all the old Task Recorder code lines with the new SYS-layer code lines. Use the compare tool to find all the old code lines which are between each pairing of these two comment lines:

```
//+TaskRecorder
...
//-TaskRecorder
```

4. Open the project **Shared > MiniRecorder**, and then delete all the items in this project.

 **Note:**

All previously saved task recordings will be deleted, as they cannot be upgraded to AX 2009. If other customizations were made to GlobalToolsMenu, you will need to manually delete the old `tskRecorder` entry from the AOT (**Menu > GlobalToolsMenu > Task Recorder**), and then bring in the new Task Recorder entry from the SYS layer (named `SysTaskRecorderToolbar`) using the compare tool.

5. Open the AOT and make sure all three classes (`Box`, `Info`, `ClassFactory`) can be compiled with no errors before proceeding.

 **Caution:**

Do not close the Microsoft Dynamics AX client until these classes compile without errors. Otherwise, you may not be able to reopen the client.

6. Close the Microsoft Dynamics AX client, and then stop the AOS. In the folder containing the application (e.g., `C:\Microsoft Dynamics AX\50\Application\Appl\DynamicsAx`), delete all existing RCT label files by opening a command line and running the following command:

```
del axRCT*.al?
```

 **Note:**

Deleting these files will result in removal of both Task Recorder and RCT labels.

7. Restart the AOS, open the Microsoft Dynamics AX client, and verify that the **Tools > Task Recorder** menu item is available and that it opens the correct **Task Recorder** form (named `SysTaskRecorderToolbar`).

Upgrade notes affecting specific modules

This topic addresses upgrade issues in the following functional areas:

- Asset depreciation schedules
- Bank
- Cubes
- Employee ID/user ID mapping
- Expense reporting
- Global address book
- Miscellaneous charges
- Modified and Created dates
- Pay agreement adjustments
- Payroll
- Print management
- Product model
- Quality management
- Request for quote (RFQ)
- Return orders
- Sales parameters
- Trade agreements

Asset depreciation schedules (ITA)

Issue: Field mapping issue may cause inaccurate depreciation schedules

Modules affected

GDL, General ledger, Fixed assets

Versions affected

Microsoft Dynamics AX 3.0 and 4.0

Upgrade note

A field-mapping issue has been identified that could cause a company to lose data in the AssetDepreciationProfile table during the upgrade to Microsoft Dynamics AX 2009, leading in some cases to invalid proposed depreciation schedules. This risk of data loss is possible only when the following conditions are met:

- The Italian configuration key is enabled for a company
- Depreciation profile records have been created for that company with these settings:
 - The method is set to Manual
 - The interval is set to Yearly
 - The calculation base is set to Months

Because of this field-mapping issue, the CalculationBase field in the AssetDepreciationProfile table will be dropped when the upgrade to Microsoft Dynamics AX 2009 is performed. (In a depreciation profile, the value of base enum CalculationBase determines whether to calculate a depreciation interval on a daily or monthly basis.) If the field is dropped during upgrade, it will be recreated with a value of 0, which represents the interval Days. This change would then yield incorrect results when depreciation proposals are generated using this particular depreciation profile, which previously had been set to Months.

To resolve this issue after the upgrade process has completed, you can review and reset the calculation base values to Months in the **Depreciation profiles** form (**General ledger > Setup > Fixed assets > Depreciation profiles**).

 **Note:**

You can also review depreciation lines and manually adjust any incorrect proposed depreciation intervals before the lines are posted.

Bank

Issue: Updating the Check logo source

Modules affected

Check logo

Versions affected

Microsoft Dynamics AX 3.0 and 4.0

Upgrade note

After you upgrade to Microsoft Dynamics AX 2009, your checks may print without the company logo.

In Microsoft Dynamics AX 2009, the logo that prints on checks is indicated in the BankChequeLayout table, whereas in previous versions, Microsoft Dynamics AX derived the logo for checks from the company logo.

If your bank account is set up to print the company logo on its check form, you can resolve this issue after you upgrade by following these steps:

1. Click **Bank > Common Forms > Bank Account Details > Setup > Check layout**.
2. Click the **Company logo** button.
3. Select the file of the logo you want printed on the checks for the selected bank account.

Cubes

Issue: Upgrade not supported

Modules affected

Reporting and business intelligence

Versions affected

Microsoft Dynamics AX 4.0

Upgrade note

If you are using the sample online analytical processing (OLAP) cubes that were provided with Microsoft Dynamics AX 4.0, or if you created a custom cube, note that there is no upgrade path for those cubes to Microsoft Dynamics AX 2009.

Microsoft Dynamics AX 2009 includes a new Business Intelligence framework that enables developers to generate cubes from the Microsoft Dynamics AX data model. Along with this framework, Microsoft Dynamics AX 2009 provides the following predefined cubes:

- Accounts Payable
- Accounts Receivable
- Customer Relationship Management
- Expense Management
- General Ledger
- Human Resources Management
- Production
- Project Accounting
- Purchase
- Sales

For information about installing these cubes, see the Microsoft Dynamics AX Installation Guide (<http://go.microsoft.com/fwlink/?LinkID=99623>).

Employee ID/user ID mapping

Issue: Restoring employee ID/user ID mapping

Modules affected

Expense management

Versions affected

Microsoft Dynamics AX 3.0

Upgrade note

When you upgrade from Microsoft Dynamics AX 3.0 to Microsoft Dynamics AX 2009, the employee/user mapping previously set up for your company will not be maintained.

To resolve this issue, after you complete the upgrade you must manually map all user IDs to employee IDs using the Active Directory Import Wizard (**Administration > Common Forms > Users > Import**).

Expense reporting

Issue: Reassigning expense management number sequences

Modules affected

Expense management

Versions affected

Microsoft Dynamics AX 3.0

Upgrade note

In Microsoft Dynamics AX 2009, expense management number sequences are no longer specified in the **Expense management parameters** form. Instead, they are specified in the **Number sequence references** form.

Note:

If you are upgrading from 3.0 to 4.1 as an intermediary step to Microsoft Dynamics AX 2009, you must open the **Expense management parameters** form and choose a new number sequence before expense reports can be created in your company.

After completing the upgrade to Microsoft Dynamics AX 2009, follow these steps:

1. Click **Basic > Setup > Number sequences > References** to open the **Number sequence references** form.
2. Locate the row for the **Expense management** module / **Expense report number** reference.

3. In the **Number sequence code** column, select the number sequence you want to assign.

 **Note:**

You can choose the same number sequence that was used in the Microsoft Dynamics AX 3.0 installation or a different one.

4. Click **File > Save**.
5. To confirm this change, click **Expense management > Setup > Parameters**.
6. Click the **Number sequences** tab and verify that the number sequence you chose has been applied.

Global address book

Issue: Professional prefix, Salutation, and Suffix field data not migrated

Modules affected

Basic, Human Resources

Versions affected

Microsoft Dynamics AX 4.0

Upgrade note

During the upgrade from Microsoft Dynamics AX 4.0, data from the employee **Professional prefix**, **Salutation**, and **Suffix** fields are not migrated to the Global address book in Microsoft Dynamics AX 2009. To restore data to these fields, you can add them to the Global address book manually (**Basic > Common Forms > Global Address Book Details > General**) or use the Microsoft Dynamics AX Fill utility to update the employee records in bulk. For information about using the Fill utility, see the help topic "Update multiple records using the Fill Utility" at **Microsoft Dynamics AX > Help > Applications and Business Processes**.

 **Note:**

The Global address book in Microsoft Dynamics AX 2009 supports user-defined values in these fields. In Microsoft Dynamics AX 4.0 these were system-defined values.

Issue: Setting up a prerequisite virtual company

Modules affected

CRM, Financials

Versions affected

Microsoft Dynamics AX 3.0 and 4.0

Upgrade note

A new step has been added to the upgrade process that creates one or more virtual companies for the Global address book (GAB), which is new to Microsoft Dynamics AX 2009.

The GAB lets you share data about customers, vendors, employees, contacts, departments, competitors, and business relations across multiple companies. This provides a holistic view of all activity for any one of these entities, such as seeing which companies share a common vendor or serve the same customer.

To share data for these entities across companies, you must set up a virtual company for each collection of one or more companies that will share a single GAB. This setup process is required for the Centralized customer statements feature, which lets you print a single customer statement for all companies that a customer does business with. The setup process is also required for the Centralized payments feature, which lets you pay a vendor in one company for an invoice in another company or receive a payment in one company for an invoice in another.

Note If you do not include one or more companies in a virtual company that are associated with the DirPartyCollection (the GAB), it is assumed that the GAB for that company will never be shared between companies.

Miscellaneous charges

Issue: 'Prices include sales tax' settings for 'Fixed or Pcs.' category

Modules affected

Accounts receivable, Accounts payable

Versions affected

Microsoft Dynamics AX 3.0 and 4.0

Upgrade note

Prior to Microsoft Dynamics AX 2009, miscellaneous charge amounts with the category of “Fixed or Pcs.” were sometimes considered exclusive of tax, but other times calculated to include sales tax per the **Prices incl. sales tax** option on the order header. This variance depended on which version and service packs were installed in an implementation.

Beginning with Microsoft Dynamics AX 2009, however, the value of the **Prices incl. sales tax** option found on the **Setup** tab of the order header determines whether the value entered for a miscellaneous charge code is considered tax-included or not.

This upgrade issue impacts implementations that use Purchase orders, Purchase requisitions, Sales orders, or Sales quotations when the following two conditions are met:

1. The **Prices incl. sales tax** check box is selected on the **Setup** tab in the order header.
2. One or more miscellaneous charge codes with a category of “Fixed or Pcs.” are used.

To respond to this issue after upgrading to Microsoft Dynamics AX 2009, first identify the list of documents that meet these criteria and then—in cases where the **Prices incl. sales tax** option is selected—make sure the miscellaneous charge amounts in the system reflect proper tax amounts.

The following scenario illustrates the behavior of Microsoft Dynamics AX 2009 in cases where miscellaneous charge amounts are included within the “Fixed or Pcs” category.

- A purchase order is created with the **Prices incl. sales tax** check box selected.
- A line item is 1 @ USD 100.00.
- A miscellaneous charge (category is “Fixed”) of USD 10.00 is included.
- Both the line item and the miscellaneous charge are subject to a 17% tax rate.

In Microsoft Dynamics AX calculations, this USD 100.00 net amount for the line item includes tax, so the amount of the line item without tax is calculated to be USD 85.47 ($100.00 / 1.17\% = 85.47$), and the tax calculated for the line item is \$14.53.

Because the purchase order header indicates that tax is included in the price, Microsoft Dynamics AX 2009 also calculates the USD 10.00 miscellaneous charge amount as already including tax. Therefore the amount of the miscellaneous charge without tax is calculated as USD 8.55 ($10.00 / 1.17 = 8.55$), and the tax is calculated to be USD 1.45.

Therefore, the line item (USD 85.47) + the miscellaneous charge (USD 8.55) + tax (USD 14.53 + USD 1.45) = USD 110.00.

Modified and Created dates

Issue: Modified and Created date values overwritten

Modules affected

Expense management

Versions affected

Microsoft Dynamics AX 3.0

Upgrade note

When you upgrade from Microsoft Dynamics AX 3.0 to Microsoft Dynamics AX 2009, the “Modified date” and “Created date” information stored in Microsoft Dynamics AX 3.0 will not be maintained. Instead, these fields will be populated with the date of the upgrade.

Pay agreement adjustments

Issue: Field-mapping issue may cause incorrect ordering of pay adjustments per each pay agreement

Modules affected

Shop Floor Control

Versions affected

Microsoft Dynamics AX 4.0 SP2

Upgrade note

A field-mapping issue has been identified that could cause a company to lose data in the JmgPayAdjustSetup table during the upgrade from Microsoft Dynamics AX 4.0 SP2 to Microsoft Dynamics AX 2009, leading to an incorrect ordering of pay adjustments per each pay agreement.

This upgrade issue causes a mismatch of the LineNum field ID between the database and the AOT, and the field is dropped and recreated during the synchronization step of the upgrade process. As a result, the LineNum field's values are lost and pay adjustments lose their original ordering.

To resolve this issue after the upgrade process:

1. Click **Shop Floor Control > Setup > Payroll > Pay agreements > Pay adjustment**.
2. On the **Pay adjustment** form, review the order of the pay adjustments that have been defined for the selected agreement.
3. Use the **Up** and **Down** buttons to reorder the payments as needed.

To resolve this issue before the upgrade process:

- Run the following script:

```
UPDATE SQLDICTIONARY SET FIELDID=1 WHERE FIELDID=2 AND TABLEID=8594
```

Payroll

Issue: Loss of Payroll data if Payroll code is not copied before main data upgrade

Modules affected

Payroll

Any LOS or BUS layer solutions

Versions affected

Microsoft Dynamics AX 4.0

Upgrade note

If you are upgrading to Microsoft Dynamics AX 2009 Payroll from Microsoft Dynamics AX 4.0 Payroll, there are additional steps to be completed during the standard Microsoft Dynamics AX 2009 upgrade process to ensure that your Microsoft Dynamics AX 4.0 Payroll data is not lost.

1. Copy your Microsoft Dynamics AX 4.0 Payroll application files as well as the other files listed in step 2 of the [Copy your existing application files](#) procedure.
2. Install Microsoft Dynamics AX 2009 according to the [Install Microsoft Dynamics AX before you upgrade](#) procedure.
3. Install the Microsoft Dynamics AX 2009 Payroll module to your Microsoft Dynamics AX 2009 production environment that you created in step 2 above.
4. Do not start the AOS until your Microsoft Dynamics AX 4.0 Payroll application files are copied into the Microsoft Dynamics AX 2009 application folder.
5. Start the Microsoft Dynamics AX 2009 AOS. (From the Start menu, click **Run**, type `Services.msc`, and click **OK**. On the **Services** window, right-click **Dynamics AX Object Server 5.0\$[DATABASE NAME]** and choose **Start**).

 **Note:**

The size and complexity of your database will affect how long it takes to start the AOS. It may take a long time to start.

6. Start the Microsoft Dynamics AX client. The upgrade checklist is displayed.
7. Open a new Microsoft Dynamics AX 2009 client workspace by clicking on the change company icon in the lower right corner of the main application window and open the same company in a new workspace.
8. Open the **License information** form (**Administration > Setup > System > License information**).
9. Verify that the appropriate Payroll entries are listed in the **Code description** column on the **Modules** tab from the Microsoft Dynamics AX 4.0 license file.
10. If the appropriate Payroll entries are listed, continue with the Microsoft Dynamics AX 2009 upgrade process.
11. Load your Microsoft Dynamics AX 2009 license file with Payroll code and verify that the **Status** column on the **Modules** tab in the **License information** form for the Payroll entries displays **OK**.
12. After the Microsoft Dynamics AX 2009 upgrade process is complete, you must upgrade payroll. Refer to your Payroll documentation for more information.

