

A blue abstract graphic with wavy, horizontal lines, transitioning from a darker blue on the left to a lighter blue on the right, positioned at the top of the page.

Microsoft Dynamics® AX 2012

# Services in Microsoft Dynamics AX 2012

White Paper

This white paper gives an overview of different types of services available in Microsoft Dynamics AX 2012 and provides guidance to developers and architects in deciding when to use a specific service type. The white paper assumes familiarity with the various programming models in Microsoft Dynamics AX 2012 and with the basics of the Services and Application Integration Framework.

<http://microsoft.com/dynamics/ax>

Date: February 2011

Author: Nitin Sharma, Senior Program Manager

Send suggestions and comments about this document to [adocs@microsoft.com](mailto:adocs@microsoft.com). Please include the title with your feedback.

---

## Table of Contents

<b>Introduction</b> .....	<b>3</b>
<b>Audience</b> .....	<b>3</b>
<b>Prerequisites</b> .....	<b>3</b>
<b>Purpose</b> .....	<b>3</b>
<b>Overview of the Microsoft Dynamics AX 2012 services</b> .....	<b>3</b>
Document services.....	3
Custom services .....	4
System services.....	5
How to distinguish between the three key service types.....	5
<b>Exposing business entities as services</b> .....	<b>6</b>
<b>Conclusion</b> .....	<b>7</b>

---

## Introduction

Microsoft Dynamics® AX 2012 provides many services that can be used by the Microsoft Dynamics AX components and with third-party applications. Services are programmable artifacts in Microsoft Dynamics AX 2012. The inventory of services shipped in Microsoft Dynamics AX 2012 includes X++ and managed code services. Developers can alter existing services and create new X++-based services.

All customizable application and framework X++ services in Microsoft Dynamics AX 2012 are compiled to Microsoft® intermediate language (MSIL). They can be grouped together and deployed as *ports*, which is a new concept introduced in Microsoft Dynamics AX 2012. A port represents a Windows® Communication Foundation (WCF) service host, and all services that it hosts are WCF endpoints. By default, a port is hosted on the Application Object Server (AOS) or on Internet Information Services (IIS) if HTTP is chosen as the adapter.

Microsoft Dynamics AX 2012 also ships with non-customizable system services that aid application development.

## Audience

This white paper targets developers and architects who are planning to use services in Microsoft Dynamics AX 2012.

## Prerequisites

To benefit from this white paper, you must understand the following areas:

- Programming models in Microsoft Dynamics AX 2012. For more information, see the white paper "Selecting the Best Development Technology for your Application Development Scenario\_AX2012", which is also on the Microsoft Connect site.
- Basics of Microsoft Dynamics AX 2012 Services and Application Integration Framework.

## Purpose

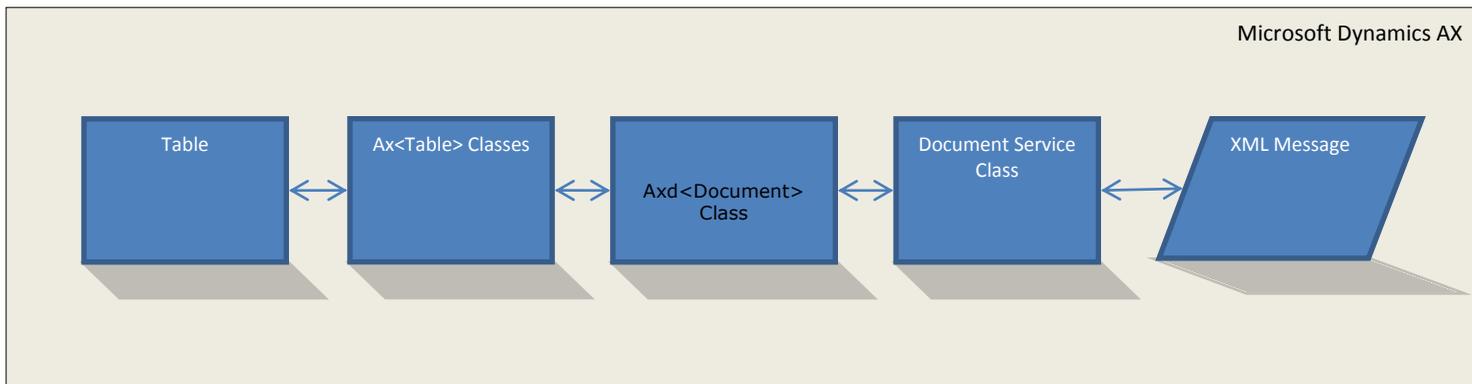
The purpose of this white paper is to provide an overview of services available in Microsoft Dynamics AX 2012 and to provide guidance about the types of services that are best suited for a given scenario.

