



What's New in Microsoft Dynamics AX 2012 for Developers

Microsoft Corporation

Published: August 2011

Microsoft Dynamics AX

Microsoft Dynamics is a line of integrated, adaptable business management solutions that enables you and your people to make business decisions with greater confidence. Microsoft Dynamics works like and with familiar Microsoft software, automating and streamlining financial, customer relationship and supply chain processes in a way that helps you drive business success.

U.S. and Canada Toll Free 1-888-477-7989

Worldwide +1-701-281-6500

www.microsoft.com/dynamics

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

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

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

© 2011 Microsoft Corporation. All rights reserved.

Microsoft, Microsoft Dynamics, Active Directory, Excel, Hyper-V, Internet Explorer, Outlook, SharePoint, SQL Server, Visual Studio, Windows, Windows PowerShell, Windows Server, and Windows Vista are trademarks of the Microsoft group of companies. All other trademarks are property of their respective owners.

Table of Contents

| | |
|--|----------|
| What's New in Microsoft Dynamics AX 2012 for Developers | 4 |
| What's New: Analytics for Developers in Microsoft Dynamics AX 2012 | 8 |
| What's New: AOS and Database for Developers in Microsoft Dynamics AX 2012 | 11 |
| What's New: Best Practices for Developers in Microsoft Dynamics AX 2012 | 16 |
| What's New: Client for Developers in Microsoft Dynamics AX 2012 | 27 |
| What's New: Enterprise Portal for Developers in Microsoft Dynamics AX 2012..... | 33 |
| What's New: Help System for Developers in Microsoft Dynamics AX 2012..... | 38 |
| What's New: MorphX Development for Developers in Microsoft Dynamics AX 2012..... | 41 |
| What's New: MorphX Tools for Developers in Microsoft Dynamics AX 2012 | 46 |
| What's New: Reporting for Developers in Microsoft Dynamics AX 2012..... | 52 |
| What's New: Security for Developers in Microsoft Dynamics AX 2012..... | 59 |
| What's New: Services & AIF for Developers in Microsoft Dynamics AX 2012 | 62 |
| What's New: Visual Studio Tools for Developers in Microsoft Dynamics AX 2012 | 70 |
| What's New: Workflow for Developers in Microsoft Dynamics AX 2012..... | 75 |
| What's New: X++ for Developers in Microsoft Dynamics AX 2012 | 79 |

What's New in Microsoft Dynamics AX 2012 for Developers

This document describes the new and enhanced features for developers that are available in Microsoft Dynamics AX 2012. Updated and additional developer documentation for Microsoft Dynamics AX can be found at the [Microsoft Dynamics AX Developer Center](#) on the Microsoft Developer Network (MSDN). For information on the new, changed, and deprecated features in Microsoft Dynamics AX 2012, see the [New, Changed, and Deprecated Features for Microsoft Dynamics AX 2012](#) document.

The following feature areas are described in this document:

| Title | Description |
|---|---|
| What's New: Analytics for Developers in Microsoft Dynamics AX 2012 | <p>The Analysis Services project wizard makes it easier to create, update, configure, and deploy Analysis Services projects. The wizard performs many of the tasks that you would otherwise have to perform manually in Microsoft Business Intelligence Development Studio (BIDS).</p> <p>You can use Microsoft Dynamics AX views to model data in perspectives.</p> <p>You can create Gregorian calendars to use as date dimensions in your Analysis Services project.</p> |
| What's New: AOS and Database for Developers in Microsoft Dynamics AX 2012 | <p>The following Application Object Server (AOS) features provide functionality for advanced applications or make it easier to write applications:</p> <ul style="list-style-type: none">Valid time state tables that contain date effective data.The UnitOfWork class to simplify database transaction management.Inheritance among tables, similar to class inheritance.Full text index support to quickly query words that are embedded in the middle of a string field of a table.TempDb tables are a second type of temporary table, in addition to the legacy InMemory temporary table. TempDb tables can be joined to regular tables in a query.Computed columns in a view.Filters on queries that perform an outer join.Disabled tables can be retained in the underlying |

| | |
|---|---|
| | database for access by external applications. |
| What's New: Best Practices for Developers in Microsoft Dynamics AX 2012 | <p>Best practice compiler-enforced checks are updated to reflect new features added for this release. Emphasis is on security and evaluating existing rules that generate warnings. Warnings that require action have been promoted to errors. Minor warnings that are typically ignored have been removed.</p> <p>A form style checker tool is integrated into the Best Practice tool.</p> |
| What's New: Client for Developers in Microsoft Dynamics AX 2012 | New forms, controls, and data source features are added to help you customize the Microsoft Dynamics AX 2012 Windows client. |
| What's New: Debugging for Developers in Microsoft Dynamics 2012 | For more information about debugging scenarios in Microsoft Dynamics AX 2012, see Debugging in Microsoft Dynamics AX 2012. |
| What's New: Enterprise Portal for Developers in Microsoft Dynamics AX 2012 | Enterprise Portal is now based on Microsoft SharePoint 2010. New user controls and support for Microsoft Dynamics AX 2012 features significantly enhance the usability. The developer experience has been improved. |
| What's New: Help System for Developers in Microsoft Dynamics AX 2012 | You can create and publish new Help documentation. The new Help server simplifies how you update and customize Help documentation. |
| What's New: MorphX Development for Developers in Microsoft Dynamics AX 2012 | <p>New and enhanced features are available in the MorphX development environment. This topic describes:</p> <ul style="list-style-type: none"> · Models and the model store · Object IDs · The AxUtil command line utility · The new development environment · Installation-specific IDs |
| What's New: MorphX Tools for Developers in Microsoft Dynamics AX 2012 | New and enhanced development tools are available in the MorphX Integrated Development Environment (IDE). |
| What's New: Organization Model for Developers in Microsoft Dynamics AX 2012 | You can extend organization model to support ISV or partner customization or configuration scenarios. You can define new custom operating unit types. |
| What's New: Reporting for Developers in Microsoft Dynamics AX 2012 | SQL Server Reporting Services is now the primary reporting platform for Microsoft Dynamics AX 2012. The reporting tools for Microsoft Dynamics AX enable you to |

define reports for the Reporting Services platform.

Reporting Services includes a complete set of tools that help you manage and define reports.

