Microsoft Dynamics AX 2012 Implementation Planning Guide

Microsoft Corporation

Published: April 2011

This content is preliminary and is subject to change.

Table of Contents

Copyright notice

Prerequisite skills and knowledge

Microsoft Dynamics® AX is built upon several other Microsoft® products and technologies. You will require advanced information technology skills and knowledge to implement the Microsoft Dynamics AX application in a way that takes full advantage of this technology platform.

Prerequisites for system administrators

System administrators preparing to deploy Microsoft Dynamics AX should be familiar with general industry practices regarding reliability, scalability, availability, performance optimization, security, and monitoring of network infrastructure and applications. Installation of the base Microsoft Dynamics AX components, including the Application Object Server (AOS), database server, model store (previously known as the application file server), and the Microsoft Dynamics AX client, requires an understanding of the following Microsoft technologies:

- The Windows Server® operating system that is used to deploy the Microsoft Dynamics AX server components
- The Windows® client operating system that is used to deploy the Microsoft Dynamics AX Windows client
- Microsoft .NET Framework
- Windows Server Terminal Services, if required
- Advanced network management in an Active Directory® directory service environment

The following table describes necessary skills and knowledge required to deploy particular components of a Microsoft Dynamics AX implementation.

Microsoft Dynamics AX component	**Skills and knowledge required**
Database server	• Microsoft SQL Server™ administration • Relational database infrastructure planning, including sizing the database infrastructure • Performance optimization and monitoring of a database server • Database backup and recovery
Reporting and analytics	• Microsoft SQL Server Analysis Services • Microsoft SQL Server Reporting Services • Internet Information Services (IIS), Web sites, virtual directories, application pools, and Web services administration • Microsoft .NET Framework 4, ASP.NET

Microsoft Dynamics AX component	Skills and knowledge required
Enterprise Portal	• Internet Information Services (IIS) administration • Microsoft SharePoint® Foundation 2010 administration or Microsoft SharePoint Server 2010 administration, depending upon the production version used • Microsoft .NET Framework 4, ASP.NET • Creating and managing Web sites, Web services, virtual directories, and application pools
Services and Application Integration Framework (AIF)	• Internet Information Services (IIS) administration, if Web services are deployed • Microsoft BizTalk® Server, if you must integrate Microsoft BizTalk Server with Microsoft Dynamics AX • Microsoft .NET Framework, especially Windows Communication Foundation • Integration concepts such as enterprise application integration (EAI), business-to-business (B2B), and synchronous and asynchronous transports • Creating and managing Web sites, Web services, virtual directories, and application pools, if you deploy Web services • Microsoft .NET Framework 4, ASP.NET • Microsoft Message Queuing (MSMQ), if used
Workflow	• Microsoft .NET Framework, especially Windows Workflow Foundation • Internet Information Services (IIS) administration (required during upgrade from the previous release)
Project Server integration functionality	• Deploying and managing Windows services • Microsoft Message Queuing • Microsoft Project Server administration

Implementation planning guide

Welcome to the Microsoft Dynamics AX Implementation Planning Guide. The Implementation Planning Guide provides prescriptive guidance to system architects, consultants, and IT professionals involved with planning a Microsoft Dynamics AX 2012 implementation. The newest version of this guide is available from the Microsoft Download Center. You can see the TechNet Library website for up-to-date information about administration of Microsoft Dynamics AX 2012. For information about other documentation that is available, see the Documentation Resources white paper.

Architecture and planning

When you understand the architecture of Microsoft Dynamics AX, you can better plan, customize, and deploy, the Microsoft Dynamics AX system. The topics in this section provide an overview of the Microsoft Dynamics AX system and its base components.

Microsoft Dynamics AX architecture

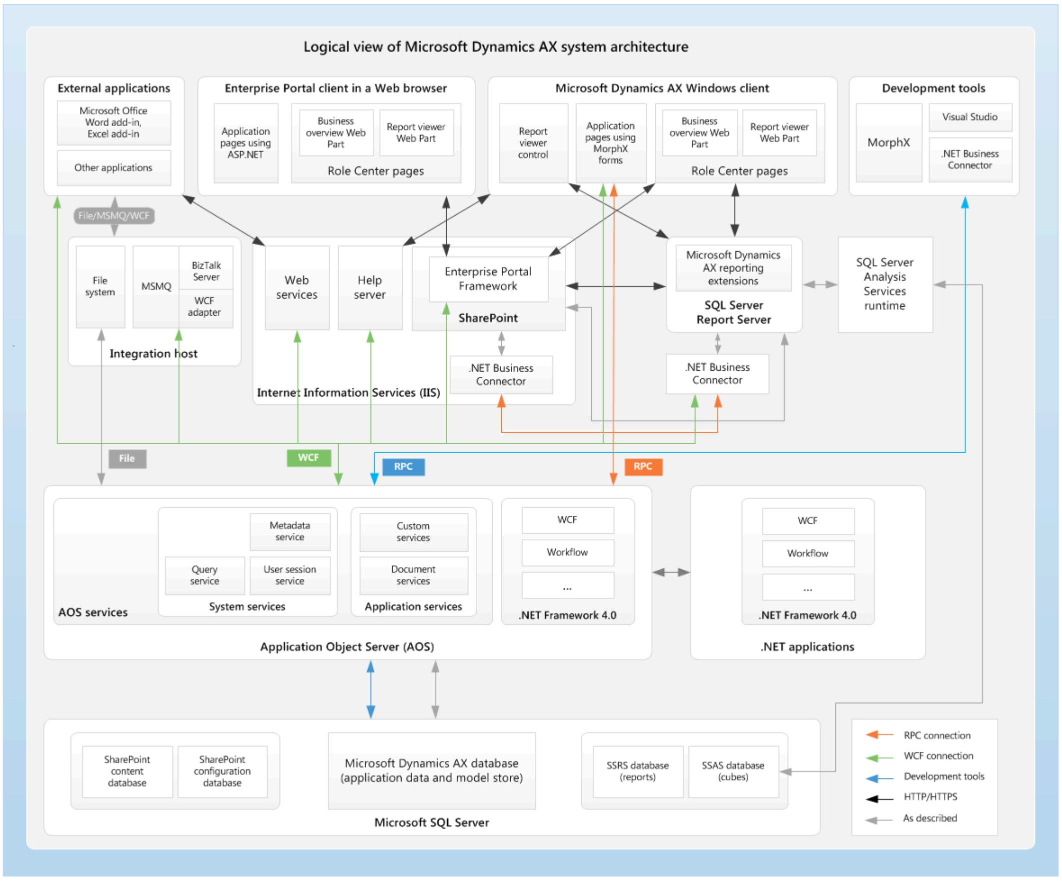
Understanding the architecture of Microsoft Dynamics AX will help you plan, customize, and deploy the Microsoft Dynamics AX system. The topics in this section provide an overview of the Microsoft Dynamics AX system and its associated components.

System architecture

Understanding the internal architecture of Microsoft Dynamics AX can help you make decisions when planning, customizing, and deploying a system. This topic provides a high-level overview of the system architecture of Microsoft Dynamics AX.

Microsoft Dynamics AX system architecture

The following diagram provides a high-level overview of Microsoft Dynamics AX system architecture. This diagram does not depict the system topology or physical infrastructure required for the deployment. Your infrastructure can consist of many Microsoft Dynamics AX components on a single physical server or on multiple physical servers. For details about Microsoft Dynamics AX components, see Component architecture. For up-to-date hardware and software requirements for Microsoft Dynamics AX, see the system requirement d.



Authentication and authorization

Microsoft Dynamics AX uses integrated Windows authentication to authenticate Active Directory users. If you configure Microsoft Dynamics AX to use a different authentication provider, users are authenticated by that provider. Authorization of access to data, business functionality, and presentation elements (forms, menus, fields and reports) is governed by Microsoft Dynamics AX security. Anonymous Web users can access the Enterprise Portal with limited functionality.

Presentation tier (clients and external applications)

A client provides an interface to Microsoft Dynamics AX data and functionality. An external application integrates with Microsoft Dynamics AX to programmatically integrate functionality or exchange data.

- The Microsoft Dynamics AX Windows client is a native 32-bit program that provides a rich user interface.
- Supported Web browsers give access to Microsoft Dynamics AX functionality and data through the Enterprise Portal.
- External applications interact with Microsoft Dynamics AX via services and the Application Integration Framework (AIF). Service and AIF provide an extensible framework for XML-based enterprise application integration (EAI), business-to-business (B2B), and service-oriented architecture (SOA) scenarios.

Note:
Use services and AIF to interact with the Microsoft Dynamics AX application. We recommend against using the .NET Business Connector to integrate with the Microsoft Dynamics AX application.

Application tier

The application tier consists of one or more of the following Microsoft Dynamics AX components or computer roles.

Windows Active Directory domain controller

Microsoft Dynamics AX uses integrated Windows authentication to authenticate Active Directory users. If you configure Microsoft Dynamics AX to use a different authentication provider, users are authenticated by that provider.

An Active Directory domain controller is a prerequisite for installing Microsoft Dynamics AX.

Application Object Server

The Application Object Server (AOS) controls communication among Microsoft Dynamics AX clients, databases, and applications. In this release, the AOS also hosts the Microsoft Dynamics AX services and workflow. You can deploy the AOS on a single computer or create a load-balanced cluster of multiple AOS servers. The AOS is a Windows service and requires a Windows Server operating system. For up-to-date hardware and software requirements for Microsoft Dynamics AX, download the system requirements document from the Microsoft Download Center.

The AOS uses many libraries from .NET Framework 4, such as the Windows Communication Foundation and Windows Workflow Foundation.

Enterprise Portal

The Enterprise Portal and its applications allow you to interact with Microsoft Dynamics AX from a Web browser. The Enterprise Portal enables internal users (employees) and external users (vendors, customers, business partners) to access data and functionality through a highly customizable, role-based Web portal. You can also create Internet facing public sites with limited functionality for anonymous users. The

Enterprise Portal requires ASP.NET, Microsoft SharePoint Foundation 2010 or Microsoft SharePoint Server 2010, and Internet Information Services (IIS).

Reporting

Microsoft SQL Server Reporting Services is a solution that enables users to create and view traditional, paper-based reports, as well as interactive, Web-based reports.

To integrate Microsoft Dynamics AX and Reporting Services, you must install the reporting extensions on a server running Reporting Services.

After you install the reporting extensions, you will be able to deploy Microsoft Dynamics AX default reports to Reporting Services.

Analytics

Microsoft SQL Server Analysis Services is a server-based solution that provides online analytical processing (OLAP) functionality. OLAP reports help users analyze business data and identify trends that they might not otherwise discover when viewing data in traditional reports.

To integrate Microsoft Dynamics AX and Analysis Services, you must install the analysis extensions on a server running Analysis Services.

When you install the analysis extensions, a default OLAP database and cubes are deployed to Analysis Services.

Workflow

Microsoft Dynamics AX supports workflow processes, such as approval of purchase requisitions, within the application. Microsoft Dynamics AX uses the Windows Workflow Foundation to support workflow on the Application Object Server (AOS). The Microsoft Dynamics AX workflow component is automatically installed on the AOS and the Microsoft Dynamics AX Windows client computers during installation.

Services and Application Integration Framework (AIF)

Microsoft Dynamics AX provides a first-class programming model for integration. Services enable Microsoft Dynamics AX to expose its functionality by means of Windows Communication Foundation-based services. AIF supports the processing of inbound and outbound messages such as message transforms and value lookups. Together, services and Application Integration Framework (AIF) provide the programming model, tools and infrastructure support for XML-based integration of application functionality and data with Microsoft Dynamics AX.

Help server

The Microsoft Dynamics AX help system uses a server to store and distribute Help documentation. The help viewer is a client application that displays help information. You open the help viewer when you press F1 or follow a help menu option to display application help topics.

Microsoft Project Server integration

The Microsoft Dynamics AX integration with Project Server requires two integration components, the synchronization service for Microsoft Project Server and synchronization proxy for Microsoft Project Server. To use this functionality, you must install both the components. For more information about Project Server integration, see Project Server integration architecture.