## Overview of the Microsoft Dynamics AX 2012 services

The following services are available in Microsoft Dynamics AX 2012.

### Document services

Introduced in Microsoft Dynamics AX 4.0, documents represent business objects, such as customers, vendors, and sales orders. Document services enable you to exchange data with external systems by sending and receiving XML documents. The following figure shows the internal details of a document service.



The **DocumentService** class exposes the operations of the service. The **DocumentService** class calls the functionality in **Axd <Document>** classes that contain any customizations to the business logic for modeled entities. For example, the **AxdCustomer** class contains the customization logic to create, update, or delete a customer entity. **Axd <Document>** classes can use **Ax<Table>** classes to create, update, or delete document tables, or optionally to apply customizations during read operations.

Document services are generated from a query that models the business entity. For example, a customer query will run the Document Service Generation Wizard to create the service artifacts. Document services include logic to handle the details of how a query will be serialized or de-serialized from XML that is sent across a network. For example, the document services framework can serialize and de-serialize dimension attributes and perform natural key substitution. The framework deals with the polymorphic nature of queries, the date effective fields, and so on. All the data-layer improvements have been handled in the framework. The creator of a document service only has to make sure that the business entity is properly mapped to the query object.

All artifacts produced in a document service are customizable so that a developer can further fine-tune the functionality based on the requirements.

### Synchronization APIs

Two new APIs have been introduced in the document services framework to enable the synchronization of entities between Microsoft Dynamics AX 2012 and external systems. These APIs are not present in the preinstalled document services or in the Document Service Generation Wizard. Developers must extend this base support to enable their own scenarios.

- **getKeys** method: Similar to the existing **findKeys** method, but applies the document filters that have been configured for the service and port. Filters provide a mechanism for restricting the overall set of documents that will be returned by a document service. For example, a filter on a customer document on the **Name** field would return only those customers whose name matches the filter.
- **getChangedKeys** method: Extends the **getKeys** method by adding a parameter that contains a date and time. The values returned are the ones that changed in the root data source or any child data source of the document after the specified date and time.

### Custom services

Custom services in Microsoft Dynamics AX 2012 allow developers to expose any X++ logic through a service interface. Custom services were supported in Microsoft Dynamics AX 2009, but the framework support for some data types was limited. For any complex type, developers were required to write custom serialization and de-serialization logic.

---

Microsoft Dynamics AX 2012 provides standard attributes that can be set on the **DataContract** class and its members to make them serialize and de-serialize data that is sent and received across a network connection. Many predefined types, such as **Collections** and **Tables**, are also supported. This feature enables you to expose any existing X++ class and its members as part of a service operation without writing new code, just by setting a few attribute values. The syntax is similar to using data contract attributes in WCF.

## System services

System services are a new category of services that have been added in Microsoft Dynamics AX 2012. These services are not customizable and are not mapped to any query or X++ code. These services are written in managed code and hosted on the AOS at a fixed address specified in the server configuration file.

### Query Service

Query Service exposes APIs to execute any Application Object Tree (AOT)-defined or ad hoc client-defined query against the system. It can also execute a dynamically generated query by using X++. The results are returned as an ADO.NET **DataSet** object. This feature is especially useful in creating Microsoft .NET Framework applications that use controls that can be bound to the returned result.

The URL for Query Service is as follows

```
net.tcp://<hostname:port>/DynamicsAX/Services/QueryService
```

where *hostname* and *port* are on the computer running the AOS.

The caller of Query Service needs to have appropriate permissions to retrieve all the requested information. If certain fields in the requested set are not permitted, they are trimmed. If any data source in the join list is not permitted, the service call fails with an error.

### Metadata Service

Metadata Service enables clients to get information about some key metadata in the AOT. Clients can get information about which queries or services exist in the AOT and various details about each instance. This feature is useful in scenarios in which some metadata details need to be checked or published by an application. Another important scenario is retrieving labels for different locales for building and displaying custom user interfaces.

The URL for Metadata Service is as follows

```
net.tcp://<hostname:port>/DynamicsAX/Services/MetadataService
```

where *hostname* and *port* are on the computer running the AOS.

### User Session Service

User Session Service allows the caller to retrieve information about the calling user. This includes details such as the user's company, timezone offset, and locale.