Reporting Services tools work in the Microsoft Visual Studio environment and integrate with SQL Server tools and components.

| | |
|--|---|
| What's New: Security for Developers in Microsoft Dynamics AX 2012 | <p>The security keys of earlier versions are gone from Microsoft Dynamics AX 2012. The record level security (RLS) feature will be removed from a future version of the product.</p> <p>The new role-based security system makes security design and administration easier.</p> <p>Responsibilities are shared between the system administrator and the developer. The developer works with the new Security node in the AOT.</p> |
| What's New: Services & AIF for Developers in Microsoft Dynamics AX 2012 | <p>New features are available in Services and Application Integration Framework. Services enable integration with Microsoft Dynamics AX, and AIF is the services framework.</p> |
| What's New: Visual Studio Tools for Developers in Microsoft Dynamics AX 2012 | <p>New features in Visual Studio Tools enable integrated development with Visual Studio.</p> |
| What's New: Workflow for Developers in Microsoft Dynamics AX 2012 | <p>New and enhanced features for Workflow include a new Workflow Editor with advanced controls, line item workflows, and automated workflows.</p> |
| What's New: X++ for Developers in Microsoft Dynamics AX 2012 | <p>Microsoft.NET Framework is integrated with Microsoft Dynamics AX in the following ways:</p> <ul style="list-style-type: none">· X++ code can now be compiled into the common intermediate language (CIL) of Microsoft.NET Framework. This enables services and batch jobs to run faster.· .NET interop to X++ is now supported. Microsoft Visual Studio can now generate C# (or Visual Basic) code for proxy classes that wrap X++ classes and tables.· Events are now supported in X++. The AOT provides useful nodes that make implementation easier. Events can be used to reduce the need to change custom code when you upgrade to newer versions of Microsoft Dynamics AX. Event handler methods can be written in X++ or in any |

Microsoft.NET Framework language.

- Attribute classes can be used to add metadata to classes, tables, and methods.
 - The X++ language has the new keywords **as** and **is**. You can use the **is** keyword to test for inheritance relationships during run time. You can use the **as** keyword to verify the correctness of your downcast assignments during run time.
-

What's New: Analytics for Developers in Microsoft Dynamics AX 2012

This section describes the changes to the Online Analytical Processing (OLAP) framework in Microsoft Dynamics AX 2012.

Analytics

Microsoft Dynamics AX provides a model driven approach for developing Business Intelligence (BI) applications that use the SQL Server Business Intelligence Development Studio (BIDS) development environment. Microsoft SQL Server Analysis Services provides OLAP functionality.

What is New or Changed?

You can use Microsoft Dynamics AX views to model data in perspectives.

The Analysis Services project wizard is introduced to make it easier to create, update, configure, and deploy Analysis Services projects. The wizard performs many of the tasks that you would otherwise need to perform manually in BIDS.

You can create Gregorian calendars to use as date dimensions in your Analysis Services project.

Microsoft Dynamics AX View Support in Perspectives

| What can you do? | Microsoft Dynamics AX 2009 | Microsoft Dynamics AX 2012 | Why is this important? |
|-----------------------------|---|---|--|
| Model data in perspectives. | Use tables to model data in perspectives. Use custom query definitions to perform the following tasks: <ul style="list-style-type: none">Join tables with complex join conditions.Reduce the scope of tables.Create measures or dimension attributes using calculated fields. | Use tables and views to model data in perspectives. | Modeling data using views and tables makes it easier to customize cubes. |

Analysis Services Project Wizard

| What can you do? | Microsoft Dynamics AX 2009 | Microsoft Dynamics AX 2012 | Why is this important? |
|--|--|--|---|
| Deploy default cubes. | Initially deploy the default cubes by using the Setup wizard. | Use the Analysis Services project wizard to deploy an existing Analysis Services project. Choose the default cubes from the AOT. | It is easier to deploy the default cubes. |
| Configure an Analysis Services project after you change the Microsoft Dynamics AX configuration. | Manually update your Analysis Services project in BIDS to remove measures, dimensions, and KPIs that you do not have access to according to the license and configuration keys in Microsoft Dynamics AX. | Use the Analysis Services project wizard to remove measures, dimensions, and KPIs that you do not have access to according to the license and configuration keys in Microsoft Dynamics AX. | It is easier to update Analysis Services projects after you change the Microsoft Dynamics AX configuration. |
| Update an existing Analysis Services project. | Use the Update a Business Intelligence project form. | Use the Analysis Services project wizard to update an Analysis Services project to do the following: <ul style="list-style-type: none"> • Add or remove dimensions. • Add, remove, or update calendars to use as Date dimensions. • Add or remove label translations. • Add or remove currency conversion. | It is easier to update an existing Analysis Services project. |
| Create a new Analysis Services project. | Use the Update a Business Intelligence project form. | Use the Analysis Services project wizard to create an Analysis Services project. | It is easier to create a new Analysis Services project. |

Date Dimensions using Gregorian Calendars

Microsoft Dynamics AX

| What can you do? | Microsoft Dynamics AX 2009 | Microsoft Dynamics AX 2012 | Why is this important? |
|-------------------------------|---|--|--|
| Add date dimensions to cubes. | Date dimensions are not available in Microsoft Dynamics AX 2009. Use a system time dimension instead. | Define Gregorian calendars in the Calendar builder form in Microsoft Dynamics AX that you can use as Date dimensions in your Analysis Services project. | <p>The limitations of using the server time dimension in Microsoft Dynamics AX 2009 are as follows:</p> <ul style="list-style-type: none">• By default, only one Gregorian calendar can be included in the Analysis Services project.• Date attributes cannot be translated to Microsoft Dynamics AX languages.• You can only use attributes from SQL Server Analysis Services.• Null dates (1/1/1900) are not available. |

What's New: AOS and Database for Developers in Microsoft Dynamics AX 2012

This section describes the new and enhanced features that are planned to be available for Application Object Server (AOS) in Microsoft Dynamics AX 2012.

What is new or changed?

The following improvements were made in AOS:

- Valid Time State Tables
- UnitOfWork Class
- Inheritance Among Tables
- Full Text Index
- TempDB Temporary Tables
- Computed Columns in Views
- Filters in Outer Join
- Add a Having Clause to a Query in the AOT
- Effect of Disabling a Table

Valid Time State Tables

| What can you do? | Microsoft Dynamics AX 2009 | Microsoft Dynamics AX 2012 | Why is this important? |
|---|--------------------------------|--|--|
| Let the system manage data relationships that are valid only during a specified date range. | The feature was not supported. | You can set the ValidTimeStateFieldType property on a table to make the table a valid time state table. The system then adds the ValidFrom and ValidTo columns that track a date range in each row. The system guarantees that the values in these date fields remain valid by preventing overlap among date ranges, based on the primary key value. The X++ select statement has the validTimeState keyword to filter rows by date or date range. The Query class has properties that provide a filter by date range, such as the ValidTimeStateAsOfDateTimeRange property. | The program code that manages data relationships over date ranges is complex. Now, Microsoft Dynamics AX can manage these relationships for you. |