Data tier

Microsoft Dynamics AX requires Microsoft SQL Server for the Microsoft Dynamics AX database, the model store database, SharePoint databases, and SQL Server Reporting Services database. Support for OLAP cubes requires a SQL Server Analysis Services database.

The Microsoft Dynamics AX database

The *database* is a SQL Server database containing transaction and reference data. This database is functionally equivalent to the principal database in Microsoft Dynamics AX 4.0 and 2009.

The model store

The *model store* is a SQL Server database where all Microsoft Dynamics AX application elements are stored, including customizations. Layer and model information are integral parts of the store. The Application Object Server (AOS) has access to the model store, handles layer-flattening, and provides model data to all the Microsoft Dynamics AX sub-systems, such as form- and report-rendering and X++ code. The model store replaces the AOD files used in previous versions of Microsoft Dynamics AX.

Other databases

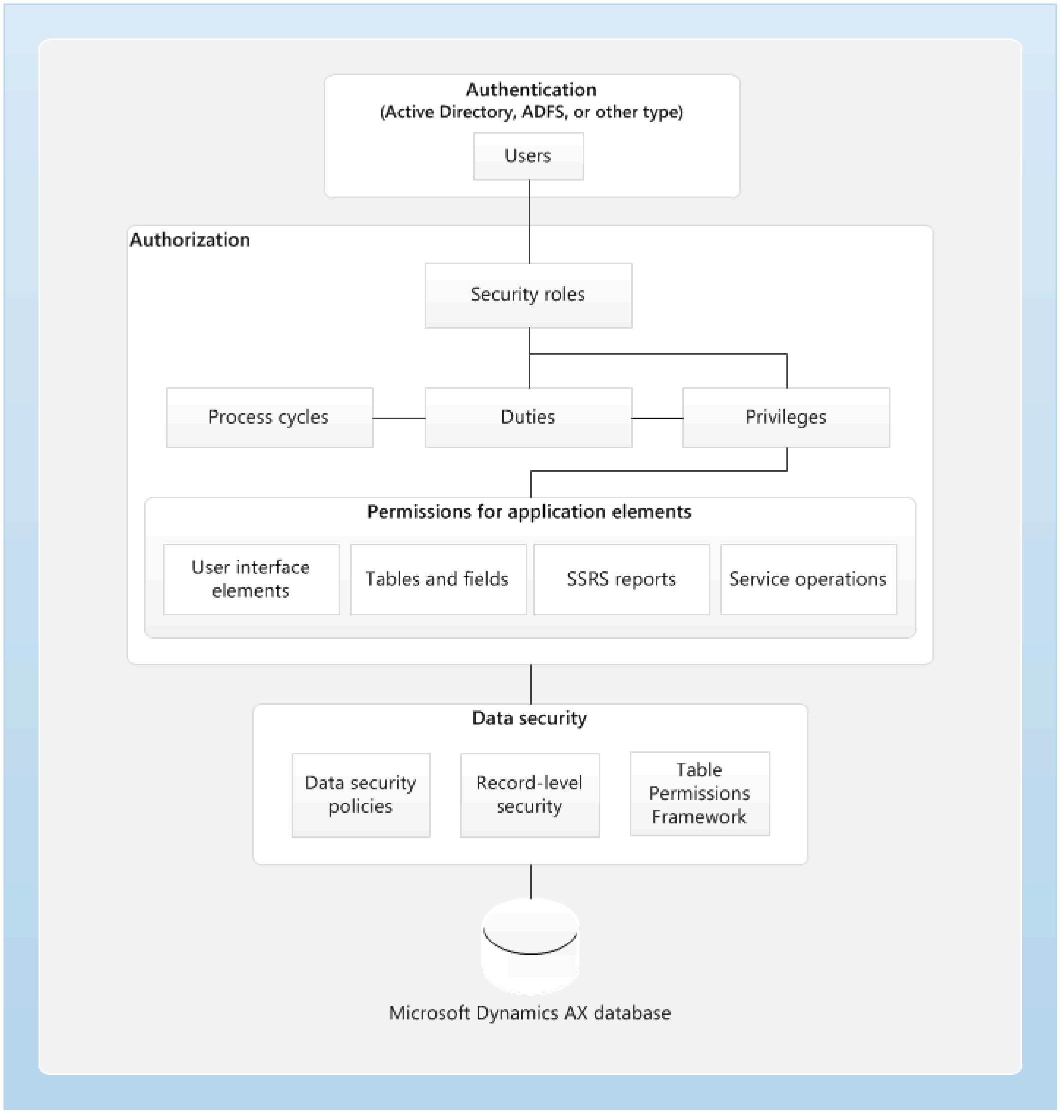
The Enterprise Portal requires SharePoint content and configuration databases. SQL Server report server requires a SQL Server Reporting Services database. Support for OLAP cubes requires a SQL Server Analysis Services database.

See Also

Application integration

Security architecture

Understanding the security architecture of Microsoft Dynamics AX can help you customize security to fit your business needs. The following diagram provides a high-level overview of Microsoft Dynamics AX security architecture.



Authentication

By default, only authenticated users who have rights in Microsoft Dynamics AX can establish a connection.

Microsoft Dynamics AX uses integrated Windows authentication to authenticate Active Directory users. If you configure Microsoft Dynamics AX to use a different authentication provider, users are authenticated by that provider.

After a user connects to Microsoft Dynamics AX, access is determined by the duties and privileges assigned to the security roles that the user belongs to.

Authorization

Authorization is the control of access to the Microsoft Dynamics AX application. Security permissions are used to control access to individual application elements: menus, menu items, action and command buttons, reports, service operations, Web URL menu items, Web controls, and fields in the Windows client and Enterprise Portal.

In Microsoft Dynamics AX, individual security permissions are combined into privileges, and privileges are combined into duties. The administrator grants application access to security roles by assigning duties and privileges to the roles.

Data security

Authorization is used to grant access to elements of the application. In contrast, data security is used to deny access to tables, fields, and rows in the database.

Use the extensible data security framework to control access to transactional data by assigning data security policies to security roles. Data security policies can restrict data based on effective date or based on user data such as sales territory or organization.

In addition to the extensible data security framework, the record-level security feature can be used to limit access to data based on a query. However, because the record-level security feature will be deprecated in a future release of Microsoft Dynamics AX, we recommend using data security policies instead.

Some data is additionally protected by the Table Permissions Framework. Data security for specified tables is enforced by the AOS. Data is not sent to the client if the user does not have access to that data. For more information about the Table Permissions Framework, see the Microsoft Dynamics AX Security Hardening Guide.

Component architecture

This section lists Microsoft Dynamics AX components by functional category. The topics in this section describe the Microsoft Dynamics AX development environment and the architecture of selected components.

Database components

Database components include the Microsoft Dynamics AX database, which contains business transaction data, and the model store, which contains application elements of Microsoft Dynamics AX.

Server components

Server components include the Application Object Server (AOS) and the Microsoft Dynamics AX components that run on the AOS or on Internet Information Services (IIS). Server components include the following:

- Application Object Sever (AOS)
- Services (hosted on the AOS)
- Workflow (hosted on the AOS)
- Enterprise Portal (hosted on IIS)
- Enterprise Search (hosted on IIS)
- Help server (hosted on IIS)

Business intelligence components

Business intelligence components provide reporting and analytical functionality that enables you to view and interpret business data. The reporting extensions enable you to create reports using Microsoft SQL Server Reporting Services. Integration with SQL Server Analysis Services enables you to use cubes for business intelligence and analytical reporting in Microsoft Dynamics AX.

Client components

Client components give users access to Microsoft Dynamics AX data and functionality. Client components include the following:

- Microsoft Dynamics AX Windows client
- Microsoft Office Add-ins

Development components

Development components are tools you can use to carry out software development tasks. The integrated development environment (IDE) in Microsoft Dynamics AX is called MorphX. It includes tools for designing, editing, compiling, and debugging code in Microsoft Dynamics AX. In addition to MorphX, the Microsoft Dynamics AX development environment includes the Microsoft Visual Studio® tools and debugger. You can use the Visual Studio tools to create customizations or extensions to Enterprise Portal and create advanced production reports for Microsoft Dynamics AX using SQL Server Reporting Services.

Integration components

Integration components enable Microsoft Dynamics AX to integrate with external applications. The following list describes the integration components:

- .NET Interop
- .NET Business Connector
- Web services hosted on IIS
- Synchronization service for Microsoft Project Server
- Synchronization proxy for Microsoft Project Server

Database components

Microsoft Dynamics AX relies on a single Microsoft SQL Server database. During upgrade, an additional database, the *baseline model store* is used. This topic provides an overview of the databases, and the types of tables they store.

The Microsoft Dynamics AX database contains two primary types of tables:

- Tables that can be accessed from the data dictionary in the AOT.
- Tables that can be accessed from the system documentation section of the AOT: kernel and *model store* tables.

The baseline model store database holds model store tables for the previous version of metadata. It used only during upgrade. The baseline model store is similar to the **old** folder in previous releases of Microsoft Dynamics AX.

Tables that can be accessed from the data dictionary

The types of tables that can be accessed from the data dictionary in the AOT include:

- **Framework**. This table group includes tables that are used by underlying Microsoft Dynamics AX frameworks, such as the Application Integration Framework. These tables are created during installation, and are not associated with configuration keys.
- **Group**. This table group includes tables that are used to categorize the tables in the **Main** table group.
- **Main**. This table group includes the principal or master tables that contain data for central business objects. These tables typically hold static, base information.
- **Miscellaneous**. This table group includes tables that have not been otherwise categorized. This is the default table group for a new table.
- **Parameter**. This table group includes tables that contain parameters or setup information for tables in the **Main** table group.
- **Reference**. This table group includes tables that contain reference data.
- **Transaction**, **Transaction header**, **Transaction line**. These table groups include tables that contain transaction data. **Transaction header** tables categorize the tables in the **Transaction line** table group. There is a one-to-many relationship between a **Transaction header** table and a **Transaction line** table.
- **Worksheet**, **Worksheet header**, **Worksheet line**. These table groups include tables that contain information to be validated and made into transactions. Unlike the data contained in tables in the **Transaction** table groups, data in the **Worksheet** table groups is temporary. After data from these tables has been rolled forward into transaction tables, the **Worksheet** tables become obsolete, and may be deleted without affecting system stability.

Tables that can be access from system documentation

Kernel tables and model store tables are visible in the AOT in the **System Documentation** > **Tables** section. They cannot be directly imported, exported, or changed.

Kernel tables

Kernel tables are used by Microsoft Dynamics AX. They are not associated with table groups.

Model store

The model store is the portion of the Microsoft Dynamics AX database where all Microsoft Dynamics AX application elements are stored, including customizations. The model store replaces the AOD files used in previous releases of Microsoft Dynamics AX. For more information, see Model store architecture.

The model store can be managed through the AXUtil command line utility, or by using Windows PowerShell®. Within Microsoft Dynamics AX, the model store tables are visible in the in the **System Documentation** > **Tables** > **SysModel*** section of the AOT.

Model store architecture

This topic describes the architecture of the model store.

The *model store* is the portion of the Microsoft Dynamics AX database where all Microsoft Dynamics AX application elements are stored, including customizations. The model store replaces the AOD files used in previous releases of Microsoft Dynamics AX.

Layer and model information are integral parts of the store. The Application Object Server (AOS) has access to the model store, handles layer-flattening, and provides model data to all the Microsoft Dynamics AX sub-systems, such as form- and report-rendering and X++ code.