Print management

Issue: Changes in print management settings

Modules affected

Accounts receivable, Accounts payable, Project, CRM

Versions affected

Microsoft Dynamics AX 3.0 and 4.0

Upgrade note

The print management feature assists with the management of documents that are generated during the posting of Accounts payable purchase orders, Accounts receivable sales orders, Inventory pick lists, and Project invoices, as well as CRM sales quotation updates.

In connection with this feature, the following data points will impact Microsoft Dynamics AX 3.0 and 4.0 implementations that are being upgraded to Microsoft Dynamics AX 2009.

1. In both customer and partner customizations, several existing tables will be deleted and replaced by new tables that support this feature. These tables, which will be linked to the SysDeletedObjects41 configuration key, include:
 - PurchPrintCopies
 - PurchPrintSetup
 - SalesPrintCopies
 - SalesPrintSetup
 - ProjPrintCopies
2. Microsoft Dynamics AX 4.0 print management records that are associated with an account or transaction that no longer exists will be deleted.
3. Microsoft Dynamics AX 4.0 print management records with a module type of "User" will be deleted.
4. Microsoft Dynamics AX 4.0 print management records with a document status of "ProjectPackingSlip" will be deleted.
5. The Language ID of print management records will default to the default language for the company.
6. Microsoft Dynamics AX 4.0 print management records with a document status of "PickingList" will be related to Inventory and not Sales after the upgrade to Microsoft Dynamics AX 2009.

During the sales upgrade process, an additional record is created based on each document of the type "Confirmation". The additional record is a copy of the confirmation record but with the document type of "SalesOrderConfirmation". The original confirmation record is still upgraded as expected.

Issue: Changes in Print management setup options

Modules affected

Accounts receivable, Accounts payable, Project, CRM

Versions affected

Microsoft Dynamics AX 3.0 and 4.0

Upgrade note

The print management feature assists with the management of documents that are generated during the posting of Accounts payable purchase orders, Accounts receivable sales orders, Inventory pick lists, and Project invoices, as well as updates of CRM sales quotations. The ability to generate footer text in multiple languages and the query-driven selection of print management feature are new to Microsoft Dynamics AX 2009.

Documents that are managed by print management are printed from a posting or update form, depending on the document. These forms include a **Printer setup** button, which opens a standard Microsoft Dynamics AX printer setup form. If the **Use print management destination** check box is selected in the posting or update form, the document destination that is specified in the printer setup form is the default for Print management. If the **Use print management destination** check box is not selected in the posting or update form, the document destination that is specified in the **Printer setup** form is the destination for all documents.

In previous versions of Microsoft Dynamics AX, the number of copies specified for print management was multiplied by the number of copies in the printer setup form. In Microsoft Dynamics AX 2009, the number of identical copies indicated on the **Printer setup** form is not used with print management. Instead, the number of identical copies comes from the information specified in the **Print management setup** form.

Product model

Issue: Resolving VAT-only invoice discrepancies

Modules affected

General ledger

Versions affected

Microsoft Dynamics AX 4.0

Upgrade note

For invoices that contain exclusively Value Added Tax charges (VAT-only invoices), where the currency on the invoice is different than that specified on the **General** tab of the **Sales tax codes** form (**General ledger** > **Setup** > **Sales tax** > **Sales tax codes**), the amount for the posted sales tax is calculated incorrectly.

For Microsoft Dynamics AX 4.0 SP1 and SP2, DIS layer hot fixes have recently been implemented to resolve this issue. We strongly recommend applying these hot fixes before upgrading to Microsoft Dynamics AX 2009.

In addition, Microsoft Dynamics AX Customer Support has made available a report that you can run to identify whether any transactions in your database are affected by this problem. Contact your Microsoft Dynamics AX support representative to obtain the code for running this report.

To correct any issues identified by this report, create a Journal entry (**General ledger > Journals > General journal**) to register the correcting entry. You can use the voucher number that the report provides to identify the accounts that were used for the previous posting so you can post the correct values to those accounts. These entries should be made in the currency of the tax code to make up the difference in the sum that should have been posted originally. The posting date for this tax correction should be in the current period.

These corrections should ideally be made before upgrading to Microsoft Dynamics AX 2009, but can be run afterward if the issues are not identified beforehand.

Quality management

Issue: Separate upgrade script required for Quality Management Solution

Modules affected

Quality Management Solution

Note:

This module was known as "Total Quality Management" in Microsoft Dynamics AX 4.0.

Versions affected

Microsoft Dynamics AX 4.0

Upgrade note

Customers running the Total Quality Management module in Microsoft Dynamics AX 4.0 will need to run separate upgrade scripts to migrate Microsoft Dynamics AX 4.0 data to the Quality Management Solution in Microsoft Dynamics AX 2009.

Important:

Before doing so, however, they may need to adjust the string size of certain fields that have been changed in Microsoft Dynamics AX 2009. (These changes were made in order to conform to best practices for the length of the keys in some tables.)

For specific instructions, refer to the QMS Upgrade document which can be downloaded with the QMS Upgrade scripts on PartnerSource.

Request for quote (RFQ)

Issue: Manual adjustments required for purchase quotations

Modules affected

Accounts receivable

Versions affected

Microsoft Dynamics AX 3.0 and 4.0

Upgrade note

A purchase order of the type "Quotation" that is created in Microsoft Dynamics AX 3.0 or 4.0 without any lines will not be converted to a Request for quote (RFQ) during the upgrade to Microsoft Dynamics AX 2009. Instead, a new RFQ will have to be created manually in Microsoft Dynamics AX 2009 after the upgrade.

In addition, tax adjustments that are entered on purchase quotations will not be converted to RFQs, nor can tax adjustments be entered on RFQs after the upgrade to Microsoft Dynamics AX 2009 because tax adjustments are not supported for RFQs.

Return orders

Issue: Return orders should be invoiced before upgrading

Modules affected

Accounts receivable

Versions affected

Microsoft Dynamics AX 3.0 and 4.0

Upgrade note

Sales orders of the type Returned item should be invoice updated before you begin the process of upgrading to Microsoft Dynamics AX 2009. Not doing so could result in inventory transactions being out of sync with sales order lines.

Sales parameters

Issue: Possible upgrade failure in Sales parameters table

Modules affected

Accounts receivable

Versions affected

Microsoft Dynamics AX 4.0

Upgrade note

During the post-synchronization process in some upgrades from Microsoft Dynamics AX 4.0 to Microsoft Dynamics AX 2009, the ReleaseDB41_Cust::createOrderEntryDeadline may fail intermittently because it cannot edit a record in the Sales parameters table (SalesParameters). To resolve this issue, run the Microsoft Dynamics AX 2009 upgrade script a second time.

Issue: Required Matrix report postings before upgrade

Modules affected

Expense management

Versions affected

Microsoft Dynamics AX 3.0

Upgrade note

If any expense reports stored but not yet posted in your Microsoft Dynamics AX 3.0 implementation are of the type "Matrix", these reports must be posted before upgrading to Microsoft Dynamics AX 2009. Matrix transactions are not supported in Microsoft Dynamics AX 2009.

We recommend that all expense reports be posted before performing the upgrade to Microsoft Dynamics AX 2009, not only those of the type "Matrix".

Note In Microsoft Dynamics AX 2009 the Split expense feature is available as an alternative to Matrix transactions.

Trade agreements

Issue: Expanded display of 'Valid for' and 'Price group' columns

Modules affected

Project

Versions affected

Microsoft Dynamics AX 4.0

Upgrade note

After you upgrade from Microsoft Dynamics AX 4.0 to Microsoft Dynamics AX 2009, the columns **Valid for** and **Price group** will display in the following forms, in addition to those where they already appeared in previous versions:

- **Sales price - Hour**
- **Sales price - Expense**
- **Sales price - Fee**
- **Sales price - Subscription**

The following tables are affected by this change:

- projHourSalesPrice
- projCostSalesPrice
- projRevenueSalesPrice
- smaSalesPriceSubscription

Existing sales price records in these tables that contain a Project ID will have the setting "Table" in the **Valid for** column. All other records will display "All" in the **Valid for** column.

Picking lists

Issue: CustPickingListJour and CustPickingListTrans data is not migrated in upgrade

Modules affected

Inventory management

Versions affected

Microsoft Dynamics AX 3.0 and 4.0

Upgrade note

Microsoft Dynamics AX 2009 upgrade only migrates data from non-picked picking lists. Data from picked picking lists (the tables CustPickingListJour and CustPickingListTrans) cannot be migrated. This is by design because there is little value in keeping old picking lists and not

migrating that data improves upgrade performance by reducing the amount of data to be migrated.

If you need to print old picking lists from your current version of Microsoft Dynamics AX to keep as hard-copy records, you should print them before you upgrade to Microsoft Dynamics AX 2009.

Upgrade notes affecting specific locales

This topic addresses upgrade issues for the following locales:

- Brazil
- Finland

Brazil

Issue: Lack of sales and purchase lines on new discount fields update

Modules affected

Accounts receivable, Accounts payable

Versions affected

Microsoft Dynamics AX 3.0 and 4.0

Upgrade note

A set of new fields and upgrade scripts has been created to ensure that header discount information continues to be applied to invoice lines. However, the upgrade scripts rely on the InventTable data to define whether the discount should be applied, while referring to PurchLine and SalesLine table information to re-calculate the corresponding amounts. In cases where the setup has changed for items where the discount header applies, or where the purchase or sales line no longer exists, the corresponding field must be updated manually.

This issue does not affect implementations where the header discount on orders has not been used.

The new fields and upgrade scripts were created to keep information on header discount applied to invoice line:

- \Data Dictionary\Tables\CustInvoiceTrans\Fields\LineEndDiscAmount_BR
- \Data Dictionary\Tables\VendInvoiceTrans\Fields\LineEndDiscAmount_BR
- \Data Dictionary\Tables\VendInvoiceTrans\Fields\SumLineDiscAmount_BR

The following fields are updated to reflect the new discount fields:

- \Data Dictionary\Tables\CustInvoiceJour\Fields\InvoiceGoodsAmount
- \Data Dictionary\Tables\CustInvoiceJour\Fields\InvoiceServicesAmount
- \Data Dictionary\Tables\VendInvoiceJour\Fields\InvoiceGoodsAmount
- \Data Dictionary\Tables\VendInvoiceJour\Fields\InvoiceServicesAmount

 **Note:**

The individual layer for Brazil is included in the Microsoft Dynamics AX 2009 consolidated GLS layer. Before you can install a consolidated GLS layer, you must install Service Pack 1 for Microsoft Dynamics AX 2009. For information about upgrading the GLS layer for a specific country, see the [Service Pack 1 page](#) on CustomerSource.

Issue: Lack of sales and purchase tables on carrier information update

Modules affected

Accounts receivable, Accounts payable

Versions affected

Microsoft Dynamics AX 3.0 and 4.0

Upgrade note

The upgrade relies on carrier information in CustInvoiceJour and VendInvoiceJour tables. In cases where the related sales or purchase order has been deleted, the fields DivStateRegistered and DivPackingName will not be updated.

Issue: Lack of sales table and sales lines on sales invoice reference update of complementary and returned invoices

Modules affected

Accounts receivable

Versions affected

Microsoft Dynamics AX 3.0 and 4.0

Upgrade note

The upgrade uses information in the SalesTable and SalesLines tables to update the sales invoice reference and sales invoice line reference in, respectively, complementary invoices (which create amendments on tax or price to a previous invoice), and in returned invoices. If the sales order related to these references has been deleted, these fields will not be updated in the invoice data.

Issue: Lack of sales lines on service code update

Modules affected

Accounts receivable

Versions affected

Microsoft Dynamics AX 3.0 and 4.0

Upgrade note

The upgrade uses carrier information stored in the SalesLine table to update service code information in sales invoices. If the sales lines related to a particular sales invoice have been deleted from the sales line table, the service code will not be updated in that sales invoice.

Issue: Add dimension and change extended data type (EDT) size of Name field from 60 to 80

Modules affected

Accounts receivable, Accounts payable

Versions affected

Microsoft Dynamics AX 3.0 and 4.0

Upgrade note

Before performing the data upgrade to Microsoft Dynamics AX 2009, you should open and run the form **SysDimensionChanger_BR**, available in the AOT, in order to create a fourth dimension, "Branch", and change the EDT of the **Name** field to 80.

Finland

Issue: Preserving payment reference numbers in Daily journals

Modules affected

Accounts receivable

Versions affected

Microsoft Dynamics AX 3.0 and 4.0

Upgrade note

An issue has been identified that will require Finnish companies to run a hot fix before beginning the upgrade process if they are using the Daily journal to import open Accounts receivable entries as part of the upgrade.

This hot fix is described in Microsoft Dynamics AX knowledge base article 949906, which is available via CustomerSource and PartnerSource.

Microsoft Dynamics AX

The issue relates to the common practice in Finland of using a payment reference number (in addition to the invoice number) as part of the invoice transaction so that customers who pay electronically can refer to this number for confirmation that their payment is being applied to the appropriate charge.

If the hot fix is not applied before the upgrade, the system will calculate the value in the Payment reference field a second time during the upgrade process, overwriting the original Payment reference value, after which the customer payment cannot be matched to the correct original invoice.

Upgrade troubleshooting

This section provides information about troubleshooting issues that can be encountered when upgrading a Microsoft Dynamics AX system. This section contains the following topics.

- [Troubleshooting the Microsoft Dynamics AX DB Upgrade Preparation tool](#)
- [Troubleshooting AOS or Microsoft Dynamics AX client startup during upgrade](#)
- [Compiling Microsoft Dynamics AX troubleshooting](#)
- [Troubleshooting the Data upgrade cockpit](#)
- [Error messages during upgrade](#)

Troubleshooting the Microsoft Dynamics AX DB Upgrade Preparation tool

This section contains information about issues that you may encounter while using the Microsoft Dynamics AX DB upgrade preparation tool during an upgrade.

Using FRx with the Microsoft Dynamics AX DB Upgrade Preparation tool

If you are using FRx when you run the Microsoft Dynamics AX DB Upgrade Preparation tool it will fail on the `frl_entity` table and stop the process of copying tables to the new Microsoft Dynamics AX database. The process fails because the `frl_entity` has an identity column.