UnitOfWork Class

Microsoft Dynamics AX

| What can you do? | Microsoft Dynamics AX 2009 | Microsoft Dynamics AX 2012 | Why is this important? |
|--|--------------------------------|--|---|
| Manage the operation sequence and transaction integrity for row-based database operations. | The feature was not supported. | You can use the UnitOfWork class to safely manage the operation sequence and transaction integrity for your row-by-row database deletes, inserts, and updates. You first register each of your SQL data modification calls with the UnitOfWork object. Then, with one method call, you instruct the object to run the data modification calls. The object calculates the correct sequence in which each data modification call is run. | Application programmers no longer have to track every detail of foreign key relationships when they perform a set of data modification calls. The UnitOfWork class helps enforce transaction integrity. |

Inheritance Among Tables

| What can you do? | Microsoft Dynamics AX 2009 | Microsoft Dynamics AX 2012 | Why is this important? |
|--|--------------------------------|--|--|
| Let a table inherit fields, methods, and relationships from another table. | The feature was not supported. | Just as an X++ class could inherit from another X++ class in Microsoft Dynamics AX 2009, a table can inherit from another table in Microsoft Dynamics AX 2012. As in earlier releases, the Common table is still the base table of every table. A table that inherits from a base table is called a derived table. The terms parent table and child table describe foreign key relationships, not inheritance. | Table inheritance can improve your design of data models and reuse of schema components. |

Full Text Index

| What can you do? | Microsoft Dynamics AX 2009 | Microsoft Dynamics AX 2012 | Why is this important? |
|--------------------------------------|--------------------------------|---|---|
| Create a full text index on a table. | The feature was not supported. | A full text index has a separate entry for each significant word in a string field. The QueryRange class can enable an instance of the Query class to benefit from a full text index. | This feature supports full text queries. Full text queries improve the performance of SQL statements that have a where clause condition that is directed at a word that is embedded in the middle of a string field. |

TempDB Temporary Tables

| What can you do? | Microsoft Dynamics AX 2009 | Microsoft Dynamics AX 2012 | Why is this important? |
|---|---|--|--|
| Have a temporary table of the TempDB type. | Only temporary tables of the InMemory type were supported. | In the properties for a table in the Application Object Tree (AOT), the TableType property is now an enum property. The enum value InMemory is the legacy type of temporary table that is hosted in the client. The enum value TempDb is the new type of temporary table that is hosted in the TempDb database of the underlying Microsoft SQL Server. TempDB temporary tables can be joined with regular tables on the database tier. Joining TempDB temporary tables with regular tables often improves performance and simplifies the programming model. | TempDB temporary tables can be joined with regular tables on the database tier. Joining TempDB temporary tables with regular tables often improves performance and simplifies the programming model. |

Computed Columns in Views

Microsoft Dynamics AX

| What can you do? | Microsoft Dynamics AX 2009 | Microsoft Dynamics AX 2012 | Why is this important? |
|-----------------------------------|--------------------------------|---|---|
| Have a computed column in a view. | The feature was not supported. | A computed column is the output of a computation that uses a regular column as input. | Using a computed column can simplify application development. Various parts of the client code can all rely on the computed column that is processed by the server in one location. |

Filters in Outer Join

| What can you do? | Microsoft Dynamics AX 2009 | Microsoft Dynamics AX 2012 | Why is this important? |
|---|--------------------------------|---|---|
| Implement filtering of the result set from an outer join. | The feature was not supported. | You can use the new QueryFilter class to filter the result set from an outer join. The QueryFilter class can produce different results than the QueryBuildRange class in outer joins. The QueryFilter class filters later in the process for internal queries, and can prevent rows that have null values from appearing in the result set. | Filtering the result set from an outer join can reduce the number of rows that the database server must scan. |

Add a Having Clause to a Query in the AOT

| What can you do? | Microsoft Dynamics AX 2009 | Microsoft Dynamics AX 2012 | Why is this important? |
|--|--------------------------------|--|--|
| Use the AOT to add a having clause on the data source for a query. | The feature was not supported. | The AOT now has a Having node under Queries > MyQuery > Data Sources . This node corresponds to the having clause in standard SQL. | In a query, you can now specify a filter condition for an aggregate value. |

Effect of Disabling a Table

| What can you do? | Microsoft Dynamics AX 2009 | Microsoft Dynamics AX 2012 | Why is this important? |
|---|--|---|--|
| Retain the table in the underlying Microsoft SQL Server database when the table is disabled in Microsoft Dynamics AX. | The underlying table was dropped at the moment that it was disabled. | When you disable the configuration key for a table that is listed in the AOT, the corresponding table in the underlying database management system is not dropped. When you modify a configuration key to disable a table, you must decide whether you want to manually delete the data that is in the table. | External programs and Microsoft SQL Server cubes no longer fail when you disable a table. The programs and cubes can continue to read the data from the table, because they bypass Application Object Server (AOS). You must decide whether you want external processes to continue to read from the disabled table. |

What's New: Best Practices for Developers in Microsoft Dynamics AX 2012

The best practice checks that are enforced by the X++ compiler were updated to reflect new features added for this release. In addition, focus was put on security, and the existing rules that generated warnings were evaluated.

New Best Practice Checks

The following table describes the new best practice rules. The message that occurs for a best practice informational, warning, or error violation will occur when you check best practices.

| Message | Description |
|---|--|
| RecID field cannot be part of the ReplacementKey index. | <p>A RecID field was set as the replacement key.</p> <p>Best practice rule that supports relational application design. The following properties were added to define a model:</p> <ul style="list-style-type: none"> • The dbTableMapNode property was added to the Table node. • The dbReferenceNode property was added to the Relation node. • The dbIndexNode property was added to the Index node. <p>This best practice check requires unique names in the inheritance hierarchy. When you create a new field in a table, the best practice logic iterates through the inheritance hierarchy to ensure the RecID and ReplacementKey index are unique.</p> |
| Class is packing element IDs instead of element names. | <p>You used an element ID in your pack and unpack methods. Examples of element IDs are ClassId, TableId, and FieldId. Use the element name in your pack and unpack methods.</p> |
| The form must include a grid that contains ValidFrom and ValidTo fields when that form has a data source with ValidTimeStateAutoQuery set to DateRange. | <p>If a form has a data source with ValidTimeStateAutoQuery property set to DateRange, the form must include a grid that contains valid from and to fields.</p> |
| Table is using CreatedDateTime or ModifiedDateTime, RecId index needs to be created. | <p>When a table uses CreatedDateTime or ModifiedDateTime then it must have a RecID field created for the table.</p> |
| The class %1 is obsolete. Instead, use the class %2. | <p>You tried to use a class that is obsolete.</p> |
| Do not use braces around the case blocks in | <p>You have braces around a case block in a switch statement that must</p> |