The URL for User Session Service is as follows:

```
net.tcp://<hostname:port>/DynamicsAX/Services/UserSessionService
```

where *hostname* and *port* are on the computer running the AOS.

## How to distinguish between the three key service types

The following table provides more information about how to distinguish between the three key service types, based on certain deployment and usage parameters.

	Document services	Custom services	System services
<b>Service and data contracts</b>	Auto-generated based on the query and wizard options	Developer defined	Fixed and cannot be updated
<b>Deployment mode</b>	Ports	Ports	Stand-alone services running on fixed address on AOS
<b>Hosted on</b>	AOS/IIS	AOS/IIS	AOS only
<b>Transport protocol</b>	Any supported adapter	Any supported adapter	Only net.tcp
<b>Usage</b>	Use to expose a business entity through a service interface. Used in integration. All AIF framework artifacts can be applied on these services.	Use to expose any custom business logic (even a business entity) through a service interface.  AIF framework support for transforms is present but not for pipelines, schema constraints, or others.	Use always on utility services for any caller.

## Exposing business entities as services

The document services framework provides a simple wizard flow to generate a document service from the query that defines the business entity. This was the recommended way to create entity-based services in the Microsoft Dynamics AX 2009. With further enhancements in the document services framework in supporting all new data-layer enhancements, this approach should still be the first choice for anything entity-related that needs to be exposed through web services.

Custom services in Microsoft Dynamics AX 2009 were intended to be used for exposing any custom logic through services. But exposing any complicated business entity was not possible because of the lack of serialization and de-serialization support for service operation parameters. However, the introduction of custom attributes in Microsoft Dynamics AX 2012 has made it easy to build a data-contract-based service.

The following table contains information that can help you to decide whether to use document services or custom data contract-based services.

	Document services	Custom services
Entity complexity	The Document Service Generation Wizard handles queries of any complexity and size and generates the service seamlessly.  For example, some queries contain data sources and relationships that include dimensions, polymorphism, date-effective information, and so on. The document framework handles these types of queries seamlessly.	Suitable when the entity complexity is low. The schema can be written as a data contract class and relevant data member attributes set.  If tables and relationships are used in a data contract, then any feature around dimensions, polymorphism, and so on, needs to be handled in code by the developer. This task may be time consuming.

Performance requirements	The incoming XML is passed to the document services framework, which parses the XML to validate the AxD schema, and then invokes the relevant operation. The framework enables customization to handle any complex pattern in the underlying query.	Custom services use the underlying .NET XML Serializer to serialize and de-serialize the XML into a data contract object. No other logic is present. For simple entity schemas, this approach will be faster than document services. For complex schemas, you might need to write a lot of custom code.
Integration requirements	All integration stack elements such as pipelines, transforms, and schema constraints can be applied to document services.	Schema constraints and value substitution are not honored for custom services. However, transforms that convert between the AifXMLMessage format and other formats and pipelines for preprocessing and postprocessing of XML messages can be used.
Flexibility in service contracts	Because a document service is derived from a query, any change to the query object or its data source schema is likely to change the service contract. This tight coupling between the service contract and the underlying query-table schema could be limiting for certain scenarios.	The data contracts are written by developers and can be controlled to make sure that the underlying schema changes do not have any effect on them. The control that a developer has in defining the service contracts can be good for service clients.
Microsoft Office add-in support	Office add-in tools, shipped as a part of Microsoft Dynamics AX 2012, have built-in support for consuming document services for updating data.	No preinstalled integration with Office add-ins exists.

## Conclusion

The following table summarizes some of the key scenarios for all the service types in Microsoft Dynamics AX 2012.

Services	Common Scenarios
<b>Custom services</b>	Use for exposing very simple business entities as a service.
	Use for exposing any custom logic from Microsoft Dynamics AX.
<b>Document services</b>	Use for easily exposing business entities with varying complexity as a service.
	Use for supporting any medium to complex integration scenario involving entities.
<b>Query Service</b>	Use for implementing read operations for any Microsoft Dynamics AX, ad-hoc, or existing AOT query.
<b>Metadata Service</b>	Use for retrieving certain key metadata information from Microsoft Dynamics AX.
<b>User Session Service</b>	Use for retrieving any user session information.

---

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](http://www.microsoft.com/dynamics)

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

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

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

© 2011 Microsoft Corporation. All rights reserved.

Microsoft, the Microsoft Dynamics Logo, Microsoft Dynamics, Windows, and Excel are trademarks of the Microsoft group of companies.

All other trademarks are property of their respective owners.

**Microsoft**