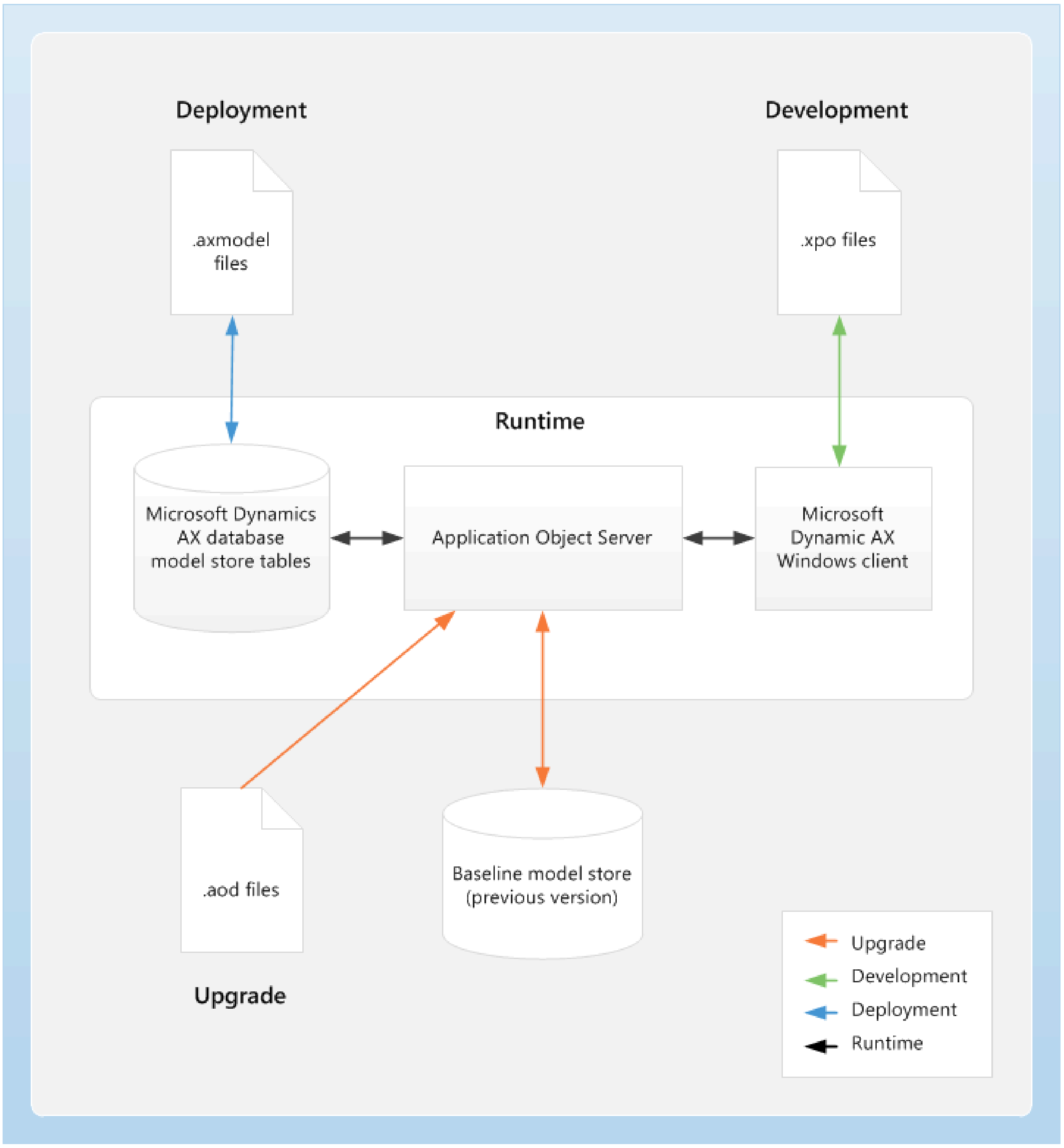
Microsoft Dynamics AX contains sixteen layers. Each layer consists of one or more logical parts called models. A system generated model exists for each layer. For example, the **VAR Model** is the system generated model for the VAR layer. The system generated models allow you to install and start working with the base Microsoft Dynamics AX system. You can leverage the capabilities of models, and tools and functionality that support the models, during customization of the Microsoft Dynamics AX application. For more information about models and layers, see:

- Model file overview
- Application Object Layers

The following list describes the different ways the model store is used by Microsoft Dynamics AX.

- **Installation.** During installation, the Setup program uses axutillib.dll to import the .axmodel files from the installation path into the model store.
- **Upgrade.** During an upgrade, the application (AOD) files from the previous release are imported into the model store (the new model) and into the baseline model store (the model store for the previous version of metadata). The baseline model store is similar to the **old** folder in previous releases of Microsoft Dynamics AX.
- **Development environment.** In the development environment, developers can continue to use .xpo files to export and import their code. Use .axmodel files to migrate application elements from one environment to another, for example, from a development environment to a test environment. Export models from the source system to .axmodel files and then impor t.axmodel files into the target system.
- **Runtime.** To respond to client requests, the Application Object Server (AOS) retrieves the application elements, such as forms, reports, and classes, from the model store at runtime.

The following diagram provides an overview of the model store architecture.

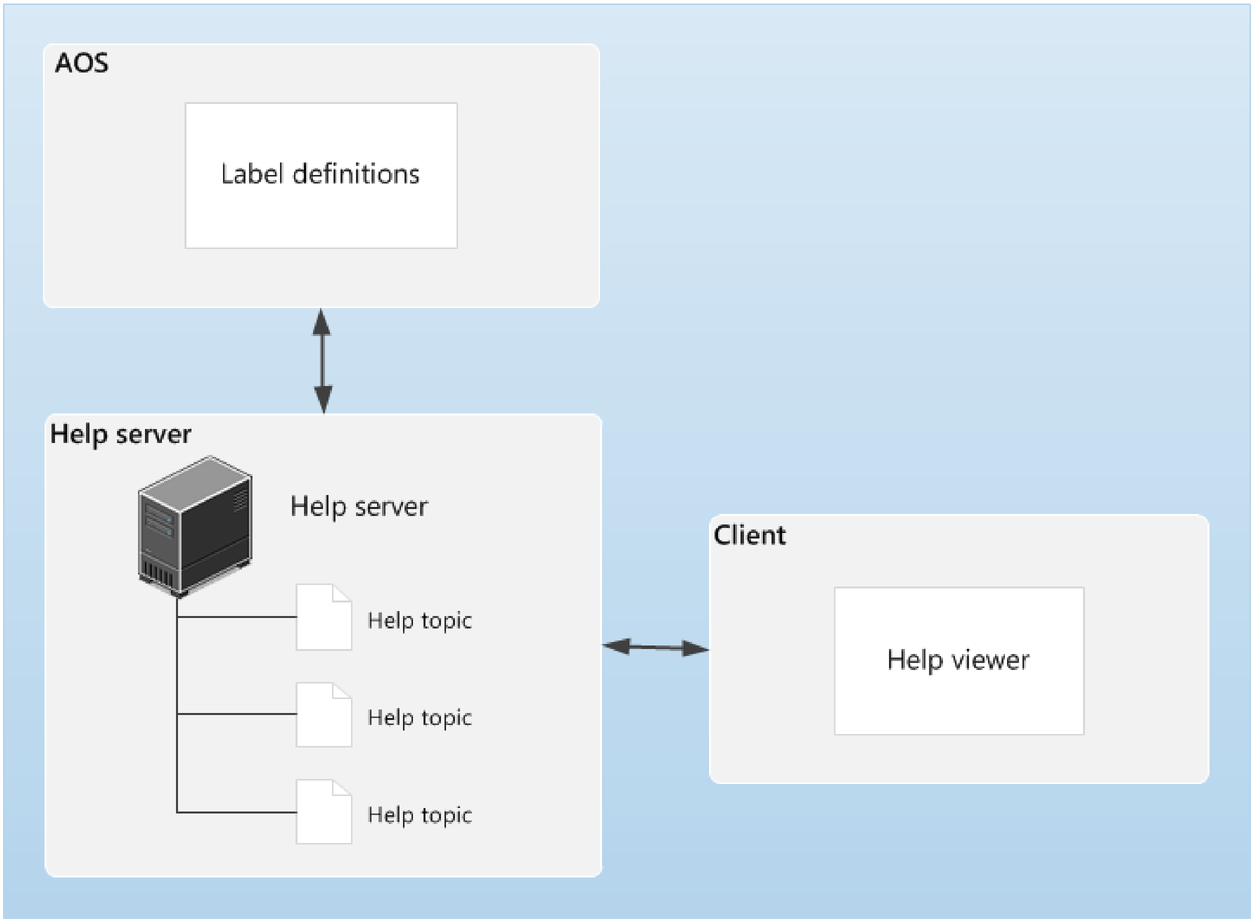

Deployment
Development
.axmodel files
.xpo files
Runtime
Microsoft Dynamics AX database model store tables
Application Object Server
Microsoft Dynamic AX Windows client
.aod files
Baseline model store (previous version)
Upgrade
Upgrade
Development
Deployment
Runtime

Server components

The topics in this section provide an overview of the Microsoft Dynamics AX server components. The server components include the Application Object Server (AOS) and those components that are either hosted on the AOS or on Internet Information Services (IIS).

Help system architecture

The following diagram illustrates the architecture of the Microsoft Dynamics AX help system.



To better understand how the components in this diagram work together, consider the following example.

1. An employee clicks the **Help** menu or presses **F1** when viewing a form in Microsoft Dynamics AX.
2. The Microsoft Dynamics AX client determines which help topic should be displayed. It requests that specific topic from the help server.
3. The help server locates the help topic and determines if there are any labels to define for that topic. If so, the help server requests the definitions of the labels from the Microsoft Dynamics AX Application Object Server (AOS).

For example, suppose a help topic contains the label *@SYS11904*. The help sever will request the definition of this label from the AOS. After the AOS returns the definition, *Customer group*, the help server replaces all instances of *@SYS11904* with *Customer group*.

4. The help server sends the help topic to the client, where it is displayed in the help viewer.

Enterprise Portal architecture

This topic introduces concepts that pertain to the architecture of Enterprise Portal for Microsoft Dynamics AX. The topic also describes the various components of the Enterprise Portal architecture.

About Enterprise Portal

Microsoft Dynamics AX provides a set of Web sites that give you access to data. On these sites, you can also participate in business processes by using Web-based forms. These sites are collectively called Enterprise Portal. Enterprise Portal requires Internet Information Services (IIS), which is a feature of Windows Server, and either MicrosoftSharePoint Foundation 2010 or MicrosoftSharePoint Server 2010.

Role Centers

Enterprise Portal can be configured to display role-specific home pages that are called *Role Centers*. Role Centers provide an overview of information that pertains to a user's job function in the business or organization. This information includes transaction data, alerts, links, and common tasks that are associated with the user's role in the company. Role Centers also include reports that are generated by SQL ServerReporting Services or SQL ServerAnalysis Services. Microsoft Dynamics AX 2012 includes more than two dozen predefined Role Centers, which users can access from Enterprise Portal or the Microsoft Dynamics AX client.

Sites and pages

An Enterprise Portal site consists of a root SharePoint 2010 products site and collections of subsites. The subsites approximate the features and functionality of the modules in the Microsoft Dynamics AX client.

An Enterprise Portal page can include standard Microsoft Dynamics AX Web parts, such as the toolbar, or User Control Web parts that display Microsoft Dynamics AX data. An Enterprise Portal page can also include standard SharePoint 2010 products Web parts, such as lists, announcements, and discussions. Users can modify these Web parts as needed. If you set up and configure Enterprise Portal with Role Centers, Role Center pages can include the following elements:

- Cues that provide a visual representation of records based on the status of the records. For example, there can be cues for pending sales orders or items that are on backorder.
- Key performance indicators (KPIs) that provide information from predefined data cubes. You can use this information to monitor business performance against a defined goal.
- A Report Web part that provides access to SQL ServerReporting Services reports.
- A Business Overview Web part that displays historical performance, such as year-over-year performance or month-over-month performance.
- A work list that displays action items that are generated either by a workflow or by an alert, according to business needs.
- Community links that provide access to items that are published on community sites for Finance, Services, and Sales and Marketing.
- Links that provide access to important internal and external sites.

Customizing Enterprise Portal

Enterprise Portal is built on ASP.NET. All Enterprise Portal objects are located in the **Web** node of the Application Object Tree (AOT).

Microsoft Dynamics AX includes a standard Web part that can host a User Control. Developers can write or modify User Controls in MicrosoftVisual Studio. User Controls are used to present Microsoft Dynamics AX content on a page, and they are the primary way to add new functionality to Enterprise Portal.