| | |
|--|--|
| a switch statement. | be removed. |
| Do not use the <syntax> tag in XML documentation. The syntax is derived from the method declaration, and must not be explicitly provided. | You have a <syntax> tag in XML documentation that must be removed. |
| Line item workflow '%1' event handler does not reference a valid class implementing the '%2' interface | Best practice rule for event handling that is now supported for approvals and tasks. |
| Line item workflow relation '%1' not found on workflow document query '%2' | Best practice rule for event handling that is now supported for approvals and tasks. |
| Line item workflow relation '%1' is invalid. Relation must be configured with a fetch mode of 1:n on workflow document query '%2' | Best practice rule to support event handling that is now supported for approvals and tasks. |
| Line item workflow relation not defined | Best practice rule to support event handling that is now supported for approvals and tasks. |
| Line item workflow type '%1' not found | Best practice rule to support event handling that is now supported for approvals and tasks. |
| Workflow document query not found on line item workflow type '%1' | Best practice rule to support event handling that is now supported for approvals and tasks. |
| Line item workflow relation '%1' does not match root datasource on line item workflow type document query '%2' | Best practice rule to support event handling that is now supported for approvals and tasks. |
| Line item workflow must have at least one line item workflow type | Best practice rule to support event handling that is now supported for approvals and tasks. |
| The Company association type cannot be used for this workflow type because the table for the primary data source used by the workflow type is not configured to 'Save Data Per Company'. | Choose an alternative association type. This best practice rule was added to support organizational model types in workflow. |
| The Global and Other association types cannot be used for this workflow type because the table for the primary data source used by the workflow type is configured to 'Save Data Per Company'. | Best practice rule to support organizational model types in workflow. |
| An association type relation must be selected when the association type is Other | Best practice rule to support organizational model types in workflow. |
| A class extending the SysTableExtension | Best practice rule to support the extension of the base class |

Microsoft Dynamics AX

| | |
|--|---|
| class must not have any member variables. | SysTableExtension. Use the SysTableExtension class to override the table data defaults. |
| The node's path exceeds the maximum length of 400 characters. Reduce number of nesting levels or rename nodes. | Best practice rule to report tree node paths that exceed a string size of 400. |
| Another interface already defines %1. | Best practice rule added to check that there are no method conflicts between the interfaces. For example, a class cannot implement two interfaces that both define a method with the same name. |
| The class %1 does not implement methods %2 | Best practice rule that checks that the implementation in a class implementing an interface contains the methods specified in the interface profile. |
| The base type of the return type of method %1 is different from the base type of the corresponding field. | Best practice rule to check that the implementation in a class implementing an interface has the return type that is specified in the parameter profile. |
| The expected parameter count does not match the actual parameter count. | Best practice rule to check that the implementation in a class implementing an interface has the same number of parameters that is specified in the parameter profile. |
| The parameter %5 must be optional on both methods %1.%2 and %3.%4, or be non-optional on both. | Best practice rule that checks that the parameter is either optional on both methods or required on both methods. |
| The parameter %5 must be of the same type on both methods %1.%2 and %3.%4. | Best practice rule that checks that the parameter has the same type on both methods. |
| The upgrade script is missing required attribute %1. | Best practice rule that checks that the upgrade script has the required Description attribute. |
| The upgrade script is missing required attribute %1. | Best practice rule that checks that the upgrade script has the required Type attribute. |
| The upgrade script is missing required attribute %1. | Best practice rule that checks that the upgrade script has the required Stage attribute. |
| The upgrade script is missing required attribute %1. | Best practice rule that checks that the upgrade script has the required Table attribute. |
| The upgrade script method cannot be static. | Best practice rule that checks that the upgrade script is a non-static method. |
| The upgrade script method cannot accept parameters. | Best practice rule that checks that the upgrade script method has no parameters. |
| Table %1 specified in the UpgradeScriptTableAttribute is not valid. | Best practice rule that checks that the table stated in the attribute of an upgrade script exists in the AOT. |
| Configuration Key %1 specified in the UpgradeScriptConfigKeyAttribute is not | Best practice rule that checks that the configuration key stated in the |

| | |
|--|---|
| valid. | attribute of an upgrade script exists in the AOT. |
| Class %1 specified in the UpgradeDependsOnModuleAttribute is not valid. | Best practice rule that checks that the class name stated in the attribute of an upgrade script exists in the AOT. |
| Method %1 in class %2 specified in the UpgradeDependsOnModuleAttribute is not valid. | Best practice rule that checks that the method name stated in the attribute of an upgrade script exists in the AOT. |
| The upgrade script does not have a valid version defined [#define.version(sysReleasedVersion)] in the class declaration. | Best practice rule that checks that the script version stated in the attribute of an upgrade script exists in the AOT. |
| Return type covariance: Return values must both be either tables or maps (%1.%2). | Best practice rule that checks that the return values must both be either tables or maps. |
| @Return type covariance: Methods must return the same map (%1.%2). | Best practice rule that checks that the methods must return the same map. |
| The %1 relation to the %2 table does not have an upgrade script assigned. You must assign an upgrade script or exclude it from upgrade validation. | <p>The unit of measure table is now global. At upgrade, company-specific units will be merged using user-defined mapping. This requires that existing tables that have relations to the unit of measure also apply the mapping. This best practice checks for relations to unit of measure and confirms that the relation to unit of measure is registered with an upgrade script or that the table is marked to be excluded from the upgrade.</p> <p>The following code illustrates how to register a unit of measure upgrade job:</p> <pre>UnitOfMeasureUpgradeExecutor::registerUpgradeJobs()</pre> <p>The following code illustrates how to exclude a unit of measure table:</p> <pre>UnitOfMeasureUpgradeValidator::registerExcludedRelations()</pre> <p>For information on pre-upgrade tasks regarding units of measure see the following topics, Units, Unit conversions, Fixed units, and Unit texts.</p> |
| A non-surrogate key primary index is not valid for a company-specific table | Best practice rule that checks that the primary key for a company-specific table must be a surrogate key. |
| Use of labels for developer documentation | Best practice rule that checks that tables have a description of the table specified in the DeveloperDocumentation property. |
| Framework classes must have a disclaimer in the ClassDeclaration to discourage customizations. | Best practice rule that checks that the framework classes providing platform functionality have a message on them to discourage developers from customizing them. Customizing and extending classes that provide platform functionality may cause problems with future upgrades. |