To prevent this failure you need to complete one of the following options:

Option 1:

Delete the `frl_entity` table before running the Microsoft Dynamics AX DB Upgrade Preparation tool. When the upgrade to Microsoft Dynamics AX 2009 is finished, use Data Transformation Services (DTS) to restore a backup of the `frl_entity` table from your Axapta 3.0 database to your new Microsoft Dynamics AX database.

Note:

The same process of using DTS can be used for any table in the Dynamics AX database that does not also exist in the AOT. Use the `DBPrep_nonAOTtable_SQL.sql` or `DBPrep_nonAOTtable_Ora.sql` scripts to identify those tables affected.

1. Before you start the Microsoft Dynamics AX DB Upgrade Preparation tool make a backup of your Axapta 3.0 database.
2. Delete the `frl_entity` table in your Axapta 3.0 database.
3. Run the Microsoft Dynamics AX DB Upgrade Preparation tool and complete the upgrade to Microsoft Dynamics AX 2009.
4. In SQL Server 2000 Enterprise Manager or SQL Server 2005 Management Studio restore the backup of the Axapta 3.0 from step 1. Export the `frl_entity` table from the Microsoft Dynamics AX 3.0 database to the Microsoft Dynamics AX 2009 database.

Option 2:

Delete the `frl_entity` table in your Axapta 3.0 database before running the Microsoft Dynamics AX DB Upgrade Preparation tool. Then recreate the `frl_entity` table after the update to Microsoft Dynamics AX 2009 is complete.

Microsoft Dynamics AX DB upgrade preparation tool and Oracle

If you are using Oracle, be sure that the Oracle 32-bit client (Administrator SKU) libraries are present before you run the Microsoft Dynamics AX DB Upgrade Preparation tool.

Virus scanner stops the Microsoft Dynamics AX DB Upgrade Preparation tool

The Microsoft Dynamics AX DB Upgrade Preparation tool creates interim files that may falsely trigger your virus scanner, causing the DB Upgrade Preparation Tool to fail.

If this problem occurs, disable your virus scanner and restart the Microsoft Dynamics AX DB Upgrade Preparation Tool. Be sure to start your virus scanner again after you are done using the Microsoft Dynamics AX DB Upgrade Preparation tool.

Microsoft Dynamics AX DB Upgrade Preparation tool fails

Issue: The Microsoft Dynamics AX DB Upgrade Preparation tool fails and a syntax error is displayed while creating objects in the target database.

Possible cause #1: You have non-AOT-based tables in the source database schema.

Solution: Follow the steps in the "Non-AOT-based tables" section of the [The Microsoft Dynamics AX DB Upgrade Preparation tool](#) topic.

Possible cause #2: In your Microsoft Dynamics AX 3.0 database, you have used database reserved keywords as object, index, and constraint names.

Solution: Rename the object, index, and constraint names that use database reserved words to something else within the Microsoft Dynamics AX 3.0 database, run the Microsoft Dynamics AX DB Preparation tool to copy your data, rename the object, index, and constraint names back to their original names. For example, if one of the table columns is named INTERNAL, rename to XINTERNAL and after you run the Microsoft Dynamics AX DB Preparation tool, rename the table column back to INTERNAL.

Microsoft Dynamics AX DB Upgrade Preparation tool fails on the first table AccessRightsList

Issue: After the Bulk Copy Process (BCP) starts, the scripts fail on the first table AccessRightsList.

Possible Cause: The Microsoft Dynamics AX DB Upgrade Preparation tool does not have rights or cannot access the BCP temp folder.

Solution: The location of the BCP temp folder for SQL Server must be local to the destination SQL server where the new Microsoft Dynamics AX 2009 database is located.

Microsoft Dynamics AX DB Upgrade Preparation tool doesn't start

Issue: The Microsoft Dynamics AX DB Upgrade Preparation tool does not open because of an unhandled exception on startup.

Possible cause: You might not have installed the prerequisite software during the installation of Microsoft Dynamics AX 2009.

Solution: Install the prerequisite software from the installation media.

Troubleshooting AOS or Microsoft Dynamics AX client startup during upgrade

This section contains information about issues that you may encounter when you start the Application Object Server (AOS) or your Microsoft Dynamics AX 2009 client for the first time during the upgrade process.

AOS or Upgrade checklist does not start

Issue: The AOS or the Upgrade checklist does not start during the upgrade process.

Possible cause #1: You store your data using multiple collations. Using multiple collations for your databases is not a supported scenario.

Solution: Convert your database to use a single collation. See the SQL Server or Oracle documentation for more information. Then complete the following steps.

1. Verify that your system is working properly and that there are no data issues.
2. Begin upgrading to Microsoft Dynamics AX 2009.

Possible cause #2: You have modified any of the following:

- System tables (for example, SysSetupCompanyLog or SysUserInfo), which are used during the AOS startup.
- Any elements used under global classes (Info or Application nodes).
- Database elements which prevent table objects from being dropped or created, for example, user statistics on a field which is synchronized during AOS startup.

Solution: Delete these modifications from Microsoft Dynamics AX 4.0 and move them to Microsoft Dynamics AX 2009 before starting the upgrade. View the Windows event viewer (**Administrative Tools > Event Viewer**) to see if there are errors related to SQL statements. Debug the failures associated with SQL statements to fix the problem.

 **Note:**

We do not recommend customizing system classes or objects.

Possible cause #3: The AOS was not upgraded properly because you did not specify the correct Microsoft Dynamics AX database to install during the Microsoft Dynamics AX 2009 installation. This results in a message that says, "The version of the stored procedures in this database is different than that expected by the Application Object Server (AOS). You must set up a new instance of the AOS that points to the updated version of the stored procedures."

Solution: Two solutions exist.

- Reinstall Microsoft Dynamics AX 2009 and specify the correct Microsoft Dynamics AX database that needs to be upgraded during the installation.
- Open Microsoft SQL Server Management Studio and copy the `sysserveressions` and `sysclientsessions` stored procedures from a working Microsoft Dynamics AX 2009 instance to your Microsoft Dynamics AX database. Next, view the value of the `SYSSPVERSION` parm (column) on the `SQLSYSTEMVARIABLES` table using the following query:

```
select * from SQLSYSTEMVARIABLES where parm='SYSSPVERSION';
```

- If the value is not equal to "8", execute the following query:

```
Update SQLSYSTEMVARIABLES set value=8 where parm='SYSSPVERSION';
```

- For a database requiring an explicit commit, run:

```
Commit;
```

AOS server crashes after you upgrade

Issue: The AOS server crashes after you upgrade from Microsoft Dynamics AX 4.0 to Microsoft Dynamics AX 2009.

Possible cause: Table constraints are missing in the `sysserveressions` table and in the `sysclientsessions` table, or in all Microsoft Dynamics AX tables.

Solution: See the "Missing constraints error message" section in the [Error messages during upgrade](#) topic.

Microsoft Dynamics AX client crashes at startup during the upgrade process

Issue: You have completed all of the steps in [Prepare to upgrade](#) and the Microsoft Dynamics AX 2009 client crashes.

Possible cause #1: You have made class modifications that affect startup.

Solution: Check whether you have modified any low-level code that is run during startup or code that is started from the Classes node, such as the Application or Info nodes.

If you have modified classes that prevent Microsoft Dynamics AX from starting up correctly, delete your modifications.

Possible cause #2: There are name changes that conflict with the new database structure.

Solution: It is possible that there are field-name modifications that conflict with additions made by `mcCoName`. Field names must have unique IDs. Open the event viewer (**Administrative tools > Event viewer**) and check the event log, which identifies any fields that are duplicated.

Compiling Microsoft Dynamics AX troubleshooting

This section contains information about issues that you may encounter when compiling Microsoft Dynamics AX 2009 during the upgrade process.

Insufficient rights error during compile

If you get insufficient rights errors when you compile Microsoft Dynamics AX during upgrade, check whether you have modified classes in either the VAR layer or BUS layer (and their patch layers) that require you to load a license. If you have modified classes that start up early and require a license, start the Application Object Server (AOS) and load the correct license before you copy the updated application files to the Standard folder. The upgrade sequence would then be as follows:

1. Install Microsoft Dynamics AX (includes application files, client, AOS, and new empty database).
2. Run the Microsoft Dynamics AX DB Upgrade Preparation Tool from the installation media.
3. Start the AOS.
4. Start the Microsoft Dynamics AX client.
5. Load your license keys by clicking **Administration > Setup > System > License information**. Then click the **Load License File** button.
6. Close the Microsoft Dynamics AX client.
7. Click **Start > Administrative Tools > Services** and stop the AOS (identified in the Service list as the **Dynamics AX Object Server**).
8. Copy the application files (*.aod) from all layers above the HFX layer to:
 \Microsoft Dynamics\5.0\Application\App\Standard
9. Copy application files (*.aod) from all layers above the HFX layer to:
 \Microsoft Dynamics\5.0\Application\App\Standard\Old
10. Delete the index file (**axapd.aoi**) from the Standard folder.
11. Start the AOS through the **Services** list.
12. Start the Microsoft Dynamics AX client.
13. Run through the Upgrade checklist.

Troubleshooting the Data upgrade cockpit

This section contains information about issues you may encounter when you use the **Data upgrade cockpit**.

Upgrade job fails to start

Issue: The upgrade job doesn't start after clicking the **Run** or **Trial Run** buttons in the **Data upgrade cockpit** and all upgrade scripts remain in the **Ready** state for a long time.

 **Note:**

You can update the upgrade status information manually if you click the **Refresh** button in the **Data upgrade cockpit**. To enable automatic updating of the upgrade status information, mark the **Auto refresh enabled** check box.

Possible cause #1: The database has been upgraded before (from Microsoft Axapta 2.5 to Microsoft Dynamics AX 3.0, for example) and a batch group named DataUpdate still exists but an associated job doesn't exist for this batch group.

Solution: Follow the steps below.

1. Open the **Batch group** form (**Administration > Setup > Batch groups**).
2. Select **DataUpdate** group.
3. Click the **Batch Servers** tab.
4. Verify that the current AOS is listed in the **Selected servers** list. If it is not listed there, move it there from the **Remaining servers** list and restart the upgrade.

Possible cause #2: The batch server defined for running the upgrade job is not running.

Solution: Open the **Server configuration** form (**Administration > Setup > Server configuration**). Verify that **Is Batch Server** is marked for the AOS instance that is running the upgrade.

Possible cause #3: The DataUpgrade batch group is not a selected group for the AOS instance that is running the upgrade.

Solution: Open the **Server configuration** form (**Administration > Setup > Server configuration**). On the **Batch server groups** tab, verify that **DataUpdate** batch group is listed in the **Selected groups** list.

Data loss during synchronization step

If you experience data loss during the synchronization step of the **Upgrade checklist**, check the following things.

1. Look for the **Synchronization warning** page and make sure there is no error.
2. During synchronization, the AOT is synchronized with the database. If you have customizations which weren't properly upgraded, then these changes may not be the part of AOT and will get dropped during the synchronization process.
3. During synchronization, if field IDs or table IDs have changed in Microsoft Dynamics AX 2009 from what they were in Microsoft Dynamics AX 3.0 or Microsoft Dynamics AX 4.0, data in those fields or tables will be lost. To avoid this, create an upgrade script to change the TableID as a part of the presynchronize process. See the white paper ["How to Write Data Upgrade Scripts for Microsoft Dynamics AX 2009"](#) for more information. Also, refer to the "Potential for dropped tables" section for information about a related issue.
4. Changes to the database schema, such as defining user statistics on columns, may prevent objects from being synchronized with the AOT, leading to errors. Review the object causing the error and investigate by running the SQL statement (obtained from the Microsoft Dynamics AX Infolog or Windows Event Viewer) directly on the database using SQL Server Management Studio or the Oracle equivalent, such as SQLplus.
5. You encounter an error mentioning **DEL_<table>**, which may indicate this table was not removed after a previous upgrade. Back up your Microsoft Dynamics AX 3.0 database, and then disable the **Keep update objects 3.0** configuration key. When you clear these check boxes, any obsolete DEL_ objects in your source database will be deleted. Verify that your system runs properly before continuing the upgrade process.
6. The following message is displayed: "Cannot execute a data definition language command on <table_name>," where <table_name> is a placeholder for the actual table where the issue occurs, and, in the AOS Application Event Log, a message is displayed stating that SQL was unable to rename an object because the object name already exists.
Start Microsoft Dynamics AX 2009, open the AOT and locate the table where the issue occurs. If there are two tables with the same name, check to see if one exists in only the SYS layer and the other exists in SYS and another layer. If this is the case, right-click on the table that exists in two different layers and choose **Compare**. Use the **Compare** tool to see if the name of the table has been changed. The table in question might have been renamed DEL_[table_name] in the new version. Make note of any modifications, and then delete the current layer object, the DEL_[table_name] will now be displayed in the AOT, and the table will no longer give a synchronization error.

7. You get a message that says that duplicate field names exist but with different ID values. One way to fix this is to write upgrade scripts for the presynchronization process to fix the issue. Refer to the white paper "[How to Write Data Upgrade Scripts for Microsoft Dynamics AX 2009](http://go.microsoft.com/fwlink/?LinkId=115169&clcid=0x409)" (<http://go.microsoft.com/fwlink/?LinkId=115169&clcid=0x409>). This issue may also be caused by exporting a table from one layer to another layer with object IDs. Be sure you do not export object IDs when exporting objects.

Data loss or upgrade task failures during Postsynchronize step

If you experience data loss during the **Postsynchronize** step of the **Upgrade checklist**, check the following things.

1. Upgrade scripts can run only if the appropriate configuration keys are enabled. Find the tables or fields where the data was not upgraded and be sure that any configuration keys associated with those tables or fields are enabled.
2. If an upgrade script fails, be sure to check the SQL statement causing the failures. You can find the SQL statements in the Microsoft Dynamics AX infolog or Windows Event Viewer.

Detecting performance issues with upgrade scripts

If you are upgrading a large database, which may take several hours, you can improve performance by increasing the auto-refresh rate in the **Auto-refresh rate time (sec)** field to 1800 seconds or turn off the auto-refresh by clearing the checkbox, which will help improve performance.

You also can increase the number of batch threads defined for the upgrade batch server. Open the **Server configuration** form (**Administration > Setup > Server configuration**). On the **Batch server schedule** tab, enter a desired number in the **Maximum batch threads** column. Be sure to test the upgrade performance whenever you change the number of threads.

If you are using Oracle, be sure that you have the latest supported Oracle client with all of the required hotfixes installed.

For more information about ways to improve performance during data upgrade, read the [Improving data upgrade performance](#) topic.

You can detect performance issues within the upgrade job by doing the following.