Users and communication

In Microsoft Dynamics AX, Enterprise Portal users, or Web users, can be any of the following individuals:

- Employees who access Microsoft Dynamics AX through an intranet or an extranet
- Customer or vendors who access Microsoft Dynamics AX through an extranet
- Unsolicited vendors who want to sign up to be vendors, and who access Microsoft Dynamics AX through a public Internet site

All Web users access Microsoft Dynamics AX through Enterprise Portal.

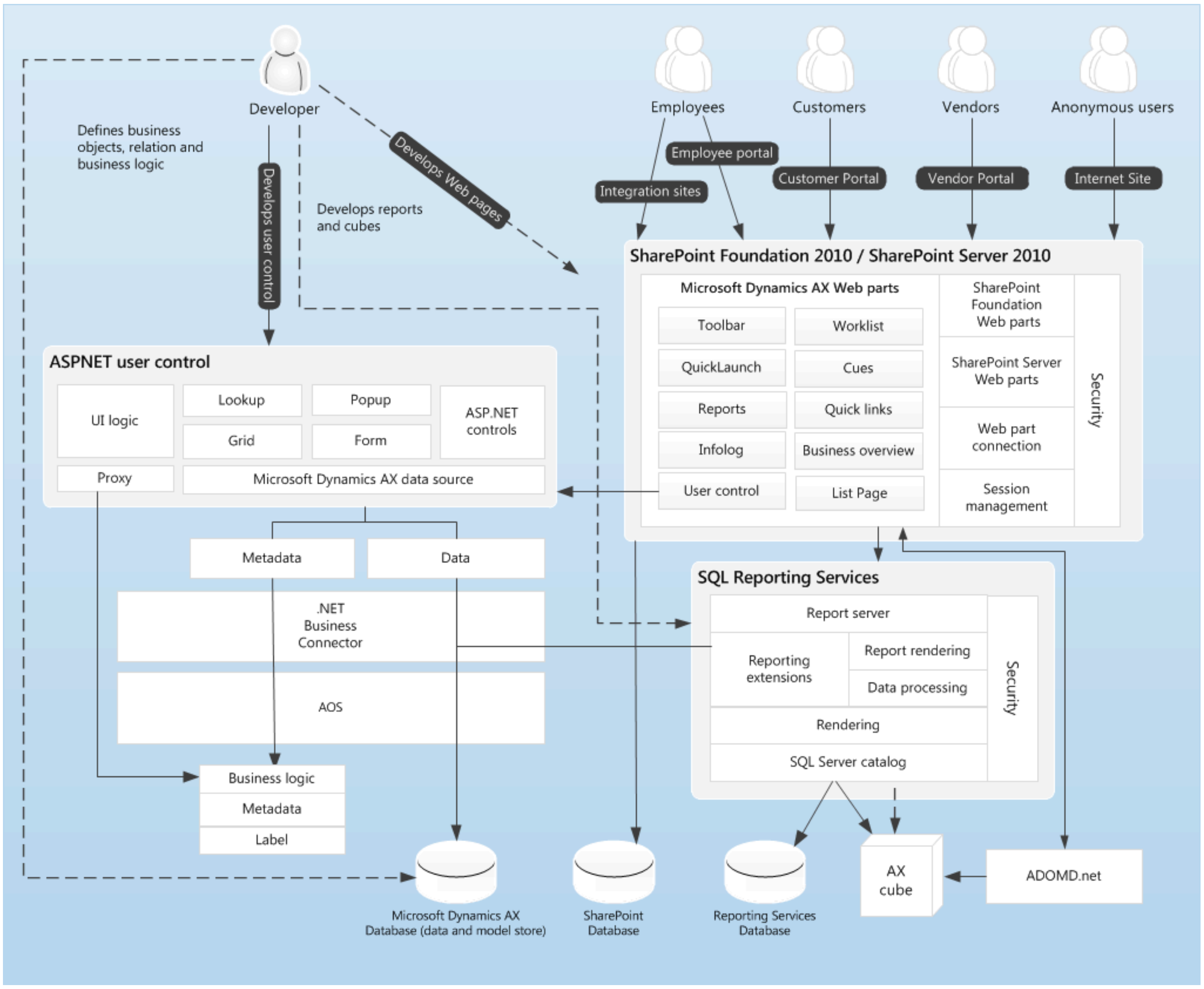
Note the following information about Enterprise Portal client connections and communications:

- All browser-based clients and Microsoft Dynamics AX clients access Role Centers through Enterprise Portal. Microsoft Dynamics AX clients use a browser control to display Role Centers.
- Enterprise Portal uses the Report Web part to display reports that exist on the SQL ServerReporting Services server.
- Enterprise Portal uses ASP.NET user controls and the Enterprise Portal framework to display Microsoft Dynamics AX data and reports.
- Enterprise Portal uses Windows Communication Framework (WCF) and .NET Business Connector to interact with an Application Object Server (AOS).

The language that is used in the user interface for Enterprise Portal is determined by the user interface language that is specified for each user in the Microsoft Dynamics AX client. The user interface language also determines how values are formatted.

Enterprise Portal architecture

The following diagram provides a high-level overview of the Enterprise Portal architecture.



AOS architecture

Introduction to the Application Object Server architecture

An Application Object Server (AOS) is a core component of the Microsoft Dynamics AX installation and is installed by using Setup. An AOS enforces security, manages connections between clients and the database, and provides the foundation where business logic for Microsoft Dynamics AX is executed. An AOS is implemented as a MicrosoftWindows service. By default, an AOS is listed in the **Services** pane as **Microsoft Dynamics AX Object Server 6.0$***InstanceName*. As a Windows service, AOS works in the following ways:

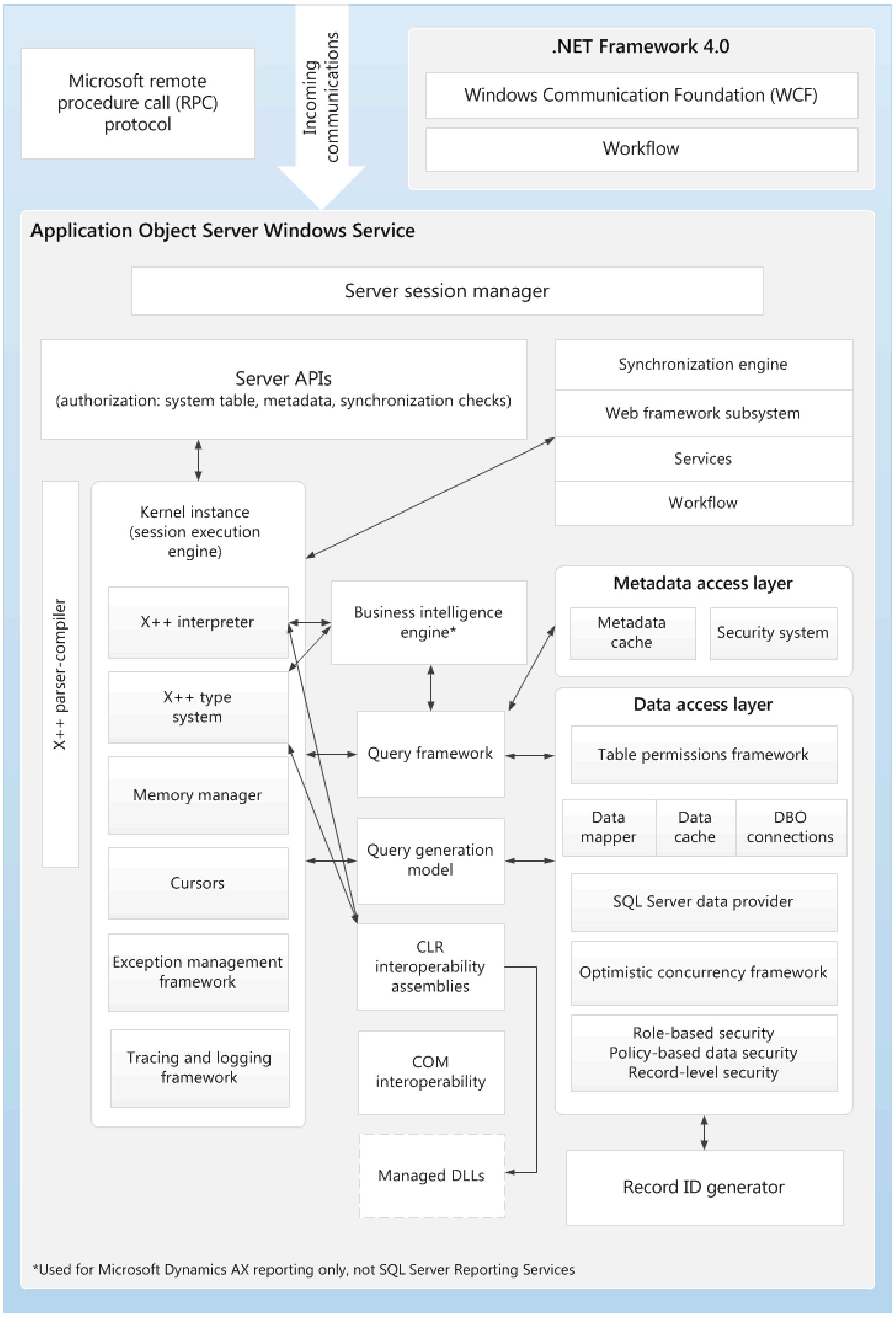
- An AOS runs in the security context of either a specific domain account or the NT Authority/Network Service account, depending on the setup.
- The status of an AOS is reported to the Windows event logs. Therefore, administrators can view errors and warnings that can help them troubleshoot problems.

You can install an AOS on a single computer, together with the database, model store, and other Microsoft Dynamics AX components. Alternatively, you can install application object servers on multiple computers and group these computers in a load-balanced cluster. Because Microsoft Dynamics AX requires Windows-integrated authentication for all servers in the system, you must be running Active Directory.

Client/AOS communications

Clients communicate with an AOS by using remote procedure calls (RPCs), Windows Communication Foundation (WCF), or AOS services. In previous releases, other components and third-party programs could communicate with an AOS by using either .NET Business Connector or Application Integration Framework (AIF). For this release, we recommend that third-party programs use AOS services to communicate with AOS.

The following diagram shows the AOS architecture.


Microsoft remote procedure call (RPC) protocol
Incoming communications
.NET Framework 4.0
Windows Communication Foundation (WCF)
Workflow
Application Object Server Windows Service
Server session manager
Server APIs
(authorization: system table, metadata, synchronization checks)
Synchronization engine
Web framework subsystem
Services
Workflow
X++ parser-compiler
Kernel instance (session execution engine)
X++ interpreter
X++ type system
Memory manager
Cursors
Exception management framework
Tracing and logging framework
Business intelligence engine*
Query framework
Query generation model
CLR interoperability assemblies
COM interoperability
Managed DLLs
Metadata access layer
Metadata cache
Security system
Data access layer
Table permissions framework
Data mapper
Data cache
DBO connections
SQL Server data provider
Optimistic concurrency framework
Role-based security
Policy-based data security
Record-level security
Record ID generator
*Used for Microsoft Dynamics AX reporting only, not SQL Server Reporting Services

Workflow system architecture

The workflow infrastructure consists of two components that are hosted on the Application Object Server (AOS): the X++ workflow runtime and the managed workflow runtime.