Microsoft Dynamics AX

| | |
|---|--|
| The framework class disclaimer must be placed after the XML documentation. | Framework classes have a disclaimer to discourage developers from customizing them. Customizing and extending classes providing platform functionality may cause problems with future upgrades. Best practice rule that checks that the disclaimer is placed after the XML documentation, meaning after the <code>///</code> comments. |
| Each perspective entity (table or view) must have at most 1 parent. | Best practice rule that checks that the perspective entity (table or view) has at most one parent. |
| Each perspective entity (table or view) must have at most 1 child. | Best practice rule that checks that the perspective entity (table or view) has at most one child. |
| Each perspective entity (table or view) must have a corresponding child for each parent. | Best practice rule that checks that the perspective entity (table or view) has a corresponding child for each parent. |
| Each perspective entity (table or view) must have a corresponding parent for each child. | Best practice rule that checks that the perspective entity (table or view) has a corresponding parent for each child. |
| In each perspective entity the parent type should be the same as the child type if present. | Best practice rule that checks that in a perspective entity, the parent type is the same as the child type if one exists. |
| Perspectives where SharedDimensionContainer=Yes cannot have fields where AnalysisUsage is Measure or Both. Perspective %1, table or view %2, field %3. | When a perspective has the SharedDimensionContainer property set to True , the AnalysisUsage property for fields should not be marked as Measure or Both. Setting the SharedDimensionContainer property to True indicates that the perspective will be used to create dimensions only. The perspective will not be used to create a cube, measure group, or measure. |
| %1 or %2 must not be used in the sys layer. | Best practice rule that checks that the UpgradeModuleDependsOnMeAttribute attribute and UpgradeTaskDependsOnMeAttribute attribute are not used in the SYS layer. |
| Inconsistent usage of AOSAuthorization property for a table in chain of inheritance. Ensure that both tables %1 and %2 have authorization either enabled or disabled. | Best practice rule that checks that all the tables in the inheritance hierarchy have the same value for authorization: either enabled or disabled. |
| Allowed CacheLookUp values for this TableGroup is %1 | Best practice rule that checks that the cache lookup is valid for the table group. |
| Service group must have a valid description | Best practice rule that checks that the Description property for a service group is provided. The description value must be a valid label. |
| Service must have a valid Namespace property | Best practice rule that checks that the Namespace property for a service is set to a valid namespace. You may not use the default URI, for example <code>http://tempuri.org</code> . |

| | |
|---|---|
| Entry point is not in any privilege | Best practice rule that checks that the entry point is in a privilege. |
| Menu item points to a class that is not a RunBase class; it should probably have a linked permission object | Best practice rule that checks that the menu item has a linked permission. |
| All duties should be part of a role | Best practice rule that checks that a duty is part of a role. |
| All duties should be part of a process cycle | Best practice security rule that checks that a duty is part of a process cycle. |
| A form that is not associated with other forms should be linked to at least one menu item | Best practice security rule that checks that a form is linked to a menu item. |
| Non-system-managed %1 resource with incorrect managed by value %2 | Best practice security check that a non-system-managed resource has Managed By value set to Manual . |
| Privilege is not in any duty. | Best practice security rule that checks that the privilege is set to a valid duty. |
| Missing SysEntryPointAttribute on service operation | Best practice rule that checks that the SysEntryPointAttribute attribute on the service operation because it is exposed externally. |
| Missing processReport method | Best practice rule that checks that the report has a processReport method. |
| Missing SysEntryPointAttribute on SSRS RDP class | Best practice rule that checks that the SQL Server Reporting Services report data provider class has a processReport method defined with a SysEntryPointAttributeSysEntryPointAttribute attribute specified. |
| Info part not linked to menu items | Best practice security rule that checks that an Info Part is linked to a menu item. An Info Part is a collection of related data fields you can associate with a record in a form. You use Info Parts to create a FactBox for a form or the preview pane for a list page. To add an Info Part to a form, you use a menu item that links to the Info Part. |
| DeveloperDocumentation label should not be marked to be translated. | Best practice rule that checks that the DeveloperDocumentation property should not be translated. The DeveloperDocumentation property should have the {Locked} keyword in the comment for the label used in the DeveloperDocumentation property. |
| The help provider class isn't specified for this online documentation set | The HelpProviderClass property must be set if the ContentLocation property is set to Online for a Help Document Set. |
| Query %1 can not be made searchable because it has zero or more that one data sources. | Best practice rule that checks that the searchable query has a data source. You will get a best practice violation if no data source is specified or if more than one data source is specified. |
| Query %1 cannot be made searchable because it does not have the searchlinkrefname property or formref | Best practice rule that checks that the searchable query must have the SearchLinkRefName property or the FormRef property set on the root table. |

Microsoft Dynamics AX

property set on the root table %2.

Query %1 is searchable but it does not have the searchlinkrefname property set to a url menu item on the root table %2. The search results in will not appear in Enterprise Portal.

Best practice rule that checks that the searchable query must have the **SearchLinkRefName** property set to a URL menu item on the root table or the search results will not appear in Enterprise Portal.

Query %1 is searchable but it does not have the formref property set to a form on the root table %2. The search results in will not appear in AX client.

Best practice rule that checks that the searchable query must have the **FormRef** property set to a form on the root table or the search results will not appear in the Microsoft Dynamics AX client.

Query %1 can not be made searchable because it does not have the titlefield1 or titlefield2 set on the root table %2.

Best practice rule that checks that the valid searchable query must have the TitleField1 or the TitleField2 fields set on the root table.

Update permissions for search crawler role from searchable queries.

Best practice rule that checks that the search crawler role has permissions for the table that was used in the query.

Query %1 can not be made searchable because it has has a root table that is a transaction table with no index on the modifieddatetime field.

Best practice rule that checks that the searchable query must have a root table that is a transaction table with an index on the ModifiedDateTime field.

Element has changed name since previous release. If the rename is intentional then suppress this violation; otherwise, update the LegacyID. Old name was: %1

Best practice rule that checks that the element has the same name as the previous release. If the change was intentional then suppress the violation. This best practice check is not new, the message has been updated. This check has been modified due to the changes regarding installation specific IDs.

The element has an origin value that is different from the origin value in a previous version of Microsoft Dynamics AX.

Best practice rule that checks that the same origin value occurs in the earlier version of Microsoft Dynamics AX. This check was added due to the changes regarding installation specific IDs.

Element has no LegacyID assigned and was shipped in a previous version. All ID-based elements already shipped must have a LegacyID. Set Legacy ID to %1.

Best practice rule that checks that the element has a **LegacyID** property assigned because the element was shipped in an earlier version of Microsoft Dynamics AX. This check was added due to the changes regarding installation specific IDs.

Element has a different LegacyID than previous shipped version. Please update LegacyID, old LegacyId was %1.

Best practice rule that checks that the value specified in the **LegacyID** property occurs in the earlier version of Microsoft Dynamics AX. If a violation of this best practice occurs, update the **LegacyID** property. This check was added due to the changes regarding installation specific IDs.

The element does not have an origin value. This element must have a non-null origin value.