1. Sort through the **Start time**, **Duration**, and **Upgrade job ID** columns in the Data Upgrade Cockpit to identify tasks that are running an unusually long time.
2. Click Cancel to prevent startup of all upgrade tasks with a status of Waiting. Tasks that have already been submitted to the batch server will continue to run. Because you abruptly stopped the process, invalid data might remain in the database, so you should restore from your backup before you attempt to perform the data upgrade again.

Synchronization fails during upgrade

Issue: If you use SQL Server, and you are upgrading from Microsoft Dynamics AX 4.0 SP1 with an application build equal to or greater than 4.0.2500.61 or from Microsoft Dynamics AX 4.0 SP2, an error can occur during synchronization. In this case, the four tables listed below will fail and you will be unable to continue with the upgrade.

- DimensionCollection
- DimensionHierarchyCombination
- DimensionSetCombinationDuplicate
- DimensionSetCombinationTemp

Possible cause: The field IDs for the four tables listed above are different in Microsoft Dynamics AX 4.0 SP2 from Microsoft Dynamics AX 2009. However, the table IDs are the same in both releases, which causes the upgrade scripts to fail to start the upgrade code process to correct the field IDs. Because the field IDs are different in the database and the AOT, the fields are dropped and recreated during the synchronization step. The synchronization step fails because there are SQL indexes that reference the fields and prevent their removal.

Solution: Before you upgrade, you must execute the following SQL script on your Microsoft Dynamics AX 4.0 SP2 database.

```
--DIMENSIONCOLLECTION
```

```
UPDATE SQLDICTIONARY SET FIELDID=1 WHERE FIELDID=50002 AND TABLEID=2897;
UPDATE SQLDICTIONARY SET FIELDID=2 WHERE FIELDID=50005 AND TABLEID=2897;
UPDATE SQLDICTIONARY SET FIELDID=3 WHERE FIELDID=50007 AND TABLEID=2897;
UPDATE SQLDICTIONARY SET FIELDID=4 WHERE FIELDID=50008 AND TABLEID=2897;
```

```
--DIMENSIONHIERARCHYCOMBINATION
```

```
UPDATE SQLDICTIONARY SET FIELDID=1 WHERE FIELDID=50002 AND TABLEID=2898;
UPDATE SQLDICTIONARY SET FIELDID=2 WHERE FIELDID=50003 AND TABLEID=2898;
UPDATE SQLDICTIONARY SET FIELDID=3 WHERE FIELDID=50005 AND TABLEID=2898;
UPDATE SQLDICTIONARY SET FIELDID=4 WHERE FIELDID=50006 AND TABLEID=2898;
UPDATE SQLDICTIONARY SET FIELDID=5 WHERE FIELDID=50009 AND TABLEID=2898;
```

```
--DIMENSIONSETCOMBINATIONDUP2899
```

```
UPDATE SQLDICTIONARY SET FIELDID=1 WHERE FIELDID=50001 AND TABLEID=2899;
UPDATE SQLDICTIONARY SET FIELDID=2 WHERE FIELDID=50002 AND TABLEID=2899;
UPDATE SQLDICTIONARY SET FIELDID=3 WHERE FIELDID=50003 AND TABLEID=2899;
```

```
--DIMENSIONSETCOMBINATIONTEMP
UPDATE SQLDICTIONARY SET FIELDID=1 WHERE FIELDID=50001 AND TABLEID=2900;
UPDATE SQLDICTIONARY SET FIELDID=2 WHERE FIELDID=50002 AND TABLEID=2900;
UPDATE SQLDICTIONARY SET FIELDID=3 WHERE FIELDID=50003 AND TABLEID=2900;
UPDATE SQLDICTIONARY SET FIELDID=4 WHERE FIELDID=50004 AND TABLEID=2900;
UPDATE SQLDICTIONARY SET FIELDID=5 WHERE FIELDID=50005 AND TABLEID=2900;
UPDATE SQLDICTIONARY SET FIELDID=6 WHERE FIELDID=50006 AND TABLEID=2900;

--PROVISIONALHIERARCHY
UPDATE SQLDICTIONARY SET FIELDID=1 WHERE FIELDID=50001 AND TABLEID=2901;
UPDATE SQLDICTIONARY SET FIELDID=2 WHERE FIELDID=50002 AND TABLEID=2901;
UPDATE SQLDICTIONARY SET FIELDID=3 WHERE FIELDID=50007 AND TABLEID=2901;
```

Potential for dropped tables

During synchronization, you might encounter an error that indicates that Microsoft Dynamics AX 2009 wants to drop and recreate a data table, which would cause the loss of all data in the table.

This issue may occur when you are re-implementing existing customizations instead of upgrading them. This generally occurs when a table ID changes, and causes the SQLDictionary table to not match what is in the AOT representation for the table. By dropping and recreating the table during the synchronization process, the SQLDictionary table is updated.

Listed below are two options for resolving this issue.

Method 1

The preferred way to resolve this issue is to use a static method of the ReleaseUpdateDB class. This method should be run during the presynchronization process to preserve the table data. For this example, the ReleaseUpdateDB41_Administration class will be used. The class used is dependent upon the module the table is related to.

1. Open the AOT.
2. Expand the **Classes** node.
3. Locate the ReleaseUpdateDB41_Administration class.
4. Double-click on this class to open the class in the editor.
5. Click on the **New** button to create a new method.

6. Type in the following code:

```
void RepairCustomTable()
{
    TableID    oldID;
    TableID    newID;
;
    oldID = 50008;
    newID = 50001;

    ReleaseUpdateDB::changeTableName("COREEXCHANGETABLE",oldID,newID);
}

```

7. Save the code.
8. Modify the `initPreSyncJobs` method to call the method created in steps 5–7. Highlight the `initPreSyncJobs` method in the left-most pane of the editor. The code is now displayed.
9. Locate the section in the code beginning with "**this.addStandardJob.**"
10. Add the following line after the jobs listed in this section.

```
this.addStandardJob(methodStr(ReleaseUpdateDB41_Administration,
    RepairCustomTable),'Backing up custom table');
```

11. Save the code. Now, when you run the upgrade process again, the custom code should load, and the table should be fixed.

Method 2

Another way to resolve this issue is to create a temporary table, populate it with the data from the table that has the issue, allow Microsoft Dynamics AX 2009 to drop and recreate the table, and then restore the data from the temporary table to the newly created table.

1. Open the AOT.
2. Expand the **Classes** node.
3. Locate the `ReleaseUpdateDB41_Administration` class.
4. Double-click on this class to open the class in the editor.
5. Click on the **New** button to create a new method.

6. Type in the following code:

```

void backupCustomTable()
{
    //back up a custom table so it may be dropped by the synch process

    SqlStatementExecutePermission    permission;
    Connection                       connection;
    Statement                        statement;
    SalesTable                       salesTable;
    SqlSystem                        sqlSystem = new SqlSystem();

    str ext;
    int tickCount;
    str sqlStatement;
    ;

    //create our connection
    connection = new Connection();
    statement  = connection.createStatement();

    // coreexchangetable is our table with issues
    sqlStatement = strfmt(@"select * into COREEXCHANGETABLE_temp from
COREEXCHANGETABLE");

    permission = new SqlStatementExecutePermission(sqlStatement);
    permission.assert();
    //BP deviation documented
    statement.executeUpdate(sqlStatement);
    CodeAccessPermission::revertAssert();
}

```

7. Save the code.

8. Modify the `initPreSyncJobs` method to call the method created in steps 5–7. Highlight the `initPreSyncJobs` method in the left-most pane of the editor. The code is now displayed.
9. Locate the section in the code beginning with `"this.addStandardJob."`
10. Add the following line after the jobs listed in this section.

```
this.addStandardJob(methodStr (ReleaseUpdateDB41_Administration,
RepairCustomTable), 'Backing up custom table');
```

11. Save the code.
12. The next step creates the method to restore the data after the synchronization process has recreated the table. Click on the **New** button to create a new method.
13. Type the following code:

```
void restoreCustomTable()
{
    // back up a custom table so it may be dropped by the synch process
    SqlStatementExecutePermission    permission;
    Connection                        connection;
    Statement                          statement;
    SalesTable                         salesTable;
    SqlSystem                          sqlSystem = new SqlSystem();

    str ext;
    int tickCount;
    str sqlStatement;
    ;

    //create our connection
    connection = new Connection();
    statement = connection.createStatement();

    // coreexchangetable is our table with issues
    sqlStatement = strfmt(@"insert into COREEXCHANGETABLE
(EXCHANGEITEMSPERASSEMBLY, EXCHANGEITEMGROUP, EXCHANGEITEMNUMBER, ASSEMBLYITEMNUMBER, CORERET
URNPERIOD, EXCHANGECONFIGID, ASSEMBLYCONFIGID,
MODIFIEDDATE, MODIFIEDTIME, MODIFIEDBY, MODIFIEDTRANSACTIONID, CREATEDDATE, CREATEDTIME, CREATE
DBY, CREATEDTRANSACTIONID, DATAAREAID, RECID) Select
EXCHANGEITEMSPERASSEMBLY, EXCHANGEITEMGROUP, EXCHANGEITEMNUMBER,
```

```
ASSEMBLYITEMNUMBER, CORERETURNPERIOD, EXCHANGECONFIGID, ASSEMBLYCONFIGID,
MODIFIEDDATE, MODIFIEDTIME, MODIFIEDBY, MODIFIEDTRANSACTIONID, CREATEDDATE, CREATEDTIME, CREATE
DBY, CREATEDTRANSACTIONID, DATAAREAID, RECID from COREEXCHANGETABLE_temp");
```

```
permission = new SqlStatementExecutePermission(sqlStatement);
permission.assert();
//BP deviation documented
statement.executeUpdate(sqlStatement);
CodeAccessPermission::revertAssert();
}
```

14. Save the code.
15. Modify the `initPostSyncJobs` method to call the method created in steps 12–14. Highlight the `initPostSyncJobs` method in the left-most pane of the editor. The code is now displayed.
16. Locate the section in the code beginning with "**this.addSharedJob.**"
17. After this line of code, type the following.

```
this.addSharedJob(methodStr(ReleaseUpdateDB41_Administration,
restoreCustomTable), 'Restore data into custom table');
```

18. Save the code. At this point, the presynchronization should be able to run, and the custom method will back up the table. The synchronization process will prompt you about dropping the table, which can now happen safely, and postsynchronization should restore the data to the newly created table.

Troubleshooting the Detect code upgrade conflicts tool

This section contains information about issues you may encounter when you use the Detect code upgrade conflicts tool.

The Detect code upgrade conflicts tool freezes or crashes

Issue: When running the Detect code upgrade conflicts tool (**Tools > Development tools > Code upgrade > Detect code upgrade conflicts**) the AX client freezes or crashes.

Possible cause: The Detect code upgrade conflicts tool encounters a customization that cannot be analyzed. For example, if you've modified a form so that it hosts a query as a data source, this customization might cause the Detect upgrade conflicts tool to crash or freeze.

Solution: While the Detect code upgrade conflicts tool is running, the name of the customization that is being analyzed is displayed on the screen. Note the name of the customization being analyzed when the problem occurs.

Then:

1. Restart the client.
2. Export the customization that causes the client to hang or crash.
3. Delete the customization.
4. Run the Detect upgrade conflicts tool again.
5. Import the file from step 2.

Assume the customization has conflicts, and perform manual conflict resolution using the Compare tool.

Error messages during upgrade

This topic addresses some of the error messages that you may encounter during the upgrade process and offers possible causes and solutions for those messages.

New unique index name [INDEXNAME] may cause unique index violations

Issue: Data loss or unique key violation occurs.

Possible cause: Field sizes were changed (reduced) during synchronization and you performed the data upgrade before the code upgrade.

Solution: Restore a backup of your database and complete the code upgrade before the data upgrade.

Missing constraints error message

Issue: If you have previously upgraded from Microsoft Dynamics AX 3.0 to Microsoft Dynamics AX 4.0, the Microsoft Dynamics AX 2009 upgrade process fails when you start the Application Object Server (AOS) for the first time, or when you are completing the presynchronize, synchronize, or postsynchronize steps in the Upgrade Checklist. Also, the Microsoft Windows Application Event Log displays an error message that indicates that SQL statements are failing due to missing constraints, such as NULL is not expected, or the alter or create command on the table/index has failed.

Possible cause: The Microsoft Dynamics AX 4.0 database schema is missing default constraints on some tables. To identify if this is the cause, run the following command in your SQL Server Management Studio:

```
Sp_help<tablename>
```

For example, `sp_help ACCESSRIGHTSLIST`

The result set for a constraint type should be similar to the example below for the ACCESSRIGHTSLIST table.

```
Constraint_type
```

```
-----  
CHECK on column RECID  
DEFAULT on column ACCESSTYPE  
DEFAULT on column ACCESSTYPEFKYUSE  
DEFAULT on column DEL_CREATEDTIME  
DEFAULT on column CREATEDBY  
DEFAULT on column CREATEDDATETIME  
DEFAULT on column DOMAINID  
DEFAULT on column ELEMENTNAME  
DEFAULT on column GROUPLID  
DEFAULT on column DEL_MODIFIEDTIME  
DEFAULT on column MODIFIEDBY  
DEFAULT on column MODIFIEDDATETIME  
DEFAULT on column PARENTID  
DEFAULT on column RECORDTYPE  
DEFAULT on column RECVERSION  
DEFAULT on column ID  
PRIMARY KEY (clustered)
```

Also, check constraints for other tables such as BANKPARAMETERS, COMPANYDOMAINLIST, and so forth. If the result doesn't show the DEFAULT constraint defined on many of the columns, then your Microsoft Dynamics AX 4.0 database needs to be fixed before upgrading to Microsoft Dynamics AX 2009.

Solution: A hotfix is available to resolve this issue. To install the hotfix, see Knowledge Base article [953636](#). (A CustomerSource logon is required for access to this Knowledge Base article.)

Troubleshooting the Compare tool

This section contains information about issues you may encounter when you use the Compare tool.

Failing to compare current objects against old objects

If you are unable to compare current Microsoft Dynamics AX 2009 objects against Microsoft Dynamics AX 3.0 or Microsoft Dynamics AX 4.0 objects, the AXAPD.AOI file in the **Microsoft Dynamics AX\40\Application\Standard\Old** folder was not built correctly. To fix this issue, complete the following steps.

1. Stop the AOS service.
2. Verify that you have full control (Windows permissions) over the Standard\Old folder.
3. Recopy the prior version *.AOD files into the Standard\Old folder and be sure that the files are not marked as read only.
4. Rename the existing AXAPD.AOI file in the Standard\Old folder.
5. Start the AOS service. The AXAPD.AOI file in the Standard\Old folder should now be recreated, and this should contain the updated reference to all of the objects that existed in the prior Microsoft Dynamics AX version.