The X++ workflow runtime consists of:

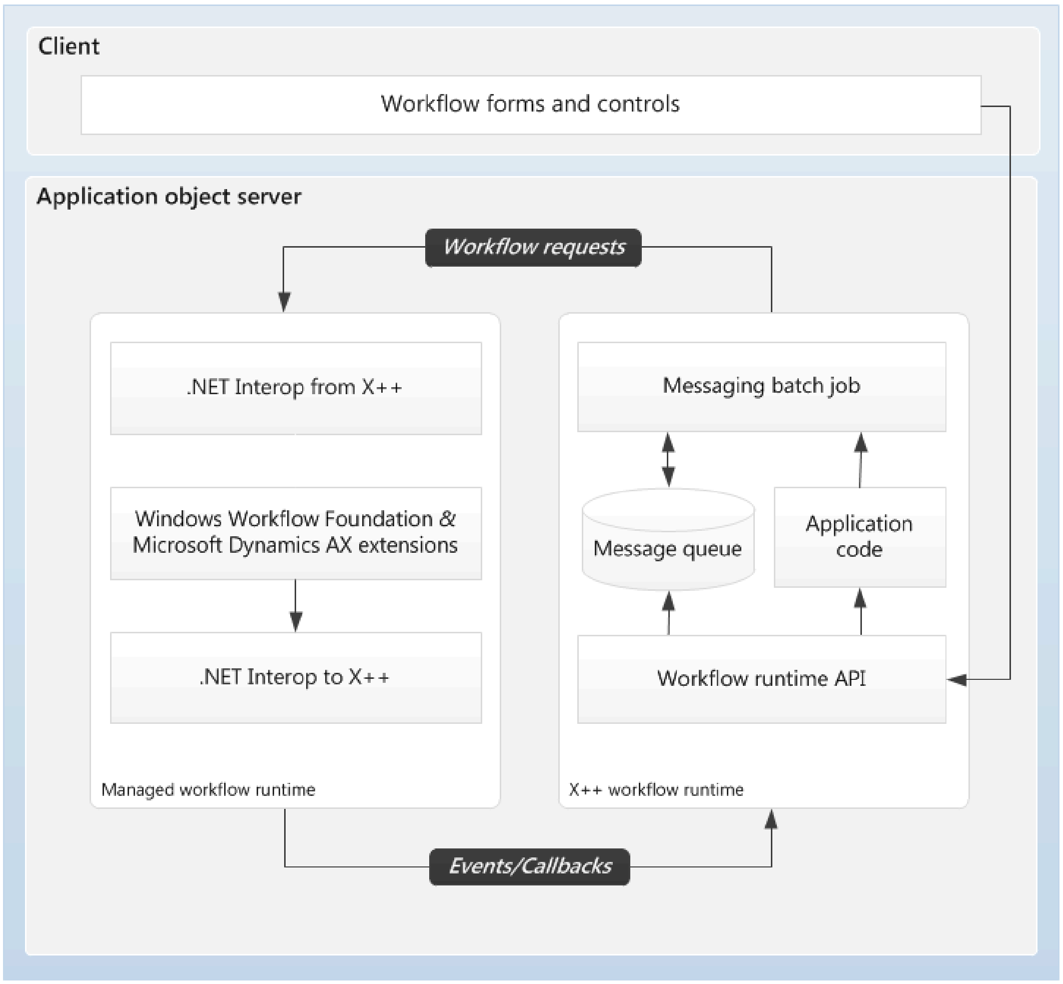
- Workflow runtime API
- A messaging batch job
- A message queue

The messaging batch job or the workflow runtime API can invoke the application code, if it is required. The X++ workflow runtime is compiled into the common intermediate language (CIL) of the .NET Framework.

The managed workflow runtime consists of the Windows Workflow Foundation and Microsoft Dynamics AX extensions.

Logically, the workflow infrastructure is an extension of Microsoft Dynamics AX and is transparent to users. Physically, both the X++ workflow and the managed workflow runtimes are hosted on the AOS. The workflow infrastructure uses batch processing on the AOS and .NET Interop to integrate both subsystems and pass messages from one subsystem to another. The X++ code executed in the batch processor is compiled to .NET CIL. The batch processing runs in the .NET common language runtime (CLR).

The following diagram provides the high-level architecture of the workflow infrastructure.



Users can use the workflow forms and controls in the Microsoft Dynamics AX client and in Enterprise Portal to participate in business processes. Programmatically, any components that can invoke X++ code can use X++ to invoke a workflow or submit a document to a workflow. The following table describes the workflow steps that occur when a user submits an expense report to the workflow system for approval.

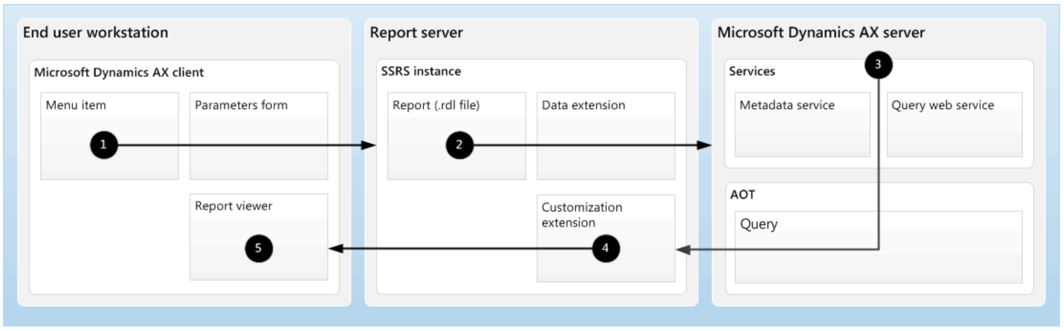
Step	Runtime	Activity
1	X++ workflow runtime	A user submits an expense report by clicking the **Submit** button on one of the workflow controls. This causes X++ code to activate a workflow instance by calling the workflow runtime API. The workflow runtime API posts a message to the message queue. The messaging batch job reads the message and sends a workflow activation request to the managed workflow runtime. **Note:** The messaging batch job processes the message queue at one-minute intervals.

Step	Runtime	Activity
2	Managed workflow runtime	.NET Interop from X++ receives the message and starts a new workflow instance via Windows Workflow Foundation. This workflow instance performs a callback to the X++ workflow runtime API via .NET Interop to X++ CIL and posts a message that the workflow has started. After posting the message, the managed workflow runtime saves the idle workflow instance to the Microsoft Dynamics AX database. Runtime then removes it from memory. When the managed workflow runtime receives another message from the X++ workflow runtime for this workflow instance, it restores the workflow instance to memory and resumes it. Each workflow instance is unique. If you have two users who submit their expense reports for approval, two workflow instances are started.
3	X++ workflow runtime	The messaging batch job reads the “workflow started” message from the message queue and invokes the application event handler to process a “workflow started” event. The batch job then posts an acknowledgement message that the event was processed.
4	Both	This same messaging pattern is repeated as necessary throughout the life cycle of the workflow instance.

The workflow architecture provides for a reliable and durable messaging system. It makes sure that the state of the workflow is always synchronized with the state of the application. In case of an unexpected hardware or software failure, the workflow instance state is returned to its last known save point and the message stays in the queue. Thus, from an architecture perspective, the recovery model is to fix the problem and resume the workflow.

Reporting architecture

The following diagram illustrates the architecture of the reporting functionality in Microsoft Dynamics AX.



To better understand how a report is rendered, review the following steps.

1. **A user requests a report.**

   Assume that a user clicks a menu item in the Microsoft Dynamics AX client. The menu item is bound to a SQL Server Reporting Services report.

   After the user clicks the menu item, a parameters form is displayed to the user. The user enters parameters to filter the data that will be displayed on the report.

   The Microsoft Dynamics AX client then requests the report from Reporting Services. (The request includes the parameters entered by the user.)

2. **Reporting Services receives the request and asks the Microsoft Dynamics AX server for the report data.**

   Reporting Services receives the request and examines the report on the server. The report is stored as an .rdl file. The .rdl file indicates the report's data source. (The data source could be a Microsoft Dynamics AX query, a report data provider class, or an external data source via report data methods.)

   In cases where a Microsoft Dynamics AX data source is used for the report, Reporting Services will use the Microsoft Dynamics AX data extension to retrieve the data.

   At this point, Reporting Services asks Microsoft Dynamics AX for metadata about the data source. Reporting Services then requests the data for the report.

3. **The Microsoft Dynamics AX server receives the request and sends the report data back to Reporting Services.**

   The Microsoft Dynamics AX services examine the query in the AOT to return the requested metadata. The services also execute the query to generate the data for the report.

   Microsoft Dynamics AX returns the metadata and data to Reporting Services.

   Note:
   Microsoft Dynamics AX enforces security on all data returned. If the user who is running the report is not allowed to see a specific field, the data for that field is not returned.

4. **Reporting Services renders the report and sends it to the Microsoft Dynamics AX client.**

   The Microsoft Dynamics AX customization extension formats the report. The customization extension uses metadata to provide automatic formatting of data and can affect the positioning and layout of elements in the report.

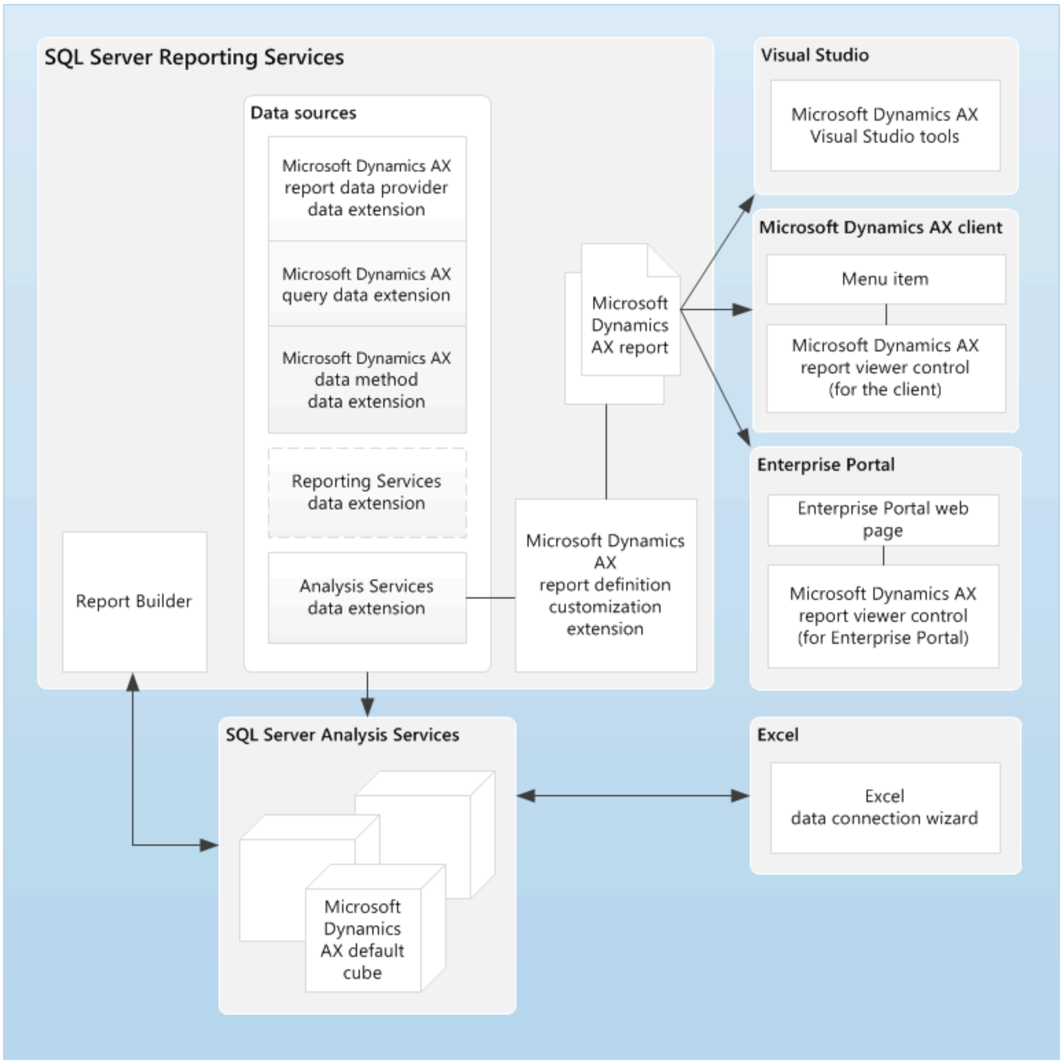
   Reporting Services then renders the report into a visual representation and sends that to the Microsoft Dynamics AX client.

5. **The report is displayed to the user.**

   The Microsoft Dynamics AX client displays the report to the user in the report viewer control.