Best practice rule that checks that the element has an origin value. This element must have an origin value that is not **NULL**. This check was added due to the changes regarding installation specific IDs.

| | |
|--|--|
| Do not use a deprecated field as title field. Please select another title field. | Best practice rule that checks that the deprecated field is not used as a title field. |
| Do not use deprecated field %1 in non-deprecated index %2. Please remove the field or deprecate the index. | Best practice rule that checks that the deprecated field is not used as an index. Remove the deprecated field or deprecate the index. |
| Do not use deprecated field %1 in field group %2. Please remove the field from the field group. | Best practice rule that checks that the deprecated field is not used in a field group. Remove the deprecated field from the field group. |

Best Practice Checks Updated for Container Controls

The best practice checks that were updated for client forms verify the name and position for container controls. The following table describes the changes that were made.

| Message | Description |
|--|---|
| Control %1 bound to table field group has name that does not match default name of %2, consequently the control requires storage and cannot be optimized | Best practice rule that checks for when a Group or Grid control has the DataGroup property set but the child control cannot be created when the form is loaded. This is because the name bound to the table field group does not match the default name. The control must be saved as metadata. This requires storage and the control cannot be optimized. |
| The form group and table group have different fields at position %1: '%2' != '%3', consequently the control requires storage and cannot be optimized | Best practice rule that checks that the Group or Grid control has the DataGroup property set but the child control cannot be created when the form is loaded. This is because the controls in the form group must match the position in the table field group. If the position does not match, the position must be saved as metadata. This requires storage and the position cannot be optimized. |

The best practice rule changes for container controls resulted in best practices that have been removed. The following table lists the best practice checks that were removed.

| Message | BPErrror code |
|---|---|
| Group control bound to Table group has %1 auto-declaration(s), consequently it cannot be AOS optimized | BPErrrorFormCannotBeAOSOptimizedAD |
| Group control bound to Table group has some other properties changed from default, consequently it cannot | BPErrrorFormCannotBeAOSOptimizedPropChg |

be AOS optimized

| | |
|---|--|
| Form group (%1) and table group (%2) have different number of fields, consequently they cannot be AOS optimized | BPErrormFormGroupControlDifNumOfFields |
|---|--|

Best Practice Checks Updated for Workflow Approval on the Web

When you create workflows for use only on the Web, only thin client menu items are required. Best practice rules were updated to account for this.

The following table describes the messages that were updated so that they now apply to Web menu items in addition to action menu items.

| Label text | BPErrorm code and label |
|--|---|
| The action menu item is not defined. The action menu item or the action Web menu item must be defined. | BPErrormWorkflowNoActionMenuItem BPErrormWorkflowNoWebActionMenuItem @SYS108556 |
| The document menu item is not defined. The document menu item or the document Web menu item must be defined. | BPErrormWorkflowNoDisplayMenuItem BPErrormWorkflowNoWebURLMenuItem @SYS108559 |
| The element outcome '%1' action menu item property is not defined. The action menu item or the action Web menu item must be defined. | BPErrormWorkflowElementOutcomeNoActionMI BPErrormWorkflowElementOutcomeNoActionWMI @SYS108547 |

Best Practice Checks Changed from Warnings to Errors

Many of the existing rules that generate warnings were evaluated for the value they provide. When the best practice warning violations should be addressed, the warnings were changed to errors.

The following table provides information about best practice rules that were changed from warning messages to error messages. Violations to these rules must now be fixed.

| Message | BPErrorm code and label |
|--|--|
| Display and edit methods must use a base enum or an extended data type as their return type. | BPErrormTableNoExtendedReturnType @SYS72489 |
| The table %1 has no fields in the AutoReport field group. | BPErrormTableMissingGroupAutoReport @SYS55415 |
| The table %1 has no record ID index but does use the | BPErrormTableNoRecordIdIndex |

| | |
|---|---|
| record ID field explicitly in relation in %2. | @SYS60524 |
| Tables with only one index must have it defined as a cluster index. Set the ClusterIndex property for the %1 table. | BPErrTableOneIndexNotCluster @SYS68395 |
| The table %1 has no record ID index but uses it %2 times. | BPErrTableNoRecordIdIndexButUsed @SYS60522 |
| A display or edit method and a field cannot have the same name. Rename either the %1 field or the display or edit method. | BPErrTableFieldHasSameNameAsMethod @SYS97063 |
| The method %1%2%3 is obsolete. Instead, use the method %4. | BPErrMethodDiscontinuedInLaterVers @SYS68910 |
| The method %1 is empty and must be deleted. | BPErrMethodIsEmpty @SYS68904 |
| Do not put parentheses around the case values in a switch statement. Remove the parentheses from %1. | BPErrMethodParenthesisRoundCaseConst @SYS55397 |
| Only use braces around code blocks. | BPErrMethodNonNeededBlockStyleUsed @SYS152538 |
| Use double-quotation marks when referring to labels, instead of using single-quotation marks. | BPErrMethodLabelInSingleQuotes @SYS55395 |
| %1 and other variable names should not start with an underscore. Only parameter names are required to start with an underscore. | BPErrMethodVariableWithUnderscore @SYS60113 |
| Use a field list in the select statement of %1, because only %2% of the record data is referenced. | BPErrSelectUsingFieldList @SYS91289 |
| All tables must have a primary index. Set the PrimaryIndex property for the table %1. | BPErrTableNoPrimaryIndex @SYS107156 |

Removed Best Practice Checks

Many of the existing rules that generate warnings were evaluated for the value they provide. When the best practice warning rules held little value and were being ignored, the best practice warning rules were removed. Removing these warning rules helps to focus on the best practice warnings that should be considered.

The following table provides information about best practice rules that were removed because they are obsolete or were evaluated to be of low value.

Microsoft Dynamics AX

| Message | BPErr code |
|---|---|
| Enum is not referenced in X++ code, in the table field or in an Extended Type. | BPErrEnumNotUsed |
| Type is not referenced in X++ code, table field, or as a parent. | BPErrTypeNotUsed |
| Table group is %1. | BPErrTableNoTableGroup |
| Table has record ID index but does not use record ID field explicitly. | BPErrTableRecordIdIndexNotUseField |
| Table has a record ID index but does not seem to use the record ID field explicitly for lookup. | BPErrTableRecIdFieldUsedUseless |
| Field is not referenced in X++ code. | BPErrTableFieldNotUsed |
| Method is not referenced in X++ code or indirectly. | BPErrMethodNotUsed |
| Method contains constant numeric value: %1. | BPErrMethodConstantNumericArgUsed |
| Illegal name %1 %2: %3. Use parent, child or sibling. | BPErrMethodIllegalName |
| Consider alternative to single quoted text %1 appearing in %2. | BPErrMethodSingleQuotedText |
| Variable %1 is not written, but read. | BPErrMethodVariableReadButNotWritten |
| Method syntax must be autogenerated using the [HEADER] keyword. | BPErrDocNodeHardCodedMethodSyntax |
| No documentation for '%1'. | BPErrDocNodeNoDocumentation |
| No unique index set up for the table. | BPErrTableNoUniqueIndex |
| A primary index should be defined if there are a unique index. | BPErrTableUniqueIndexNoPrimary |
| Table Field of type InventDimId must be added to the Multisite Activation | BPErrTableFieldInventDimIdNotMultiSiteActivated |
| Security Key = 'NotDecided' (or others!) | BPErrSecurityKeySpecific |
| Security Key is unknown (May have been deleted!) | BPErrSecurityKeyUnknown |
| Security Key is not defined | BPErrSecurityKeyNotDefined |
| Configuration Key is unknown (May have been deleted!) | BPErrSecurityParentKeySpecific |
| Configuration Key is unknown (May have been deleted!) | BPErrSecurityParentKeyUnknown |
| Not Connected to a Security Key | BPErrSecurityKeyNotConnected |
| Security key should not be specified on container controls, as it prevents personalization. | BPErrSecurityKeyNotAllowed |