Upgrade Guide appendix

This appendix provides reference information that you may need when upgrading to Microsoft Dynamics AX 2009. This section contains the following topics.

- [Development access licenses](#)
- [Walkthrough: upgrading date and time table field pairs into UtcDateTime](#)
- [Prerequisite software installed by Setup](#)
- [AIF upgrade process](#)
- [Code upgrade: Examples](#)

Development access licenses

Development access in Microsoft Dynamics AX is divided into four different licenses:

- Base Package
- MorphX
- Web MorphX
- X++

Base Package

Base Package provides access to:

- Create or modify menus (user menus, main menu, and so on).
- Create or modify user forms.

MorphX

The MorphX development license provides access to the Data Dictionary node, where tables, fields, enums, and extended data types are maintained. The MorphX development license also lets you create or modify forms.

You can add, change, and delete forms, tables, extended data types, macros, menus, reports, jobs, and queries. Methods related to tables, such as forms, can be changed. For example, a change to a method allows you to change the standard information when you add new customers. Whereas, an example of a change in extended data types by using MorphX allows you to change the lengths of account numbers.

The Application Object Tree (AOT) displays the following nodes:

- Data Dictionary
- Forms
- Macros
- Reports
- Queries
- Jobs
- Menus
- Menu Items
- Resources
- System Documentation
- Application Developer Documentation
- Application documentation

Web MorphX

The Web MorphX Development Suite license provides access to the Microsoft Dynamics AX Web Development Framework.

The AOT displays the following nodes:

- Web Forms
- Web Reports
- Web Menus
- Web page
- Web Editor
- Style Sheet Editor

X++

The X++ development license provides full access to all classes in Microsoft Dynamics AX and the development code for the CUS layer. This lets you modify processes such as the invoice routine, or the posting process for a ledger journal. Having the development code for the CUS layer allows customers to place their modified application objects in the CUS layer, or the USR layer, or both.

You can add, change, and delete classes. For example, you could change the logic in an invoice update, which is defined in the class **SalesFormLetter**.

Walkthrough: upgrading date and time table field pairs into **UtcDateTime**

If custom **Date** or **Time** fields have been added to Microsoft Dynamics AX, you need to decide whether those fields should be merged into new **UtcDateTime** fields. **UtcDateTime** fields store date/time data in Coordinated Universal Time (UTC).

 **Note:**

This information is relevant when you must upgrade from any version earlier than Microsoft Dynamics AX 2009, to Microsoft Dynamics AX 2009 or later.

During upgrades it is common for a pair of fields, one **Date** and one **Time**, to be merged into a new **UtcDateTime** field. **UtcDateTime** enables every part of your organization to see the date/time value automatically adjusted for their local time zone.

If **Date** and **Time** fields must be merged into **UtcDateTime** fields, you must make certain preparations before you run the upgrade. The **UtcDateTime** preparations are only some of the many preparations that you will make before you run the Microsoft Dynamics AX 2009 upgrade process on your production environment.

This walkthrough illustrates the following tasks:

- Turning on **SysDeletedObjects41** in production
- Installing a test database
- Installing the new version in a test environment
- Turning on **SysDeletedObjects41** in Test
- Identifying date and time fields to upgrade
- Renaming fields to have the prefix DEL_
- Assigning **SysDeletedObjects41** to the renamed fields
- Adding replacement **UtcDateTime** fields
- Updating X++ code to use the new **UtcDateTime** fields
- Running the code upgrade on your test environment
- Overview of the next steps
- Adding a method to `ReleaseUpdateDB41_*`
- Registering your new method
- Summary of method relationships
- Rerunning the code upgrade on your test environment
- Handling special cases
- Running the upgrade on your production environment
- Consider other details

Prerequisites

This walkthrough is written for the administrator (or the lead developer) who must upgrade to Microsoft Dynamics AX 2009.

To complete this walkthrough, you will need:

- A temporary installation of Microsoft Dynamics AX 2009, its Application Object Server (AOS), and its client including the Application Object Tree (AOT)
- A license file that has access to the Microsoft Dynamics AX 2009 client
- A database server accessed by the Microsoft Dynamics AX 2009 (AOS)

Turning On SysDeletedObjects41 in Production

This procedure pertains to your production environment that you have not yet upgraded. You use the configuration key `SysDeletedObjects41` to tag items that you want to be able to easily delete after the upgrade completes.

In this procedure you will use the **Navigation Pane** to keep objects that you mark for deletion. Later you can clear this setting and the objects will be deleted.

To turn SysDeletedObjects41 on

1. In the **Navigation Pane**, click **Administration**, click **Setup**, click **System**, and then click **Configuration**.
2. In the **Configuration** form that appears, expand **Administration**, to reveal the **Keep update objects 41** node.
3. Select the **Keep update objects 41** the check box.

Installing a Test Database

In this procedure you will create the test database environment that is required to support the preparation steps for the eventual upgrade of your production system. After your production system is upgraded, this test environment can be discarded.

To install a test database

1. Create an empty database on a test database server.
2. As an option, load a copy of your production database into the empty database.

If you choose an empty database, the tables and other structures will be created when you connect the new Microsoft Dynamics AX to it. The synchronization process will create the structures.

Installing the New Version in a Test Environment

In this procedure you must install Microsoft Dynamics AX 2009 (or a newer version of Microsoft Dynamics AX) that you intend to upgrade your production system to. This installation is to a test environment that can later be discarded.

Turning On SysDeletedObjects41 in Test

This procedure pertains to the new Microsoft Dynamics AX test environment that your production environment will be upgraded to.

Next, you will use the **Navigation Pane** to keep objects that you mark for deletion. Later you can clear this setting and the objects will be deleted.

The technique is that same as was previously described for the production environment.

Identifying Date and Time Fields to Upgrade

In this procedure you will scan all the custom **Date** and **Time** fields that have been added to your installation. You must decide which of those fields will be upgraded to the new **UtcDateTime** type.

Make a list of all the custom **Date** and **Time** fields that must be upgraded.

To decide whether to upgrade a date or time field to utcdatetime

1. Identify pairs of Date and Time fields that are used together (in the same table) to store a precise moment. In such pairs, both fields might share the same name distinguished only by their suffix of **Date** or **Time**.

 **Note:**

An example could be a field pair of **YourMeetingDate** and **YourMeetingTime**. A new field **YourMeetingDateTime** would be a good replacement for this pair.

2. Identify **Date** fields that are not paired with a **Time** field, but that do imply a local time. Some legal or contractual obligations of due dates might be examples where a time such as (midnight or 00:00) is implied.
3. Identify **Time** fields that are not paired with a **Date** field, but that would be more useful if represented in UTC. Such time fields are rare.

Renaming Fields to Have the Prefix DEL_

In this procedure you will rename the Date and Time fields that you identified earlier. Each must be given the four character prefix DEL_.

To rename a field in a table

1. In the AOT, expand the **Data Dictionary** node, and then expand **Tables**.
2. Expand the specific table that contains a field that you want to rename, and then expand the **Fields** node.
3. Right-click the field that you want to rename, and then select **Properties**.
4. In the **Properties** window, click the text box for the **Name**.
5. Type the DEL_ prefix to modify the name, and then press ENTER.

Assigning SysDeletedObjects41 to the Renamed Fields

In this procedure you will assign the `SysDeletedObjects41` configuration key to the **Date** and **Time** fields that you want to upgrade to **UtcDateTime**.

Each version of Microsoft Dynamics AX has a configuration key with a similar name, except the version number at the end differs. Here the version suffix **41** refers to Microsoft Dynamics AX 2009.

By assigning **SysDeletedObjects41** to a field that you renamed with the DEL_ prefix, you track all the fields that will be obsolete after the production environment is upgraded. The configuration assignment also makes it easy to delete all obsolete fields at one time.

To assign the SysDeletedObjects41 configuration key to fields

1. In the AOT, expand **Data Dictionary**, expand your table, and then expand **Fields**.
2. Right-click the field that has the DEL_ prefix, and then select **Properties**.
3. In the **Properties** window, click the drop-down list for the **ConfigurationKey** property, and then select **SysDeletedObjects41**.

Adding Replacement UtcDateTime Fields

In this procedure you will add a **UtcDateTime** field for every obsolete **Date** field, **Time** field, or field pair that you want to replace during upgrade.

To add replacement utcDateTime fields

1. In the AOT, expand **Data Dictionary**, and then expand **Tables**.
2. Expand the specific table that needs a new **UtcDateTime** field.
3. Right-click the **Fields** node, click **New**, and then select **UtcDateTime**.
4. Right-click the new field node, and then select **Properties**.
5. In the **Properties** window, change the **Name**, and any other properties as needed.

Updating X++ Code to Use the New UtcDateTime Fields

In this procedure you will scan your X++ code to find all references to the **Date** or **Time** fields that you are replacing.

One way to find the references is to recompile all your X++ code. The compile errors that occur from the earlier field renames indicate the references that must be updated.

You must manually determine the X++ code change that is needed in each case. The new code must reference the new **UtcDateTime** field.

To update X++ code to use the new utcDateTime fields

1. Find the references to the old DEL_ fields by recompiling all your X++ code.
2. Research each error to determine how to fix the code by referencing the new **UtcDateTime** field.
3. Recompile your X++ code change.

UtcDateTime Functions

The following table lists some of the functions that can manipulate **UtcDateTime** values.

Function	Description
DateTimeUtil class	This has many methods for manipulating <code>utcDateTime</code> values.
Global class	The <code>Global</code> class has some methods for manipulating <code>utcDateTime</code> values, including the following: <ul style="list-style-type: none"> • <code>clrSystemDateTime2UtcDateTime</code> • <code>dateToBeginUtcDateTime</code>

Function	Description
	<ul style="list-style-type: none"> • <code>dateToEndUtcDateTime</code> • <code>utcDateTime2SystemDateTime</code> • <code>utcDateTimeNull</code> <p>Note <code>utcDateTime2SystemDateTime</code> refers to the .NET Framework class <code>System.DateTime</code>. This function is equivalent to <code>clrSystemDateTime2UtcDateTime</code> (where the prefix <code>clr</code> refers to the common language runtime of the .NET Framework).</p>
Functions	<p>The system has X++ functions for manipulating <code>utcDateTime</code> values, including the following:</p> <ul style="list-style-type: none"> • <code>datetime2Str</code> • <code>str2Datetime</code>

Running the Code Upgrade on Your Test Environment

In this procedure you will run the code upgrade function on your test environment from the Microsoft Dynamics AX 2009 client. This produces an .AOD file that is used when you perform the upgrade on your production system.

This procedure can be run many times. You might rerun this procedure after each change, instead of waiting until after you make all changes. Repeated runs after smaller sets of changes might make problems easier to diagnose or fix.

For more information, see the documentation for the Microsoft Dynamics AX upgrade.

To run the code upgrade

1. In the client's menu, click **Tools**, click **Development tools**, and then click **Code upgrade**.
2. You will see four upgrade related actions. Run each of these actions, one at a time, in the sequence that they appear in the menu.
 - a. **Detect code upgrade conflicts**
 - b. **Compare layers**
 - c. **Estimation report**
 - d. **Parameters**

Overview of the Next Steps

In the next procedures you will write the X++ code necessary for the upgrade. You will add a method to a class that is provided by Microsoft Dynamics AX 2009. Your method will not be called during the phase of manual preparations. Instead you will register your new method, which will cause the production upgrade process to start your new method.

Adding a Method to ReleaseUpdateDB41_*

In this procedure you will add a method to one of the classes that is named by the `ReleaseUpdateDB41_*` pattern. There are several such classes, each intended for a particular module. Each table is assigned a configuration key. The configuration key indicates the module that the table belongs to. The trailing asterisk (*) in the name pattern is a placeholder for a module identifier. You should add your method to the class that corresponds to the module of the affected table. If you are unsure of which class to use, you can use the `ReleaseUpdateDB41_Basic` class.

By convention and for consistency, the method name should begin with `updateToDateTime_*`. Include your name as the suffix for easy identification.

Note:

Notice that some method names that look the same at first glance actually begin with different words, either `update` or `upgrade`. They are not the same method.

The method that you add calls a system-provided method named `upgradeToDateTime`. The `upgradeToDateTime` method performs an SQL insert into the `DEL_SysUpgradeTimeZone` table. Your method can make several such calls.

To add a method to ReleaseUpdateDB41_Basic

1. In the AOT, expand the **Classes** node, and then expand **ReleaseUpdateDB41_Basic**.
2. Right-click **Methods**, and then select **New Method**.
3. Add a new instance method that resembles the following example. The name of your method can be anything unique. Your new method must call `upgradeToDateTime`, as shown in the example.

 **Note:**

Pass zero (0) as the value for the `fieldId` of your time column if your date column is not paired with a time column.

 **Note:**

The final parameter shown is optional, and the default is the value shown (classnum for `ReleaseUpdateDB_TimezoneUpgrade`).

```
void updateToDateTime_Jane
{
    ReleaseUpdateDB41_Basic ::upgradeToDateTime
    (
        tableNum( YourActivitiesTable )
        ,fieldNum( YourActivitiesTable ,YourMeetingStartDateTime )
        ,fieldNum( YourActivitiesTable ,DEL_YourMeetingStartDate )
        ,fieldNum( YourActivitiesTable ,DEL_YourMeetingStartTime )
        ,classNum( ReleaseUpdateDB_TimezoneUpgrade )
    );
}
```

4. In the code editor window, click the save icon, and then close the window.

Class ReleaseUpdateDB_TimezoneUpgrade

In the previous code example, the `ReleaseUpdateDB_TimezoneUpgrade` class is given as a parameter. The class has a method named `runSQLCode_UpdateDateToUTCDateTime`, which applies to Microsoft SQL Server database installations. The class has a very similar corresponding method named `runORACode_UpdateDateToUTCDateTime` for Oracle database installations. These methods update your new **UtcDateTime** type field by assigning a UTC time value to it (and a time zone to a hidden related field). The UTC value is derived from your obsolete **Date** and **Time** field pair, and from the local time zone of your AOS.

The example call to `upgradeToDateTime` leads to a call of one of the two methods.

Consider the Affected Module

The `ReleaseUpdateDB41_Basic` class is not specific to any one module. There are several classes that resemble this class, but that are intended to hold code that is specific to a particular module. One such class is `ReleaseUpdateDB41_smm`. The others classes follow the naming convention of `ReleaseUpdateDB41_*`. Your new method can be added to the `*_Basic` class. However, you are encouraged to add it to the similar class named for the module that contains the items that your method is upgrading.