Analytics architecture

The following architecture diagram shows the online analytical processing (OLAP) cubes that are provided with Microsoft Dynamics AX and the components used to access them.


SQL Server Reporting Services
Data sources
Microsoft Dynamics AX report data provider data extension
Microsoft Dynamics AX query data extension
Microsoft Dynamics AX data method data extension
Reporting Services data extension
Analysis Services data extension
Report Builder
Microsoft Dynamics AX report
Microsoft Dynamics AX report definition customization extension
Visual Studio
Microsoft Dynamics AX Visual Studio tools
Microsoft Dynamics AX client
Menu item
Microsoft Dynamics AX report viewer control (for the client)
Enterprise Portal
Enterprise Portal web page
Microsoft Dynamics AX report viewer control (for Enterprise Portal)
SQL Server Analysis Services
Microsoft Dynamics AX default cube
Excel
Excel data connection wizard

To better understand this diagram, consider how developers, IT professionals, and users access the cubes.

- **Developers** — Developers use the Visual Studio tools that integrate with Microsoft Dynamics AX to build SQL Server Reporting Services (SSRS) reports that use cubes as a data source. In order for such reports to be displayed, the SQL Server Analysis Services (SSAS) data extension retrieves data from the cube, and then the Microsoft Dynamics AX report definition customization extension formats the report.
- **IT professionals** — IT professionals are typically responsible for installing the default cubes that are included with Microsoft Dynamics AX and for processing them on a routine basis.
- **Users** — Users can view the default reports that are included with Microsoft Dynamics AX, or they can create new, customized reports.

Microsoft Dynamics AX includes hundreds of default, preconfigured reports. Users can access these reports using the Microsoft Dynamics AX client or Enterprise Portal. In order for the reports to be displayed, the SQL Server Analysis Services data extension retrieves data from the cube, the Microsoft Dynamics AX report definition customization extension formats the report, and then the Microsoft Dynamics AX report viewer control displays the report to the user.

Users can create new, customized reports by using SQL Server Report Builder or Microsoft Excel®. Each of these applications accesses the cubes directly.

Client architecture

This topic describes the high-level architecture of the Microsoft Dynamics AX Windows client.

The client application is a 32-bit Windows application that provides a rich user interface for the Microsoft Dynamics AX application. The client is typically used by employees within the organization. You can use Enterprise Portal for browser-based access by external users or for users who do not require the rich user interface offered by the Windows client.

Note:
Your network must meet the minimum requirements for latency and bandwidth to get proper performance from the client. Because of the volume of communication that passes between the client and the server, you may experience diminished response time if your network does not meet the minimum requirements. For more information about system requirements, download the system requirements document from the Microsoft Download Center.

Client functionality

The client provides the following functionality:

- **Rich user interface.** The client is a Windows application with a rich user interface consisting of forms, menus, and controls. The client includes over 3,000 forms built using a combination of metadata and X++ code. The Microsoft Dynamics AX forms use X++ to process events and business logic. Forms can host managed WinForms or Windows Presentation Foundation (WPF) controls, and X++ can interoperate with managed (.NET) classes and assemblies.
- **The MorphX development environment.** The development environment is integrated into the client application. Authorized developers can use the MorphX development environment to enhance or customize the Microsoft Dynamics AX application.

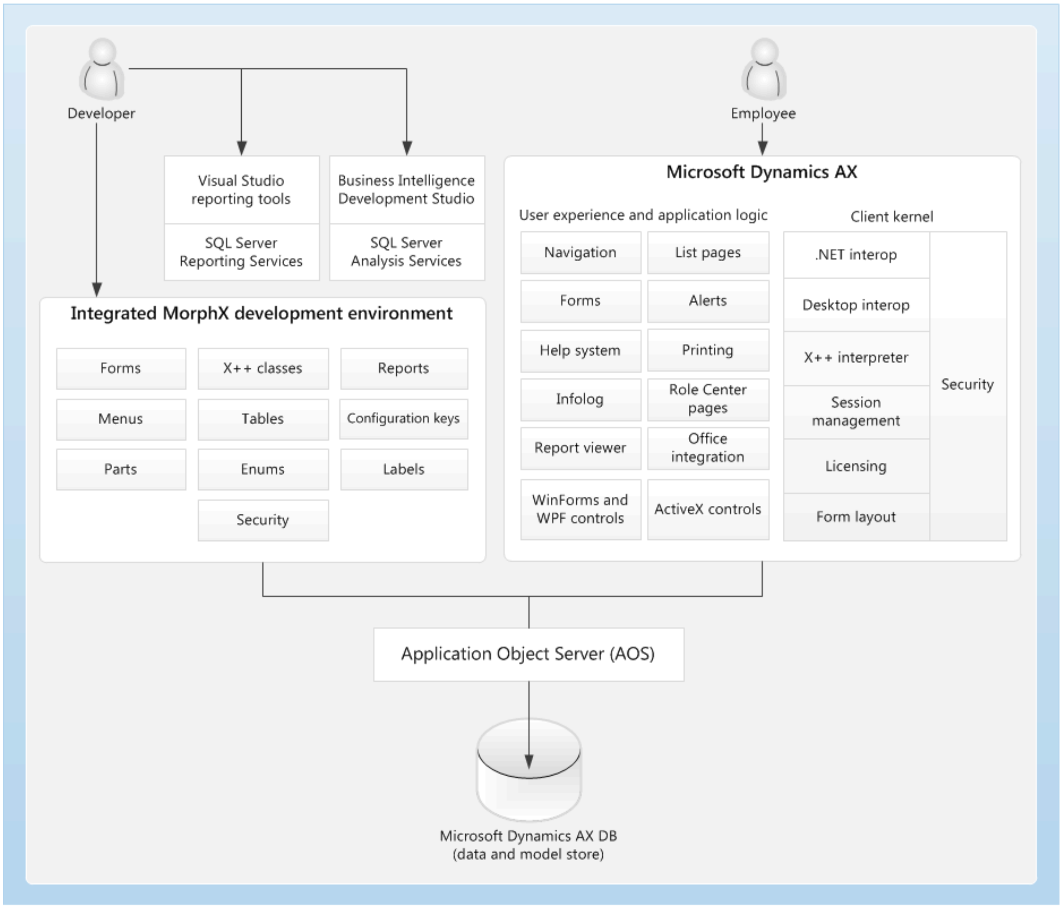
- **Integration with Microsoft Office.** The Microsoft Dynamics AX application integrates with Microsoft Office. Data in grids can be exported to Microsoft Excel, where that data can be formatted, manipulated, refreshed, modified, and saved back into Microsoft Dynamics AX. You can integrate Outlook® with the CRM module to synchronize schedules and tasks bi-directionally.
- **Unified communications.** The client provides integrated unified communications using Microsoft Office Communicator. Key forms and controls are presence-aware for contacts and employees. These forms and controls also provide a visual indicator of the availability of contacts. Users can also use real-time messaging such as instant messaging and outbound voice communication.
- **Integration with the Telephony Application Programming Interface (TAPI).** TAPI is a Windows standard interface for integration between telephone systems and Windows-based software. For example, your application displays information about the caller when you receive a call. The Microsoft Dynamics AX client supports TAPI.
- **Reports.** The Microsoft Dynamics AX application provides reports based on SQL Server Reporting Services (SSRS).

Client/server communication

The client communicates with various Microsoft Dynamics AX components as follows:

- The client uses the remote procedure call (RPC) protocol to directly communicate with the Application Object Server (AOS). The client never accesses the database or metadata directly. The AOS sends the application objects and data to the client.
- Form data sources and queries specified in metadata are the basis of the data layer that is used by the client. In addition, any X++ code that needs to retrieve data can use the built-in language support for querying and manipulating data.
- The client uses a report Web Part to interact with the report server. The report control in the Web Part calls the Web services exposed by the report server to display information that is contained in SQL Server Reporting Services reports. These reports can include either transactional data from the Microsoft Dynamics AX application or OLAP cubes from SQL Server Analysis Services. Cubes provide business analytics and key performance indicators (KPIs).
- The client provides workflow forms, alerts, and controls that users can use to participate in the business process using the Workflow system, a Microsoft Dynamics AX component to enable workflow processes using Windows Communication Foundation classes.
- The client provides a help viewer application that displays context sensitive help topics that are retrieved from an on-premise help server.
- The client also provides role centers, or role-based home pages, for users. These role centers provide role-specific tasks, activities, alerts, reports, and business intelligence to users to increase their productivity. The client uses a browser control to interact with the role centers which are part of Enterprise Portal and are hosted on IIS.

The following diagram describes the high-level client architecture.



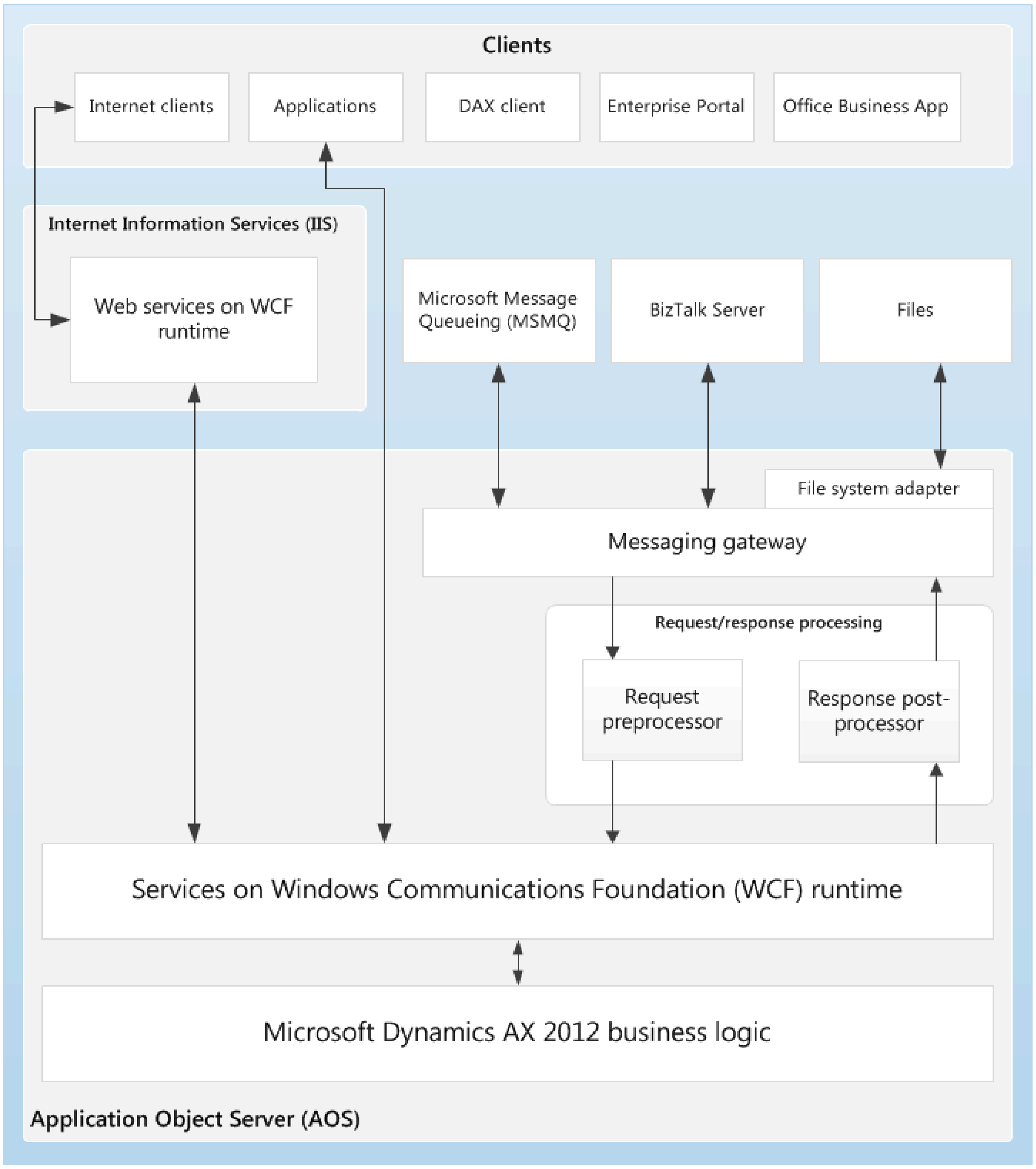
Services and AIF architecture

This topic describes the high-level architecture of services and the Application Integration Framework (AIF).