| | |
|--|---|
| SecurityKey AnalysisVisibility property should not be Auto if there is no parent security key | BPErrorSecurityKeyAnalysisVisibilityAuto |
| No self relation set up for the Table. | BPErrorTableNoSelfRelation |
| TwC: Validate access to return value from display/edit method. | BPErrorTwCDisplayEditMethod |
| Do not use TypeId() | BPErrorDoNotUseTypeId |
| Object has changed ID since previous release. Old ID = %1. | BPErrorObjectIdConflict |
| Set the property AutoDeclaration to Yes for the form control, instead of manually declaring the form control %1. | BPErrorMethodNotAutoDeclared @SYS68393 |

What's New: Client for Developers in Microsoft Dynamics AX 2012

The client is the Windows application that you use to view, update, and create data in Microsoft Dynamics AX 2012. This section describes the changes and new features that affect how you use forms, controls, and data sources to customize the client.

What is new or changed?

To customize the client in Microsoft Dynamics AX 2012, you can use the changes and new features in the following areas:

- Forms
- Controls
- Data Sources
- Parts
- Search

Forms

| What can you do? | Microsoft Dynamics AX 2009 | Microsoft Dynamics AX 2012 | Why is this important? |
|-----------------------------------|----------------------------|---|--|
| Set the Style property of a form. | Feature not available. | Specify the type of form that you want to create. | After you set the Style property, several form properties are updated to use the values required for the specified form. |

Microsoft Dynamics AX

| | | | |
|---|--|---|---|
| Create a Table of Contents form. | Feature not available. | Use the table of contents pattern for small forms that display data on one or more tabs. | The Table of Contents form is a useful design pattern that you can use to create a form that sets parameter values. For example, the Accounts receivable parameter form uses the table of contents pattern. |
| Use metadata to create a list page that appears in both the Microsoft Dynamics AX client and Enterprise Portal for Microsoft Dynamics AX. | If you want the same list to appear in the client and Enterprise Portal, you had to create two separate list pages. | The list page that you create can appear in both the client and Enterprise Portal. | A single list page saves development time, and ensures that the same list is viewed in both the client and Enterprise Portal. |
| Add an action pane to a form. | Action panes were limited to list pages and content pages. | You can add an action pane to a details form. | Action panes improve access to actions that were previously available on the menu or toolbar of the form. |
| Specify that a form opens in read-only document view mode. | When you view a form that has editable controls, you could accidentally change a field value. | To quickly view data values, you use forms that are in read-only mode. | To reduce the risk of accidental changes, you can specify that the form show a read-only version of the form. |
| Add enhanced previews for foreign key fields or controls. | When you rest the pointer on a record in a list, a tooltip appears that shows one or two preview fields for that record. | When you rest the pointer on a record, the preview shows the data fields that are specified by a FactBox. | The use of a FactBox gives you the option to include more data fields in the preview. |
| Specify how a form opens. | You use a form property to specify the mode for a form. | You use the OpenMode property of menu items and menu item buttons to specify how the form opens in the client. | You specify whether the form opens to create a new record, update an existing record, or view an existing record in read-only mode. |
| Add FactBoxes to a form. | To supply additional information about the selected record in a list, you add a preview pane to the bottom of the list | You use FactBoxes and a preview pane to provide a variety of additional information that is related to the selected record in a form. | You can add several Factboxes that show various types of information. A FactBox uses a part object that defines the data that |

page.

appears in the FactBox. You can create new parts, and use them as FactBoxes or a preview pane.

| | | | |
|--|--|---|--|
| Specify whether the query from the parent form is copied to a child form. | Feature not available. | You can use the CopyCallerQuery property of menu items and menu item buttons to specify that the query of the parent form is used in the child form. | By copying the query from the parent form to a child form, you ensure that the forms display a consistent set of data. |
| Changes to form controls and methods are stored to your customization layer. | Customizations were stored at the main type level or the Concept level. Forms and tables are examples of main types. | Customizations to controls and methods are stored at the child level or the Element level. An individual control in a form is an example of an Element. | The additional granularity reduces the risk of conflicts with other customizations. |

Controls

| What can you do? | Microsoft Dynamics AX 2009 | Microsoft Dynamics AX 2012 | Why is this important? |
|-------------------------------------|---|---|--|
| Use a DropDialogButton control. | Feature not available. | You click a button to open a dialog box where you can complete an action. | You can complete actions without opening another form. |
| Add an action pane strip to a form. | Feature not available. | You can add an action pane strip that provides access to the actions that are associated with the form. | You use an action pane strip when the form does not have enough actions for a full action pane. In addition, an action pane strip enables you to put actions close to the records and data fields that they relate to. |
| Add FastTabs to a form. | Tabs were arranged horizontally across the top of the form. | A FastTab is a new vertical presentation style that replaces the existing tabs for forms. | FastTabs let you display large amounts of information in a single form. |
| Use the ReferenceGroup control. | Feature not available. | You can replace a surrogate foreign key with one or more | You use the ReferenceGroup control in |

Microsoft Dynamics AX

| | | | |
|--|---|--|---|
| | | data fields. | forms and lookup forms that contain a surrogate key. The ReferenceGroup control provides data values that are understandable to the user. |
| Use the segmented entry control. | Feature not available. | You use the segmented entry control to enter an account number and associated dimensions. | The task of entering complex combinations of accounts and dimensions is simplified. |
| Use the managed host control to add a .NET control to a form. | Feature not available. | You can add and use .NET controls in forms. | You can add and use Windows Presentation Foundation (WPF) or managed form controls. |
| Use the MSChart control. | To display data in a chart, you had to use the ChartFX ActiveX control. | To display data in a chart, you use the MSChart .NET control. | MSChart replaces ChartFX, and provides new capabilities to create and save a chart. |
| Disable or hide controls that are not used in a specified country or region. | Feature not available. | The form shows unique controls and navigation elements to users in a specified country or region. In addition, users from other countries or regions cannot access these controls and navigation elements. | You can target controls and navigation elements to users in specific countries or regions. The targeted controls and navigation elements are disabled or hidden when the form is viewed by users from other countries or regions. |
| Set the Style property of a control. | Feature not available. | Specify the type of control that you want to create. | After you set the Style property, several control properties are updated so that they use the values that are required for the specified control. |