Registering Your New Method

In this procedure you will write X++ code to register your new method for the eventual production upgrade process. In the example, you call the method `addSharedJob`. The parameters will be the name of the new method that you added, and the class on which you added your method.

To register your new method

1. In the AOT, expand the **Classes**, expand **ReleaseUpdateDB41_Basic**, and then expand **Methods**.
2. Right-click **initPreSyncJobs**, and then select **Edit**.
3. In the editor window, inside the method, write the call that is shown in the following example.

```

this .addSharedJob
(
    methodStr( ReleaseUpdateDB41_Basic
        ,updateToDateTime_Jane
    )
    ,"Handles new fields like YourMeetingStartDateTime."
)

```

4. Click the save icon, and close the editor.

For more information, see [How to write upgrade scripts](#).

Summary of Method Relationships

The following table provides a summary of the relationships between the classes and their methods, and other objects that are discussed in the previous sections.

Initiator	Operand	Relationship
DEL_YourMeetingStartDate, DEL_YourMeetingStartTime (table fields)	YourMeetingStartDateTime	Data: Data is read from DEL_YourMeetingStartDate and DEL_YourMeetingStartTime. Time zone information is applied, and then the data is stored in the new field YourMeetingStartDateTime.
.updateToDateTime_You method (on class ReleaseUpdateDB41_Basic)	ReleaseUpdateDB41_Basic::upgr adeToDateTime	Call: updateToDateTime_You calls upgradeToDateTime.
ReleaseUpdateDB41_Basic:: upgradeToDateTime	ReleaseUpdateDB_TimezoneUpgra de /* .runSQLCode_UpdateDateToUTCda teTime */	Parameter: The operand class is a parameter into upgradeToDateTime. An important method on the operand class is runSQLCode_UpdateDateToUTCDateTime. When you provide this operand class as a parameter, it is this method (or its Oracle equivalent) that is ultimately run.
ReleaseUpdateDB41_Basic:: upgradeToDateTime	DEL_SysUpgradeTimeZone (table)	SQL insert: The method performs an insert into the operand table. This table is read by the processes that upgrade your production environment.
ReleaseUpdateDB41_Basic.i nitPreSyncJobs	ReleaseUpdateDB41_Basic.addSh aredJob	Call: initPreSyncJobs calls addSharedJob.
ReleaseUpdateDB41_Basic.a ddSharedJob	updateToDateTime_You method (on class ReleaseUpdateDB41_Basic or on a similar class that is more specific to a module)	Parameter: updateToDateTime_You (a method name) is a parameter into addSharedJob.

Rerunning the Code Upgrade on Your Test Environment

This procedure was described previously.

Handling Special Cases

It is possible that your customized **Date** and **Time** fields might be designed to work with specialized X++ code. Special application rules might be involved.

The following table describes how to handle special cases.

Special case	Description
Extend the class <code>ReleaseUpdateDB_TimezoneUpgrade</code>	<p>You can extend this class to provide the specialized logic that is required to handle your special case Date and Time fields.</p> <p>The class <code>ReleaseUpdateDB_TimezoneActivityUpgrade</code> is a sample that you can mimic. This child class contains an override of the <code>runSQLCode_UpdateDateToUTCDateTime</code> method. It also contains an override of the corresponding Oracle method. Many installations would override only one of those two methods.</p>
Add your method to a class similar to but different from <code>ReleaseUpdateDB41_Basic</code>	<p>The AOT shows there are several classes named by the pattern <code>ReleaseUpdateDB41_*</code>. Some correspond to different modules in Microsoft Dynamics AX. You can add your method to one of those classes</p>
Bypass the upgrade of application Date and Time fields that ship with Microsoft Dynamics AX	<p>It is possible that external systems access a Date and Time field pair. It might not yet be plausible to modify the external system to switch to a replacement UtcDateTime field.</p> <p>You can bypass the upgrade of a particular Date and Time pair by commenting out the appropriate calls in one of the <code>ReleaseUpdateDB41_*</code> classes.</p>
Administrator has added a table for synchronization at application startup	<p>You may have added a call to the <code>syncTable</code> method in <code>Application.syncApplTables</code>. This practice is not recommended.</p> <p>Tables that you have added to <code>syncApplTables</code> are not properly upgraded for UtcDateTime. One solution is to comment out the call in <code>syncApplTables</code> for the table that you added. Another solution is to manually upgrade your table after the automated upgrade has completed.</p>

Running the Upgrade on Your Production Environment

It can take a while to prepare the test environment before your organization is ready to upgrade its production environment. The **UtcDateTime** portion of the overall upgrade preparation is relatively small.

Now you are ready to upgrade your production environment. The following table describes the two phases of upgrade that directly relate to the **UtcDateTime** upgrades.

Phase of upgrade	Description
Pre-synchronization	<p>The <code>initPreSyncJobs</code> method is run, as the last step of pre-synchronization. Upgrade does no other processing while this step runs. The processing during this step takes table locks in the SQL database.</p> <p>This adds your method to a list of methods that the system will run during the post-synchronization phase.</p> <p> Note:</p> <p>Any executions of the method <code>ReleaseUpgradeDB41_Basic::upgradeToDateTime</code> after post-synchronization has begun are ignored.</p>
Post-synchronization	<p>The <code>runSQLCode_UpdateDateToUTCDateTime</code> method is run, as the first step of post-synchronization. This updates the UTC values for your new UtcDateTime column.</p>

Consider Other Details

Additional details about the **UtcDateTime** upgrade are as follows:

- During the upgrade checklist, you are prompted to enter the time zone that you want set for your AOS. Select the time zone that most of your Date and Time data matches.

 **Caution:**

- After the upgrade begins to use your selected time zone, there is no way to change to a different time zone. Be sure to select the correct time zone. You must make a backup copy of your database before beginning the upgrade.
- Null **Date** values will still be null after the upgrade to **UtcDateTime**. For these data types, their lowest possible value is treated as null.

- When upgrade occurs for a **Date** field that is not paired with a **Time** field, the time part of the **UtcDateTime** value is set to 12:00 (at the midpoint of the day). Next, the time zone offset is applied, which changes the 12:00 time setting. The expectation is that the modified time is still within the same day.
- If the upgrade stops in mid-process, the upgrade to **UtcDateTime** will complete when the upgrade is restarted. The system tracks status during each step of the upgrade.

Prerequisite software installed by Setup

If you are installing a component that requires one or more of the following prerequisite software components to be installed, Setup installs and configures the software for you on the local computer.

The following table lists prerequisites and the components that require them.

Prerequisite	Component it will be installed for
Internet Information Services (IIS)	Workflow Role Centers and Enterprise Portal framework AIF Web services
Microsoft Message Queuing (MSMQ)	Synchronization service (Project Server 2007)
Microsoft Windows SharePoint Services 3.0, Service Pack 1	Role Centers and Enterprise Portal framework
Microsoft Visual C++ 2005 Redistributable Package	AOS Client .NET Business Connector
Microsoft Visual C++ 2008 Redistributable Package	AOS Client .NET Business Connector
Microsoft SQL Server 2005 ADOMD.NET	Role Centers and Enterprise Portal framework
SQL Server 2005 Native Client	AOS
DHTML Editing Control for Applications Redistributable Package	Client

If you are not the system administrator, consult a system administrator before installation. You cannot use Microsoft Dynamics AX Setup to uninstall this software later.

AIF upgrade process

Before you upgrade

The message format has changed in Microsoft Dynamics AX 2009. Therefore, all AIF messages must be completely processed before you run the upgrade. Make sure that all messages have been processed from the following locations:

- All inbound message locations such as file system directories, Message Queuing queues, or any location where AIF retrieves inbound messages.
- The queue manager. All messages in the queue manager are unprocessed and must be deleted. To navigate to the queue manager, click **Basic > Periodic > Application Integration Framework > Queue manager**.

Code upgrade

The AIF code upgrade step upgrades existing `Axd<Document>` classes and methods. The upgrade process does not change any of the existing `Axd<Document>` classes but rather it creates corresponding services and artifacts. There are three types of classes that the code upgrade will process as shown in the following table.

Category	Description
<code>Axd<Document></code> classes that ship with Microsoft Dynamics AX	Service classes and artifacts are generated for all <code>Axd<Document></code> classes that ship with Microsoft Dynamics AX. If you have no custom classes or did not implement any custom actions, you do not need to take any further action. For more information about the standard services, see the Standard <code>Axd<Document></code> classes section below.
Custom <code>Axd<Document></code> classes	Service classes and artifacts are generated for all custom <code>Axd<Document></code> classes. The results of the upgrade process differ depending on how these classes were coded. For more information about custom services, see the Custom <code>Axd<Document></code> classes section below.
Custom classes that implement the <code>AifServicable</code> interface	The upgrade process generates a warning message but will not generate any service artifacts for these classes.

Standard Axd <Document> classes

Service classes and data objects are generated for each existing *Axd<Document>* class as shown in the following table (for the *AxdSalesOrder* class). The classes are found in the AOT by navigating to **AOT > Classes**. The AOT service node for each service is found by navigating to **AOT > Services**.

Generated artifact	Description	Example
Document class	The root data object.	<i>SalesSalesOrder</i>
Data objects	A data object class is created for each data source in the query. These objects are children of the document class.	<i>SalesSalesOrder_SalesTable</i> , <i>SalesSalesOrder_DocuRefHeader</i> , <i>SalesSalesOrder_DocuRefLine</i> , <i>SalesSalesOrder_InventDim</i> , <i>SalesSalesOrder_MarkupTransHeader</i> , <i>SalesSalesOrder_MarkupTransLine</i> , <i>SalesSalesOrder_SalesLine</i>
Service class	A service class is generated for each <i>Axd<Document></i> class. For each action supported by the <i>Axd<Document></i> class, an operation is created on the service class.	<i>SalesSalesOrderService</i>
AOT service node	An entry in the AOT Services node where you can view and modify the properties of the service and its operations.	<i>SalesSalesOrderService</i>

Action mapping for standard methods

Microsoft Dynamics AX 2009 supports a new set of actions, and the document services have operations that correspond to these actions. In some cases, the new actions have been designed to contain the functionality for two actions in the previous version.

Standard actions for the `Axd<Document>` classes that ship with Microsoft Dynamics AX will be upgraded as part of the code upgrade; no additional work is necessary. The following table lists the new actions and how actions from the previous version are mapped.

Previous version action	Mapped to service operation
createList create	create
deleteList	delete
findList	find
findEntityKeyList	findKeys
readList read	read
updateList update	update

In Microsoft Dynamics AX 4.0, not all `Axd<Document>` classes that were included with the product supported all actions. If you customized a standard `Axd<Document>` class and enabled support of a standard action, the code upgrade adds a method to the service class with a name according to the following naming convention: action name + document class name.

For example, the `AxdASN` document class does not support the `create` action. If you added support for the `create` action to `AxdASN`, the code upgrade process adds a method called `createASN` to the new service class. You can find the action ID in the `getActionList` method of the `Axd<Document>` class. The following line of code in this method creates the full action name:

```
aifActionInfo.parmActionId(strfmt('%1%2',
    methodstr(AxdBase, findEntityKeyList), this.getName()));
```

Action mapping for custom methods

If you have implemented custom actions on a standard `Axd<Document>` class, the code upgrade adds a method to the service class with a name according to the following naming convention: action name + document class name. For example, if you added a method called `test` to the `AxdASN` document class, the code upgrade process adds a method called `testAxdASN` to the new service class.

Custom Axd <Document> classes

Custom `Axd<Document>` classes are also upgraded during the AIF upgrade process. The services and artifacts that are generated are the same as those for the standard `Axd<Document>` classes as shown in the following table. However, the upgrade process will handle methods (actions) on custom document classes differently depending on how they were coded.

Generated artifact	Description	Example
Document class	The root data object.	Test
Data objects	A data object class is created for each data source in the query. These objects are children of the document class.	Test_Table1
Service class	A service class is generated for each <code>Axd<Document></code> class. For each action supported by the <code>Axd<Document></code> class, an operation is created on the service class.	TestService
AOT service node	An entry in the AOT Services node where you can view and modify the properties of the service and its operations.	TestService

Action mapping

Standard actions on a custom `Axd<Document>` class are mapped to the new actions supported in Microsoft Dynamics AX 2009 as shown in the following table. The new actions are implemented as methods in the service class.

Previous version action	Mapped to action
<code>createList</code>	<code>create</code>
<code>deleteList</code>	<code>delete</code>
<code>findList</code>	<code>find</code>
<code>findEntityKeyList</code>	<code>findKeys</code>
<code>readList</code>	<code>read</code>
<code>updateList</code>	<code>update</code>

If the actions `create`, `read`, `update`, or `delete` are supported by a custom `Axd<Document>` class, these actions are added to the new service class with the following naming convention: action name + document class name. For example, if you have a custom document class named `AxdTest` with an action called `create`, the upgrade process adds an operation called `createAxdTest` to the new service class.

If you have implemented a custom action on a custom `Axd<Document>` class, the upgrade process adds a method to the new service class with the following naming convention: action name + document class name. For example, if you have an `Axd<Document>` class named `AxdTest` and a custom method called `createTest`, the upgrade process adds an operation called `createTestAxdTest` to the new service class.

Artifact naming conventions

The following table lists the naming conventions used when the upgrade process generates services and artifacts. These naming conventions are applied to artifacts generated for the standard `Axd<Document>` classes as well as custom document classes, except for the prefix; no prefix is applied to services generated from custom document classes.

Artifact type	Name description	Name generation rules	Example
Document object	Name of the X++ class for the root data object.	<i><Prefix></i> + <i><Document Name></i> + "Service"	SalesSalesOrder
Data objects	Name of the X++ class for the child data objects of the document class.	<i><Root data object name></i> + "_" + <i><Query data source Name></i>	SalesSalesOrder_SalesTable, SalesSalesOrder_DocuRefHeader, SalesSalesOrder_DocuRefLine, SalesSalesOrder_InventDim, SalesSalesOrder_MarkupTransHeader, SalesSalesOrder_MarkupTransLine, SalesSalesOrder_SalesLine
Service class	Name of the AIF service class.	<i><Prefix></i> + <i><Document Name></i> + "Service"	SalesSalesOrderService
AOT service node	Name of the AOT node for the service.	<i><Prefix></i> + <i><Document Name></i> + "Service"	SalesSalesOrderService
Service external name	Name of the service published to WCF.	<i><Document Name></i> + "Service"	SalesOrderService

 **Note:**

When custom document classes are upgraded, or when creating your own document classes, you do not need to use a prefix.

Data upgrade

The data upgrade process upgrades AIF data in the Microsoft Dynamics AX database. The tables listed in this section are new or upgraded in Microsoft Dynamics AX 2009. Any tables or fields that have a name prefaced with "DEL_" are tables used during the data upgrade and are for internal use only.