Microsoft Dynamics AX exposes its functionality through Windows Communication Foundation (WCF)-based services that are hosted on the Application Object Server (AOS). External applications and client applications on the local area network consume Microsoft Dynamics AX services by directly accessing them from the AOS. These clients and applications include the Microsoft Dynamics AX client, the Office add-ins, and Microsoft Dynamics AX components such as the Enterprise Portal. Internet-based external applications and clients access the Microsoft Dynamics AX services through Internet Information Services (IIS). IIS routes the incoming Microsoft Dynamics AX service requests to the AOS. Regardless of the origin of the service request, all the requests are handled by the WCF-runtime that is hosted on the AOS.

The AIF request preprocessor, if configured, can intercept the inbound request messages for custom preprocessing such as message transforms or value substitutions. The Microsoft Dynamics AX service invokes the necessary business logic to process the inbound request message. Similarly, the AIF response postprocessor, if configured, can intercept the outbound response messages for custom post-processing such as message transforms or value substitutions before returning the response to the client.

The following diagram describes the services and AIF architecture.


Clients
Internet clients
Applications
DAX client
Enterprise Portal
Office Business App
Internet Information Services (IIS)
Web services on WCF runtime
Microsoft Message Queueing (MSMQ)
BizTalk Server
Files
File system adapter
Messaging gateway
Request/response processing
Request preprocessor
Response post-processor
Services on Windows Communications Foundation (WCF) runtime
Microsoft Dynamics AX 2012 business logic
Application Object Server (AOS)

.NET Business Connector architecture

The .NET Business Connector architecture

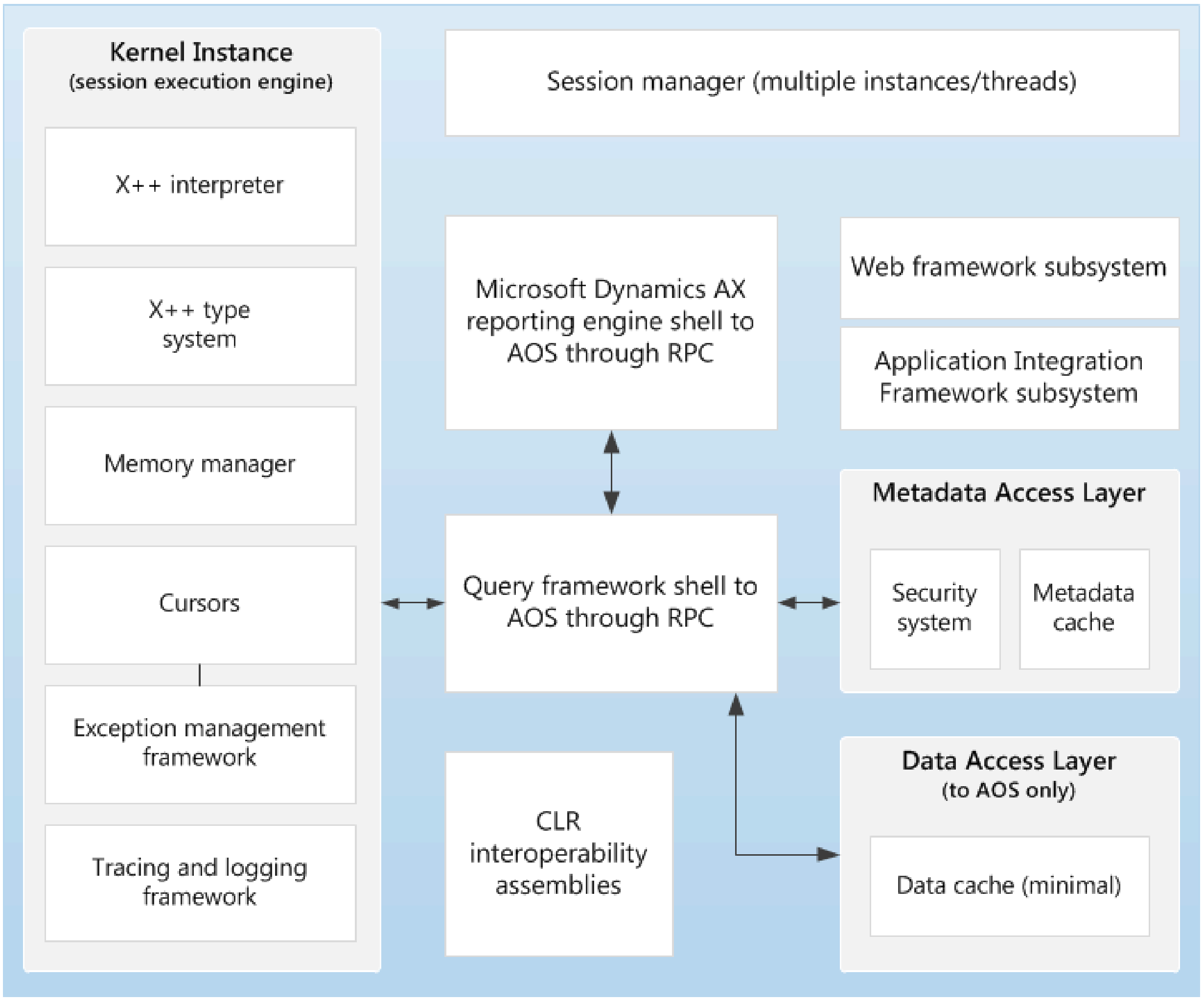
The .NET Business Connector is a component of the development environment for Microsoft Dynamics AX. You use the .NET Business Connector to build software applications that can be integrated with Microsoft Dynamics AX. You can think of the .NET Business Connector as a Microsoft Dynamics AX client that does not have a user interface. You can use the .NET Business Connector to access the same X++ code, business logic, and security model that are available to the Microsoft Dynamics AX client. The .NET Business Connector contains a kernel that is used to interpret and execute code, and provides a runtime environment for interacting with elements in the Application Object Tree (AOT). For more information about the .NET Business Connector, see Developer documentation on MSDN®.

During execution, applications that are created by using the .NET Framework are managed by the common language runtime (CLR). These applications are called managed applications. The .NET Business Connector enables these managed applications to interact with instances of an Application Object Server (AOS) by providing a set of .NET managed classes. These .NET managed classes, in turn, enable access to X++ classes in Microsoft Dynamics AX.

By default, the .NET Business Connector is installed together with the Application Integration Framework (AIF). However, the .NET Business Connector can also be installed as a stand-alone component and used to develop third-party applications that can be integrated with Microsoft Dynamics AX.

For more information about how to integrate other applications with Microsoft Dynamics AX, see Application integration. For information about how to call .NET methods from X++ code, see Developer documentation on MSDN.

The following diagram shows the Business Connector architecture.


Kernel Instance
(session execution engine)
X++ interpreter
X++ type system
Memory manager
Cursors
Exception management framework
Tracing and logging framework
Session manager (multiple instances/threads)
Microsoft Dynamics AX reporting engine shell to AOS through RPC
Web framework subsystem
Application Integration Framework subsystem
Metadata Access Layer
Security system
Metadata cache
Query framework shell to AOS through RPC
CLR interoperability assemblies
Data Access Layer
(to AOS only)
Data cache (minimal)

Project Server integration architecture

By setting up integration between Microsoft Dynamics AX 2012 and Microsoft Project Server, project managers can create projects in either of the products and synchronize project data between them. This solution enables you to draw on both the core project management capabilities in Project Server and the financial management capabilities in Microsoft Dynamics AX.

For more information about setting up integration with Project Server, see the Microsoft Project Server 2010 Integration white paper that is available from the Microsoft Download Center.

Development environment

Depending upon the task, developers can use MorphX or Microsoft Visual Studio for development of Microsoft Dynamics AX. Use MorphX to work with the data model, the user interface such as forms, and the application code. Use Visual Studio and Visual Studio Tools for Microsoft Dynamics AX for development in managed code such as customization of the Enterprise Portal.

The following sections describe the Microsoft Dynamics AX development using MorphX and Visual Studio.

Development with the MorphX IDE

MorphX is an integrated development environment (IDE) that developers use to work with the application metadata, design Microsoft Dynamics AX forms, and to edit and compile X++ code within a single interface.

Some of the architectural elements and foundational technologies in Microsoft Dynamics AX that MorphX relies on are described in the following list:

- **The Application Object Tree (AOT)**. The AOT contains definitions of all the model elements that are used to build Microsoft Dynamics AX, such as classes, tables, forms, and so on. Developers use the AOT for many tasks, including:
  - Drag-and-drop development. For example, developers can drag a table on a form to associate the table as a data source for that form.
  - Edit properties. Developers use the AOT to set properties of the model elements such as fields, tables, classes, forms, and so on.
- **X++ language**. X++ is an object-oriented language, with similarities to C#. X++ can also incorporate SQL data manipulation statements.
- **Application Object Layers**. Layers constitute a hierarchy of levels in the application. Layers include metadata and code. These enable developers to make modifications and additions without interfering with the application objects in a lower level. When developers make an object modification on one level, the modification overshadows the object on a lower level. Developers could, for example, decide to add email information to a standard form. The addition would be saved on your level only. The revised form replaces the standard form. However, you can always return to the original form at the next lower level by removing the new form.
- **Enterprise Portal development**. The resources for Enterprise Portal are stored in the AOT. Some development tasks are performed directly in the AOT, such as defining data sets, business logic, security, site structure, and navigation for Enterprise Portal. Developers can use web-based development tools in Microsoft® SharePoint® 2010 technology for basic customizations. For advanced customizations, use Enterprise Portal framework and Visual Studio.

- **Report development**: Developers create and use queries in the AOT to access data for reports. The AOT stores reports and Visual Studio projects that are created in Visual Studio when developers define reports.
- **Analysis cube development**: Developers define perspectives and analysis properties using the AOT. These perspectives are used to generate SQL Server analysis services projects and to deploy these projects as analysis cubes.

For more information about MorphX, see Microsoft Dynamics AX IDE.

Development with Visual Studio

The Microsoft Dynamics AX development environment, MorphX, and the Visual Studio development environment are integrated through Visual Studio Tools for Microsoft Dynamics AX. Visual Studio Tools for Microsoft Dynamics AX is a collection of tools that enable managed code development for Microsoft Dynamics AX. These tools provide developers with a rapid application development (RAD) experience in the Visual Studio IDE. The Visual Studio Tools for Microsoft Dynamics AX include:

- **Managed code support**. Developers can write managed code to customize components such as reporting and the Enterprise Portal. Developers can develop business logic in managed code. Managed code refers to code that executes under the management of the Common Language Runtime.

  Developers can use type safe proxies to call X++ classes from managed code, bringing the ease of use of X++ classes on a par with the managed classes. Visual Studio Intellisense works with the proxies that you add to the Visual Studio project. X++ code can also call managed code. These capabilities provide a seamless interoperability between X++ and managed code.
- **Report development**. Developers can create reports for Microsoft Dynamics AX using Visual Studio and Reporting Services. For more information about reporting, see Creating Reports for Microsoft Dynamics AX in Visual Studio.
- **Enterprise Portal development**. Developers can use web-based development tools in SharePoint for basic customizations or Enterprise Portal framework and Visual Studio for advanced customization.

Close integration between MorphX and Visual Studio means that developers can leverage the benefits of each development tool and work in the environment that best suits their development scenario.

Application integration

The ability to integrate Microsoft Dynamics AX with other systems inside and outside the enterprise is a common requirement. Application Integration Framework (AIF) enables such integration by enabling the exchange of data through formatted XML. This formatted XML is referred to as a *document*, and each document contains *data* and *business logic*. Documents are based on a document class and defined by using Microsoft Dynamics AX.