 **Note:**

The AIF code upgrade process must be run before the data upgrade process. This means that the **Upgrade Aif Code** step in the upgrade checklist must have been run previously. To verify that the code upgrade step has been run, check the **AOT > Services** node and you will see a list of AIF services. This signifies that the code upgrade process has already been run.

AifGlobalSettings

The field `ResponseCacheLifetime` has been added to the table and is set to 24 (hours).

AifService

This is a new table with the following fields:

- `ClassId`
- `Name`
- `Description`
- `Enabled`
- `ErrorState`
- `ErrorMessage`
- `Namespace`
- `ExternalName`

The upgrade process adds a service to this table for each `AxD<Document>` class. For each unique document `ClassId` in the `AifAction` table the upgrade process will:

1. Locate the service node under **AOT > Services**.
2. Add a record to the `AifService` table and populate it with data from the AOT.

3. If the upgrade process encounters any errors while processing service actions in the `AifAction` table, these errors are concatenated and saved in the `ErrorMessage` field for each service.
4. Sets the `Enabled` field to `true` if the service has an action enabled in the `AifAction` table.

**Note:**

Only `Axd<Document>` classes that have registered actions (the action exists in the `AifAction` table) will be upgraded.

AifAction

The following fields have been removed:

- `ActionType`
- `WebSvcOutOfSync`
- `ActionEnabled`
- `WebMethodEnabled`
- `InfoLogData`
- `ErrorState`

The following fields have been added:

- `EnforceIdempotency`
- `ServiceName`

The data in the following fields has changed:

- `ActionId`

In Microsoft Dynamics AX 2009, the actions `create`, `read`, `update`, and `delete` are not supported and can no longer be used. However, the functionality in these actions has been replaced by the new `create`, `read`, `update`, and `delete` service operations which can take parameters that contain multiple entity key values or multiple documents.

For each action where the `Axd<Document>` class has a matching service class in the `AifService` table, the upgrade process does the following:

1. Verifies that the action can be upgraded. For documents that ship with Microsoft Dynamics AX, the methods `create`, `read`, `update`, and `delete` are not upgraded because these methods work with a single document or entity key. If the document class action cannot be upgraded, it is skipped and the upgrade process moves to the next action for the document class.

- If the action can be upgraded, the upgrade process gets the new method name as shown in the following table.

Previous version action	Mapped to method
createList	create
deleteList	delete
findList	find
findEntityKeyList	find
readList	read
updateList	update

- The record in the `AifAction` table is updated as follows.

Field	Value
ActionId	serviceClassName.Method
Name	serviceExternalName.serviceNodeOperationName
ServiceName	Service name in the AOT
ClassId	Service ClassId
ExternalAction	Service node operation name
MethodName	Service class method name
EnableIdempotence	Set depending on the value in the service reference node

After the data upgrade process completes, you may see actions in the `AifAction` table that are disabled (the field `ActionEnabled` is set to `false`). This can occur because:

- The record is for a `create`, `read`, `update`, or `delete` action in the previous version. These actions will be disabled in the table. You do not need to take any further action.
- If an unexpected error occurs during the upgrade process and an action cannot be upgraded, it will be disabled in the `AifAction` table.

AifDocumentSchemaTable

The `DocumentClassId` is updated to the `ClassId` of the data object.

AifDocumentFieldTable

The `DocumentClassId` is updated to the `ClassId` of the data object.

AifEndpoint

The `PreventLoopbackSend` and `PropagateErrors` fields are set to `false`.

AifResponse

The following fields have been added:

- `CreatedDateTime`

All records are deleted during the data upgrade because the old message format is not forward compatible.

AifEndpointActionValueMap

The `ActionId` is updated to the new `ActionId` for actions that are successfully upgraded.

AifDataPolicy

The `ActionId` is updated to the new `ActionId` for actions that are successfully upgraded.

AifPipeline

The following fields have been added:

- `Direction`
- `EndpointId`

A pipeline can be inbound or outbound depending on the direction of the document flow which is determined by the action type of the endpoint action policy. For any records in `AifEndpointActionPolicy` that have a corresponding record in `AifPipeline`, the `AifPipeline.Direction` field is updated as follows:

- Inbound if the action type is `ReceiveDocument`, `ReceiveDocumentList`, `ProcessEntity`, or `ProcessEntityList`.
- Outbound if the action type is any other value.

The data upgrade process also updates the `AifPipeline.ActionId` field with the new `ActionId` and updates the `AifPipeline.EndpointId` with the `EndpointId` from the `AifEndpointActionPolicy` table.

AifEndpointActionPolicy

The following fields have been added:

- InboundPipelineId
- OutboundPipelineId
- IsDefault
- AutoErrorResponse

The follow columns have been removed:

- XsltFilePath
- SchemaRecId
- PipelineId
- DataPolicySynchronized

The `ActionId` field is updated with the new `ActionId`, and the `InboundPipelineId` and `OutboundPipelineId` fields are updated if there are any pipelines associated with the endpoint action policy.

AifEndpointActionParameterSchema

This is a new table with the following fields:

- EndpointId
- ActionId
- ParameterClassId
- SchemaRecId
- DataPolicySynchronized

The data upgrade process creates a record in this table for each record in the `AifEndpointActionPolicy` table. The fields `EndpointId`, `ActionId`, `SchemaRecId`, and `DataPolicySynchronized` are all updated from the `AifEndpointActionPolicy` table. The `ParameterClassId` is set to the ID of the serializable data object.

AifDataPolicyXPath

The following fields have been added:

- DocumentClassId

The `ActionId` is updated to the new `ActionId` for actions that have been upgraded. The `DocumentClassId` is set to the `SerializableClassId`.

AifXmlTransformConfig

The following fields have been added:

- ParameterName

The data upgrade process sets the `ParameterName` field to the parameter name of the service operation.

AifValueSubstitutionComponentConfig

This is a new table with the following fields:

- PipelineComponentRecId
- ParameterName

The data upgrade process adds a record in this table for each record in the `AifValueSubstitutionConfig` table. The `ParameterName` field is set to the parameter name of the service operation. The `PipelineComponentRecId` field is set to the `PipelineComponentRecId` field value in the `AifValueSubstitutionConfig` table.

Data that is not upgraded

Data that is configured to use an `ActionId` that has not been upgraded will be moved according to the following table.

Table name	Moved to table name
AifAction	DEL_AifAction
AifDataPolicy	DEL_AifDataPolicy
AifDataPolicyXPath	DEL_AifDataPolicyXPath
AifEndpointActionPolicy	DEL_AifEndpointActionPolicy
AifEndpointActionValueMap	DEL_AifEndpointActionValueMap
AifPipeline	DEL_AifPipeline
AifPipelineComponent	DEL_AifPipelineComponent
AifSchemaStore	DEL_AifSchemaStore
AifValueSubstitutionConfig	DEL_AifValueSubstitutionConfig
AifWebsites	DEL_AifWebsites

The `AifWebsites` table is not upgraded and any Websites created in a previous version will be contained in the `DEL_AifWebsites` table. In Microsoft Dynamics AX 2009, the administrator must add the Websites manually. For more information, see "Configure Web sites for document exchange" in the [Server and Database Administration Help](#).

Schema changes

In Microsoft Dynamics AX 2009, the message and document schemas have changed. After upgrading, you will need to refresh the AIF services by following these steps:

1. Click **Basic > Setup > Application Integration Framework > Services**.
2. Click **Refresh**.

There are various schemas that define the format and content of messages in AIF as shown in the following table.

Schema	Description
Message schemas	Define the format and content of the message metadata. This includes elements such as the envelope, the header, query criteria, entity key lists, and so on. The message schemas are included with Microsoft Dynamics AX. For more information, see "Message XSDs" and "Schemas" in the Microsoft Dynamics AX Developer Help.
Document schema	Defines the format and content of the business data in a message. This schema contains all the fields that can participate in a data exchange. Any message that comes into AIF must validate against the document schema.
Endpoint schema	Further restricts the document schema fields that can participate in a data exchange for a particular endpoint. The endpoint schema is defined by the data policy that you set for each endpoint action.

Due to these schema changes, any external systems that create inbound AIF messages must be modified to correctly format messages according to the new schemas. Any external systems that rely on data in outbound AIF messages must be modified to handle any changes in the outbound data.

Message schemas

Any message schema elements in inbound messages must validate against the new message schemas located in <Microsoft Dynamics AX Installation Directory>Application\Share\Include. For more information see, "AIF Messages" and "Message XSDs" in the Microsoft Dynamics AX Developer Help.

Document schemas

The document schemas have changed and any inbound messages that contain business data must be modified to validate against the new schemas. Also, outbound messages containing business data will be formatted to validate against the new schemas. For more information, see "Schemas" and "Document Schema Overview" in the Microsoft Dynamics AX Developer Help.

As a result of the schema changes, you may need to update the XSLTs that are used in pipeline transformations. After you have updated the XSLTs, you can reimport them by clicking **Basic > Setup > Application Integration Framework > XSLT repository**. You may also need to update your value lookups which you can do by clicking **Basic > Setup > Application Integration Framework > Value lookup**.

Endpoint schemas

The endpoint schemas are also known as endpoint action data policies. After upgrading, any document schema changes will not be reflected in existing endpoint data policies. New elements in the document schema will be turned off in the data policy, unless they are also required elements. The AIF administrator must modify existing endpoint action policies and add any new elements to the data policy if necessary. For more information, see "Configure endpoint action data policies" in the [Server and Database Administration Help](#).

After you upgrade

Refresh the AIF document services

In order to call the AIF document services, you must refresh them.

1. Click **Basic > Setup > Application Integration Framework > Services**.
2. Click **Refresh** to load the services. There may be a delay while the services load.

For more information, see "Configure services" in the [Server and Database Administration Help](#).

User permissions

Be sure that all source endpoint users are granted view permissions to AIF tables.

Code upgrade: Examples

Two code upgrade scenarios are described in this topic:

- Scenario 1: Upgrading the CustTable table
Code modifications typically involve the addition of one or more fields to an existing table.
- Scenario 2: Upgrading the VendTable form
Another typical modification is to add a button to an existing form.

Table and form objects are discussed in separate examples because they are treated differently during an upgrade.

Forms (as well as reports) are designed as a single entity. This means that any modification to a form, whether the addition of a button or a change in a method, requires the whole form to be updated.

All other types of application objects (tables, classes, menus, and so on) consist of a number of components, each of them shown as a node in the Application Object Tree (AOT). If only one table method has been modified, only that method is considered during the upgrade process.

The field names and numbers used in the following scenarios are fictional examples that do not necessarily correspond to any real-life situation. The examples use a SYS layer and a USR layer. The modifications might also be in another layer, or even in several layers.

Scenario 1: Upgrading the CustTable table

In this scenario, Microsoft Dynamics AX has been installed as an upgrade to an existing Microsoft Dynamics AX 3.0 installation.

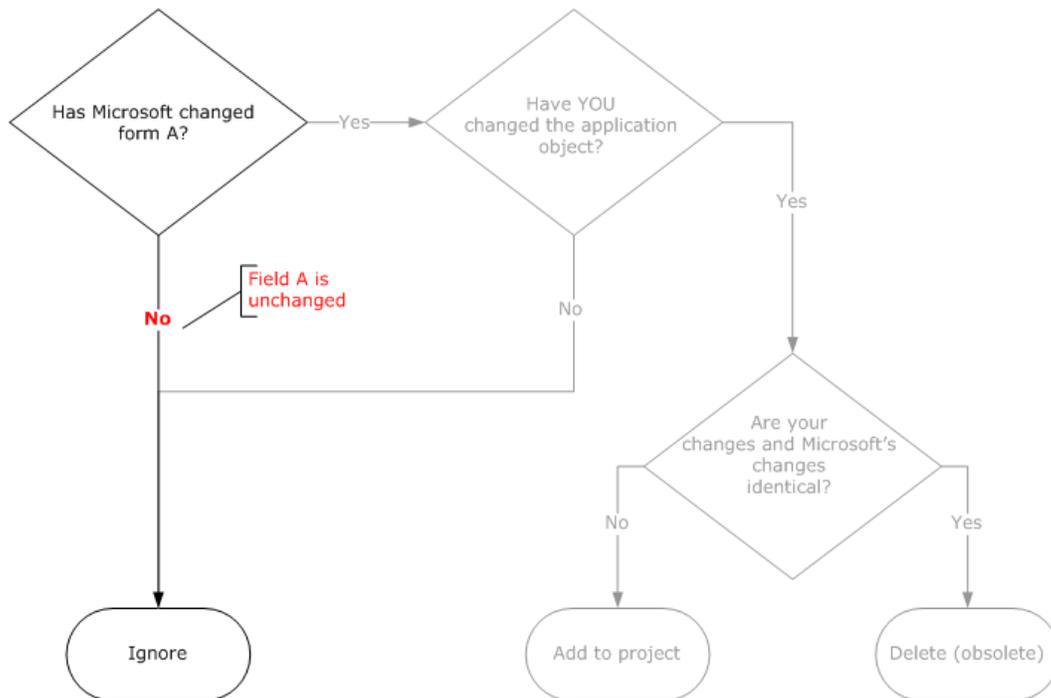
In the SYS layer, CustTable has two fields, A and B.

In the SYS layer, CustTable has 4 fields, A, B, C and D. That is, two fields have been added in the new version.

In the USR layer, CustTable has one field, CompanyLogo.

In the USR layer, CustTable has one field, CompanyLogo. The USR layers in the new and old versions are identical.

When the Create upgrade project algorithm is applied to each field in the CustTable table, fields A and B will be ignored since both existed in Microsoft Dynamics AX 3.0. No upgrade is necessary.



The same applies for the new fields, C and D: there are no changes to the fields as compared to the previous version, since the fields did not exist then. Therefore, no upgrade is required.

Similarly, the field CompanyLogo is not relevant for the upgrade. There are no conflicts with the previous version.

The final result of the Create upgrade project for the CustTable table is that the table is not included in the upgrade project since there are no conflicts to resolve. In Microsoft Dynamics AX, CustTable has five fields, four in the SYS layer and one in the USR layer.

Scenario 2: Upgrading the VendTable form

As in the first scenario, Microsoft Dynamics AX has been installed as an upgrade to an existing Microsoft Dynamics AX 3.0 installation.

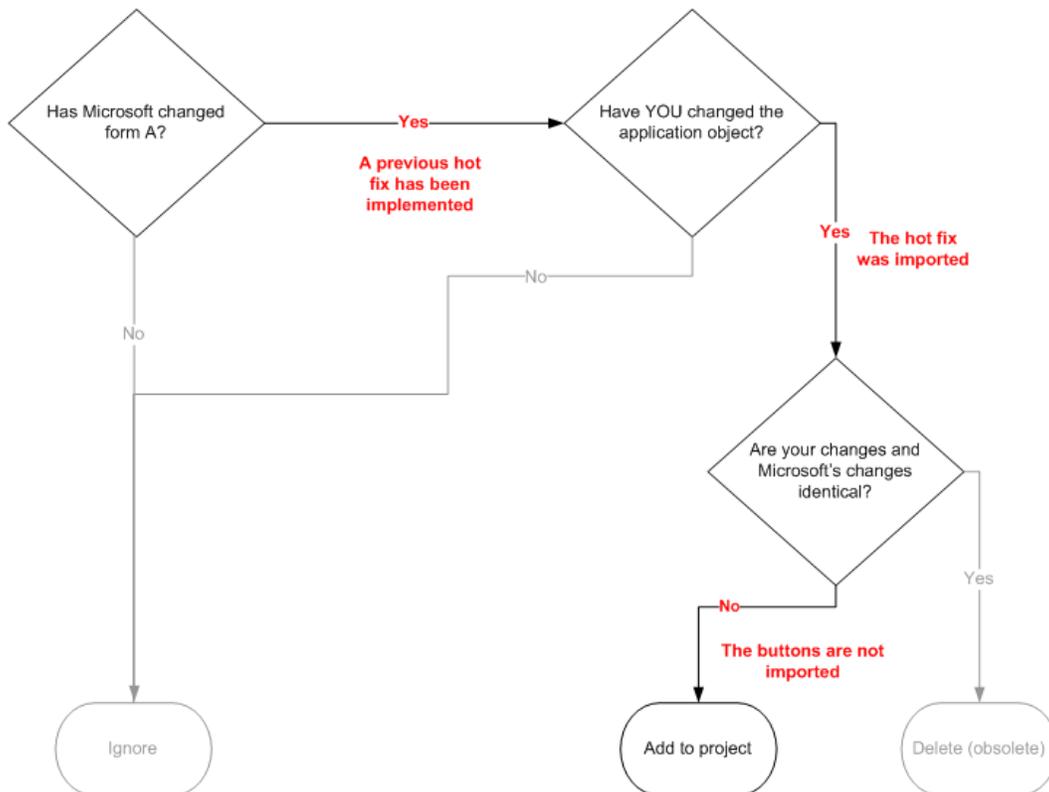
The VendTable form exists in the SYS layer.

In the SYS layer, the VendTable form has an additional button Y, added in the new version.

The VendTable form also exists in the USR layer, where it has an additional button X.

The VendTable form also exists in the USR layer, where it has an additional button X. The new and the old versions are identical.

As explained above, the Create upgrade project algorithm will be applied to the VendTable form as a whole. This results in the form being added to the upgrade project.



There is a potential conflict in the form because it has been changed in two layers. The form is added to the upgrade project and the conflict must be examined and resolved manually.

A variant on the form scenario

A variant on the form scenario occurs when a "hot fix" for Microsoft Dynamics AX 3.0 (a fix sent quickly for an issue that is causing a major problem) has been applied. When the .xpo file containing the hot fix was imported, the form was added to the USR layer. With the release of Microsoft Dynamics AX 2009, the hot fix has been implemented and is part of the standard application.

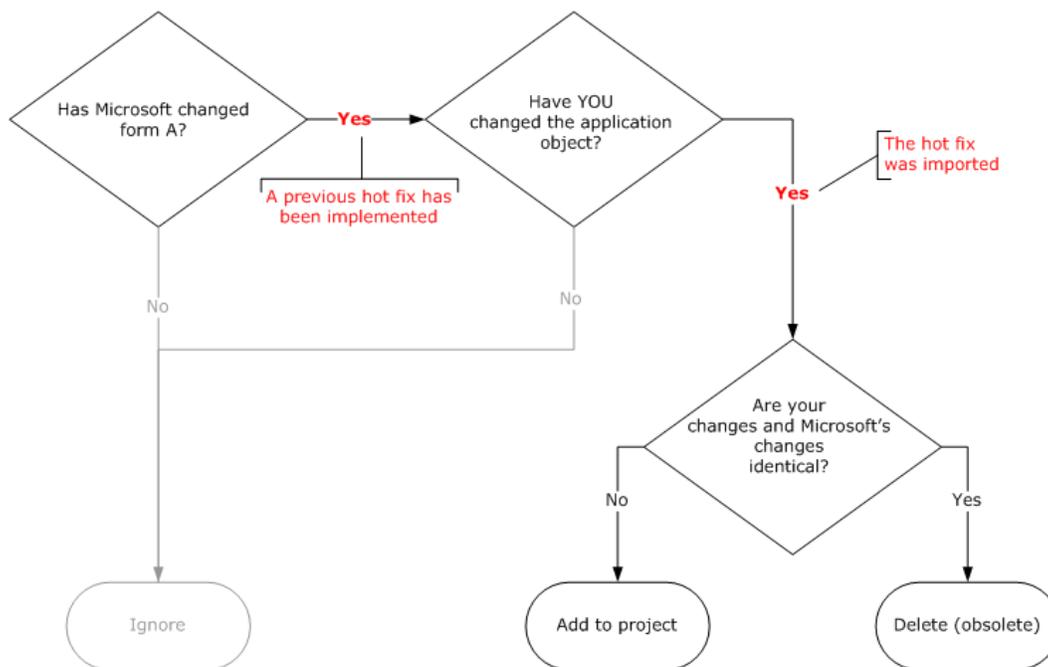
Form A exists in the SYS layer.

Form A exists in the SYS layer, in a new version as compared to the old SYS-layer version.

Form A exists in the USR layer in a modified version.

Form A exists in the USR layer in a modified version.

The following graphic shows the decision tree when implementing a hot fix.



Because form A is identical in the new SYS layer and the new USR layer, form A in the USR layer is obsolete and is deleted.

Additional code upgrade information

This section provides additional information about helping make the code upgrade process successful. Read through each section below before you begin any of the procedures.

Complete the code upgrade process on a blank database

It might be beneficial to complete the code upgrade process on a blank database so that you don't have to worry about any data during the code upgrade.

1. Install Microsoft Dynamics AX 2009 and create a new Microsoft SQL Server database using the Setup wizard. For more information, refer to the "Create a Microsoft SQL Server database using the Setup wizard" topic in the Installation Guide.
2. After installation is complete, open Windows Explorer and browse to your application files installation folder. The default location is **C:\Program Files\Microsoft Dynamics AX\50\Application\Appl\DynamicsAx1**.
3. Inside the **application files installation** folder, create a folder called **Old**.
4. Copy your prior version AOD files (*.AOD) and the label files (*.ALD, *.ALI), if needed, into the **Old** folder.
5. Copy your prior version AOD files that are outside of the DIS layer (axbux.aod, axvar.aod, axcus.aod, axusr.aod, and the patch layers if they exist: axbup.aod, axvap.aod, axcup.aod, axusp.aod) and any relevant label files (*.ald, *.ali) into the application files installation folder.

 **Note:**

Since the BUS/BUP layer is generally reserved for 3rd party additional products, you may not want to copy the axbus.aod and axbup.aod files. You may want to install new versions of the same applications that are certified for Microsoft Dynamics AX 2009.

6. Start the AOS service. This may take a few moments. The AOS service will create the axapd.aoi file for the current application files installation folder, and in the **Old** folder that you created in step 3.

Compare one layer at a time

It is a good idea to compare only one code layer at a time. For example, if you have three additional layers, you should work from the innermost layer to the outermost layer and create upgrade projects for each layer.

 **Note:**

The SYS layer is considered the innermost layer. The layer structure [inner to outermost] is SYS, GLS, LOS, DIS, BUS, VAR, CUS, USR, and then the associated patch layers for each layer.

As you work through each layer, you will need to either promote new code from the inner layers to the outer layer, or remove it if it is no longer valid.

Complete the following procedure for each layer. Once you have processed all of the upgrade projects, you should be ready to upgrade your data.

1. Complete steps 1-5 from the previous procedure.
2. Start the AOS service. This may take a few moments. The AOS service will create the AXAPD.AOI file for the current application folder, and in the **Old** folder that you created.
3. Open the Client Configuration Tool and be sure to log into the layer that corresponds to the layer file that you just copied into the application folder in step 5.
4. Start the Installation checklist. (First time only)
5. Start the **Detect code upgrade conflicts** tool (**Microsoft Dynamics AX menu > Tools > Development tools > Code upgrade > Detect code upgrade conflicts**).
6. Run the upgrade projects for this layer.
7. Stop the AOS and back up the current layer file (ax*.aod).
8. Repeat steps 5-12 with the next innermost layer that exists.

Consolidate your prior version layers into one layer file

If your previous version of Microsoft Dynamics AX has multiple layer files contained within the application, you may want to consolidate the functionality from the multiple layers into a single layer. Consolidating your previous version's layers will help make upgrading to Microsoft Dynamics AX 2009 easier.

Also, by consolidating the layers before upgrade, you can test the functionality of the merged layer against known data and processes in your modified, pre-upgrade installation of Microsoft Dynamics AX, and validate that the code is functioning as expected. It will also help make maintenance easier in the future when service packs are released, as there will be fewer layers to compare against to propagate the service pack modifications to the outer layers.

 **Note:**

Moving objects between layers may cause object ID issues. Upgrade scripts or jobs may need to be created to adjust the Microsoft Dynamics AX 2009 data model. For more information, refer to the white paper "How to Write Data Upgrade Scripts" (<http://go.microsoft.com/fwlink/?LinkId=115169&clcid=0x409>).

Take the easiest approach

Evaluate your code modifications. If Microsoft Dynamics AX 2009 does not have the functionality that existed based on your changes, then you should probably re-implement your changes in Microsoft Dynamics AX 2009.

If Microsoft Dynamics AX 2009 now contains functionality that is similar to your customizations, it would be to your advantage to remove your customization and modify the existing functionality to meet your specifications.

Also, this may be a good time to remove functionality that is not used anymore.

Show all layers

Use the **Show all layers** setting (**Microsoft Dynamics AX menu > Tools > Options > Development tab > Application object layer**) to allow developers to see all of the layers an object exists in.

If you are using Product Builder

If you plan on using Product Builder, consider devoting a layer to just the classes that Product Builder will create in the AOT. In Microsoft Dynamics AX 3.0, Product Builder could optionally write code into the AOT to help increase performance. In Microsoft Dynamics AX 4.0 or later, this is now a requirement.

If an issue arises in which Product Builder corrupts a layer file, it would be advantageous to have a layer devoted exclusively to Product Builder code. Any customizations to the system should NOT reside in the layer that Product Builder is using, so that if the layer does become corrupted, the fix can then be as simple as restoring a backup for that specific layer, recreating the AXAPD.AOI file, and restarting the AOS. Also, if there is not a backup of the Product Builder specific layer, the AOS can be stopped, the layer file removed, the AXAPD.AOI file removed, the AOS restarted, and Microsoft Dynamics AX can be functioning again.

Reduce core code modifications where possible

When creating a modification, try to avoid modifying the base classes. Instead, extend them either by creating new methods, or by creating new classes which inherit a base class. This way, when service pack or major version upgrades are done, the stand-alone code should not be adversely affected by the changes. This way, the only needed modification to the base class might be a simple call to the new functionality that has been created. This is much easier to implement and upgrade than propagating lines of code from inside many different methods to re-implement the desired functionality.

Additional data upgrade information

This section contains additional data upgrade information that you may find useful during your upgrade to Microsoft Dynamics AX 2009.

When is data upgrade necessary?

Data upgrade is necessary in the following situations:

- When a table is deleted and the data must be saved
- When a field is deleted and the data must be saved
- When unique indexes are added or changed
- When a non-unique index is changed to a unique index
- When the location where data is stored is moved—for example, when data is moved from one field to another
- When inconsistencies of old data are fixed
- When new fields need to be populated with existing data

What triggers the data upgrade process?

Microsoft Dynamics AX 2009 checks the SYSSETUPLOG table to see if information is stored in the table. If the table does not contain information, this indicates that a new installation of Microsoft Dynamics AX is being performed, and the **Installation checklist** is displayed.

If the SYSSETUPLOG table contains information, Microsoft Dynamics AX runs a script resembling the SQL script `SELECT LASTONLY FROM SYSSETUPLOG`. The script returns information including the last successfully launched version of the product, the application build value, and the kernel build value. If these values all match those of the current installation, Microsoft Dynamics AX 2009 starts in the normal way. If any values are different, the **Upgrade checklist** is displayed.

When is a manual data upgrade necessary?

Manual data upgrade may be necessary if you have created new tables or fields that depend on data in tables or fields that have been deleted or moved.

You can locate deleted tables and fields by searching for application objects prefaced with DEL_.

Deleted tables and fields

Some objects in previous versions might be removed during the upgrade to Microsoft Dynamics AX 2009. These objects are given a prefix of DEL_, and they are assigned to specific configuration keys. When these configuration keys (**Keep update objects 4.0** and **Keep update objects 4.1**) are unselected in the **Configuration** form (**Administration > Setup > System > Configuration**), the objects assigned to those configuration keys are deleted. Never manually delete tables and fields from your database.

What happens to renamed application objects during data upgrade?

Microsoft Dynamics AX tables and fields are identification-based rather than name-based, which means that they have unique identification codes and should not cause problems during the data upgrade process. Renamed application objects might cause issues with data upgrade only when the objects are referenced in your X++ code. If you reference the renamed application objects in your X++ code, you might get compile errors during the upgrade process, which means that you'll need to modify your code to use the correct objects so that you can properly compile.

What happens to deleted application objects during data upgrade?

The data upgrade process uses DEL_ as a special prefix for some application objects, such as tables, that will be deleted in the Microsoft Dynamics AX 2009 version.

The DEL_ tables and fields are necessary for data upgrade because 'old' table structures need to be maintained until the data upgrade is completed so that data is not lost.

When an object with a DEL_ prefix is introduced because of a schema change in Microsoft Dynamics AX 2009 or because you have changed your modifications, the Microsoft Dynamics AX 2009 upgrade scripts handle the changes in the standard application. One example of this is when the ReleaseUpdateDB_<old version>to<new version> class moves the fields and X++ code to the table that replaces the one with the DEL_ prefix.

The configuration keys SysDeletedObjects30 and SysDeleteObjects40 are assigned to application objects that have the DEL_ prefix.

Note:

If you have X++ code that references an application object that has the DEL_ prefix, you will need to re-evaluate the existing modifications and update them to use the new object that replaces the DEL_ object.