Microsoft Dynamics AX ships with over 70 standard documents that support common business processes. AIF also provides the ability to customize existing documents or create your own documents. For more information about the standard documents, see topic "Documents that ship with Microsoft Dynamics AX" in the Microsoft Dynamics AX SDK Help. For developer information about how to create new AIF documents, see topic "Creating New Documents" in the Microsoft Dynamics AX SDK Help.MS

How documents are exchanged

AIF provides an extensible framework that supports multiple asynchronous transports, as well as synchronous transport using Web services, to exchange documents in XML format with external systems.

AIF can be used to either send data into Microsoft Dynamics AX, called an *inbound* exchange, or retrieve data during an *outbound* exchange. An example of an inbound exchange would be an external system sending a sales order to be saved to the Microsoft Dynamics AX database. An example of an outbound exchange would be an external system sending a request for a purchase order and receiving the purchase order back. The inbound and outbound exchanges can be categorized as follows:

- **Send data** - Microsoft Dynamics AX sends documents to an external system.
- **Send data in response to requests** - Microsoft Dynamics AX receives requests for documents from another authorized system, retrieves the requested information (such as a document or a list of documents) from the Microsoft Dynamics AX database, and returns it to the requesting system, with appropriate filtering and security. The request message would contain the entity keys or a query that specifies the data that the external system is requesting.
- **Receive and create data** - Microsoft Dynamics AX receives documents from another authorized system and creates new records in the Microsoft Dynamics AX database.

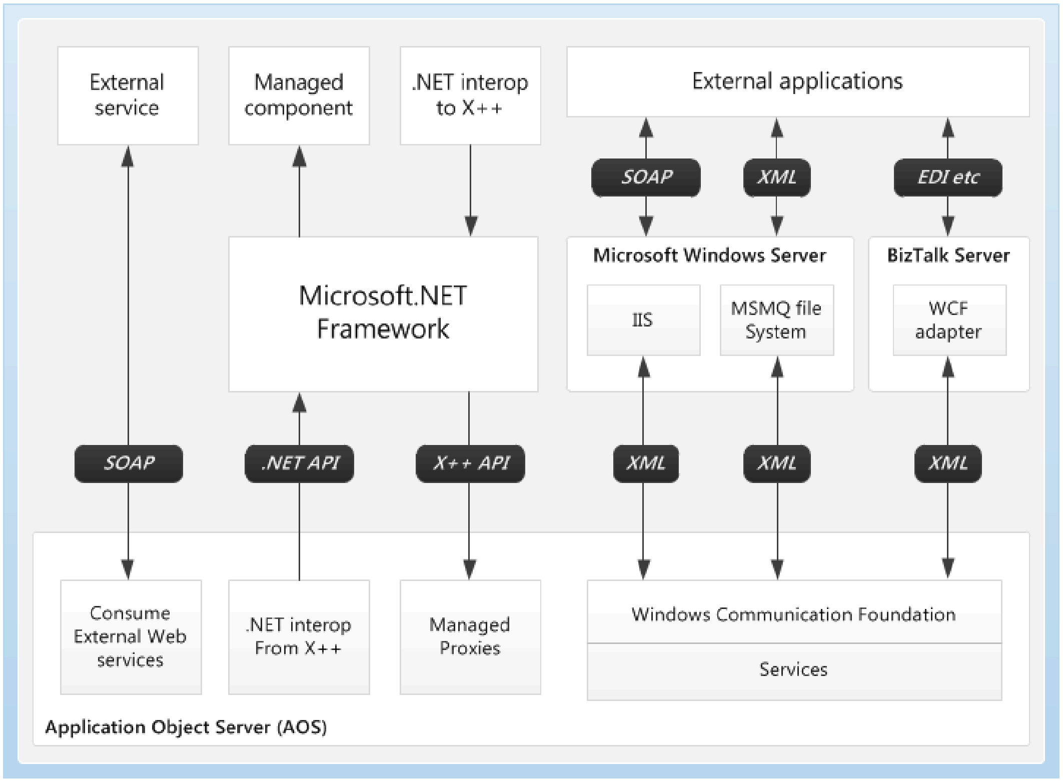
Integrating with applications

This section describes the ways you can integrate Microsoft Dynamics AX with other applications. For a high-level overview of the Microsoft Dynamics AX system architecture, see System architecture.

You can use the following methods to integrate Microsoft Dynamics AX with other applications:

- **Services and the Application Integration Framework (AIF)**. Services are the preferred option for integration with Microsoft Dynamics AX. Services enable Microsoft Dynamics AX to expose its functionality through Windows Communication Foundation-based services. Microsoft Dynamics AX code and external applications can consume Microsoft Dynamics AX services. AIF supports the processing of inbound and outbound messages such as message transforms and value lookups. Together, services and AIF provide the programming model, tools, and infrastructure support for XML-based integration with external applications and data
- **.NET interop to X++**. The .NET interop to X++ feature enables you to call X++ code using C# or another managed language. A proxy is an automatically generated .NET class, in C# or another managed language, that mimics an X++ class of Microsoft Dynamics AX.
- **.NET interop from X++**. The .NET interop from X++ (also known as the CLR Interop in the previous release) provides interoperability with external .NET components and enables you to execute managed-code components from within X++ code. .NET interop from X++ is useful when you want your X++ code to access functionality provided by a CLR-managed assembly. This includes assemblies that are installed with the .NET Framework and any assemblies that you create with a language such as Visual C#® or Visual Basic®.
- **Consume external Web services**. The Microsoft Dynamics AX programming model lets you consume external Web services from within X++ code. To consume an external Web service from X++, you must first create a reference to the Web service. After creating a reference to the Web service, you can invoke it from X++ and see the available methods with IntelliSense. Calling and managing external Web services is done completely within Microsoft Dynamics AX.

The AIF integration components interact with Microsoft Dynamics AX through the Application Object Server (AOS), as shown in the following diagram.



See Also

[Visual Studio Integration]

[Microsoft Dynamics AX Developer Center]

Plan an implementation

This section provides information about hardware and software requirements, security, and other components so that you can plan your implementation.

Implementation methodology

Microsoft Dynamics Sure Step is the prescribed methodology for deploying Microsoft Dynamics AX. The Sure Step application provides product-specific and general project-based templates, workflows, process maps and tools to assist the implementation partners. Sure Step is currently available for download from PartnerSource. (This website requires that you log in.)

The Sure Step methodology is divided into the following phases:

Phase	Tasks during phase
Diagnostic	• Evaluate a customer's business processes and infrastructure • Assist the customer with their due diligence cycle, including ascertaining requirements and their fit with the solution, and assessing the resource needs for the solution delivery • Prepare the project plan, proposal, and the **Statement of Work**
Analysis	• Analyze current business model and finalize the **Functional Requirements** document • Finalize the fit-gap analysis • Develop the **Environment Specification** documentation
Design	• Develop the **Functional Design**, **Technical Design**, and **Solution Design** documents • Finalize the data migration design • Establish test criteria
Development	• Finalize configurations and setup of the standard solution • Develop and finalize the custom code that is required to support the solution • Conduct functional and feature testing of the solution • Create the user training documentation
Deployment	• Set up the production environment • Migrate data to the production environment • Conduct user acceptance test of the system • Train users and finalize the user documentation • Conduct go-live check and promote the system to production
Operation	• Resolve pending issues • Finalize user documentation and knowledge transfer • Conduct a post-mortem of the project • Provide on-going support (activities that continue through any future involvement with the customer after the project is closed)

The Sure Step methodology also provides guidance for the following areas:

Activity area	Actions performed
Optimization	• Leverage **Review Offerings** to determine proactively if the system is being designed and delivered optimally to meet the customer's requirements • Analyze the system to determine how it can be optimized for the best performance based on customer's needs
Upgrade	• Assess the customer's current business processes and solution • Document the requirements for new functionality • Upgrade the system to new release—including the addition of new functionality, promotion of existing customizations that are required, and elimination of custom code no longer required

Hardware and software requirements

For up-to-date hardware and software requirements for Microsoft Dynamics AX, download the system requirements document from the Microsoft Download Center.

Planning hardware infrastructure

This topic describes key factors to be taken into account when you plan the hardware infrastructure for Microsoft Dynamics AX.

Planning Hardware

Decisions about appropriate hardware depend upon a number of factors. The following list provides some key factors.

1. Evaluate and document the existing infrastructure, including:
   - Network bandwidth
   - Storage system in use
   - Operating system in use
   - Databases in use
   - Servers in use
   - Current processes for disaster recovery, availability, and scalability
   - Existing applications that need to integrate with Microsoft Dynamics AX
2. Define and document:
   - Uses of system: Components and modules of Microsoft Dynamics AX you plan to deploy
   - Number of transactions over a period of time as well as total number of transactions during peak business hours
   - Number of active or concurrent users over a period of time as well as total number of active or concurrent users during peak business hours

- External user access required
- Web access required
- Required availability
- Projected growth rate
- Number of sites and number of users connecting through a Wide Area Network (WAN)
- Integration requirements: Are there any applications that need to integrate with Microsoft Dynamics AX and what is the workload generated by these applications? Are these transactions real-time or can they be batched?

3. With information from Steps 1 and 2, you can start to determine how to structure the system. Key decisions are:
    - Whether any Microsoft Dynamics AX server components can be combined on a single computer, and if they can, which server components to combine.
    - What is your deployment plan for high availability and scalability for Microsoft Dynamics AX components?
    - What is your backup and recovery strategy?

Transactional Volume

The total average number of transactions processed per work hour is a key indicator of the hardware and software requirements. Use the transactional volume to plan your hardware and software components such as those listed in the following list:

- The database server infrastructure, including type and number of drives
- Number of Application Object Server (AOS) clusters
- Number of AOS servers within a cluster
- Number of batch servers
- Network capacity

In Microsoft Dynamics AX, a transaction is defined as processing of a single line item. For example, a sales order with 100 line items is considered as 100 transactions.

Estimate the number of transactions required for each module you plan to use and any corresponding transactions that may be triggered by these changes. Determine if there are any integration points to internal or external applications. For example, a large volume of transactions may be coming in from Microsoft BizTalk Server. This volume of transactions needs to be factored into your infrastructure and topology planning.

Determine if these transactions are real-time or if they can be batched and processed at off-peak hours. Microsoft Dynamics AX is an integrated ERP application and provides real-time updates throughout all modules as information is changed. However, it also provides a batch system for scheduled processing.

Number of Concurrent Users

The total number of concurrent users is another indicator of the size of the Application Object Server (AOS) system required for proper response time and throughput. While this is not the only criterion used to plan the capacity of AOS servers or server clusters, it is an important factor.

Concurrent users are defined as Microsoft Dynamics AX rich clients, web clients, mobile clients, or third-party applications that require some processing to take place in the Microsoft Dynamics AX system. Note that the number of concurrent users also impacts your network bandwidth and latency.

Network Requirements

Determine the number of users accessing Microsoft Dynamics AX with the rich client, web client, or mobile client. Users accessing Microsoft Dynamics AX using the rich client must meet minimum network requirements. If those requirements are not met, consider deploying Windows Server Terminal Services.

Planning hardware for additional components

The core components of a Microsoft Dynamics AX implementation consist of a Windows client, Application Object Server (AOS), and a database server. Additional components include Enterprise Portal, workflow, reporting, analytics, Help server, and IIS-based web services. Determine the workload generated for each component and the resource requirements for appropriate deployment with acceptable response time and throughput.

For example, if you have users who access Microsoft Dynamics AX over a WAN using the Microsoft Dynamics AX Windows client, you must deploy Terminal Services. Similarly, users who access role-based home pages will create workload for the Enterprise Portal. Users accessing reports will create workload for the Microsoft SQL Server 2008 Reporting Services report server, the report server database, and the Microsoft Dynamics AX database.

Install Microsoft Dynamics AX

The Microsoft Dynamics AX Installation Guide provides step-by-step installation instructions for deployment of Microsoft Dynamics AX components.

Upgrade

The Upgrade Guide provides the information to upgrade from the previous releases of Microsoft Dynamics AX.