Data Sources

| What can you do? | Microsoft Dynamics AX 2009 | Microsoft Dynamics AX 2012 | Why is this important? |
|-------------------------|----------------------------|----------------------------|---------------------------|
| Use date effective data | There was no standard | You create forms that use | Forms display the current |

| | | | |
|--|---|---|--|
| from a table. | framework for displaying date effective data. | from dates and to dates for data records. | data for fields that support date effective data. You can reference a table that contains date effective dates, and specify query and update behavior. |
| Use a modeled query with a menu item. | Feature not available. | You use a menu item to specify the query that is used for a Cue or a secondary list page. | The development and maintenance of Cues and secondary list pages is simplified. |
| Replace a surrogate foreign key with specified data fields. | Tables and lookup forms used replacement keys to identify a related record. | Tables and lookup forms use a surrogate key to identify a related record. | A form or lookup form that uses a surrogate key shows values that are understandable to the user, instead of the value of the surrogate foreign key. |
| Defer the loading of replacement values for surrogate foreign keys. | Feature not available. | The replacement fields are not loaded until they are visible. | Performance is improved, because joined data is not retrieved until it is needed. |
| Use a derived data source to access data from tables in an inheritance hierarchy for tables. | Feature not available. | Controls can be bound to fields from the derived tables in an inheritance hierarchy. | The derived data source lets you use data from a table in an inheritance hierarchy. |
| Specify updates to several tables as a single unit of work. | When an update occurred, each table was updated in a separate trip to Application Object Server (AOS). In addition, each trip was a separate transaction. | When you save a record that changes multiple tables, the changes are combined into a single transaction that is initiated by AOS. | The single transaction provides a more efficient and consistent framework for completing data changes. |
| Specify how optional records in outer joined tables are created or deleted. | Optional records were always created and saved. | The optional record is only created if you enter data for an outer joined table. | You can specify the behavior of optional records. You can choose to always create, implicitly create, or explicitly create the optional record. |
| Filter on the fields in all joined tables. | Filtering on outer joined fields was disabled. | The Filter-by-Grid , Filter-by-Field , Filter-by-Value , Filter-by-Selection , and Advanced Filter dialog boxes support filtering on | You can sort and filter on the fields in all of the joined tables. |

outer joined fields.

Parts

| What can you do? | Microsoft Dynamics AX 2009 | Microsoft Dynamics AX 2012 | Why is this important? |
|--|---|--|---|
| Create or update a FactBox. | Feature not available. | You can create or update a FactBox that appears on a list page or in a form. | Info parts enable you to specify the data that appears in a FactBox. |
| Use an existing form as a FactBox. | Feature not available. | You can use a form as a FactBox that provides additional information about the current record. | You have more options to customize the types of controls that appear in the FactBox. |
| Create a Cue in the Application Object Tree (AOT) that uses an existing query. | Cues used copies of the list page query and could break if the list page query was later changed. | You use the AOT to create a Cue that uses an existing query. | Changes to the list page query no longer affect the Cue. |
| Create a Cue group that you can use as a FactBox. | Feature not available. | A Cue group includes references to one or more Cues. | You use a Cue group as a FactBox that provides summary information that is related to the current record. |

Search

| What can you do? | Microsoft Dynamics AX 2009 | Microsoft Dynamics AX 2012 | Why is this important? |
|---|----------------------------|---|---|
| Make all forms and reports available from Search. | Feature not available. | To make a form or report appear in search results, you create a menu item, and add the search terms that you want to associate with the specified form or report. | You can use Search to find the forms and reports that are associated with a task. |

What's New: Enterprise Portal for Developers in Microsoft Dynamics AX 2012

This section describes the new and enhanced features of Enterprise Portal for Microsoft Dynamics AX 2012. Enterprise Portal provides a web-based application framework that allows for users to interact with data in Microsoft Dynamics AX through a web browser.

What is new or changed?

The following improvements have been made for this release of Enterprise Portal:

- Microsoft SharePoint 2010
- Microsoft Visual Studio 2010
- Support for Microsoft Dynamics AX 2012 Features
- User Interface and Usability Improvements
- List Page Framework
- Services Framework
- Search
- Pluggable Authentication
- Proxy Support
- Cross-reference Support

Microsoft SharePoint 2010

| What can you do? | Microsoft Dynamics AX 2009 | Microsoft Dynamics AX 2012 | Why is this important? |
|---|--|--|---|
| Microsoft SharePoint 2010 provides core features that are used by Enterprise Portal, such as document libraries that store Enterprise Portal pages. | Enterprise Portal uses Windows SharePoint Services (WSS) 3.0 or Microsoft Office SharePoint Server 2007. | Enterprise Portal uses SharePoint Foundation 2010 or SharePoint Server 2010. | SharePoint 2010 provides features, such as the SharePoint Ribbon, that improve the functionality and user experience for Enterprise Portal. |

Microsoft Visual Studio 2010

Microsoft Dynamics AX

| What can you do? | Microsoft Dynamics AX 2009 | Microsoft Dynamics AX 2012 | Why is this important? |
|---|---|---|---|
| You use Visual Studio 2010 to develop Enterprise Portal integrations. | Enterprise Portal integrations are developed with Visual Studio 2008. | Enterprise Portal integrations are developed with Visual Studio 2010. | Visual Studio 2010 is a standard development tool for Microsoft Dynamics AX 2012. |

Support for Microsoft Dynamics AX 2012 Features

| What can you do? | Microsoft Dynamics AX 2009 | Microsoft Dynamics AX 2012 | Why is this important? |
|---|--|--|---|
| Enterprise Portal supports tables that are structured into a table hierarchy. | Feature not available. | In Enterprise Portal, you can use tables that are structured in a hierarchy. | Allows for a more efficient database structure. |
| Enterprise Portal supports tables that use surrogate keys. | Tables typically displayed the key values. | Tables can have surrogate keys that are not displayed to the user. | Provides a user interface that is more user-friendly. |
| Enterprise Portal uses the new security model. | The security model was less flexible. | The security model is much more flexible and configurable. | System administrators have better control over security settings. |

User Interface and Usability Improvements

| What can you do? | Microsoft Dynamics AX 2009 | Microsoft Dynamics AX 2012 | Why is this important? |
|--|---|--|--|
| Pages can be opened as modal windows. | Pages always opened in-line, replacing the previous page that was displayed. | Pages can open as modal windows. This allows the previous page to remain. | Context is preserved when the user performs actions in Enterprise Portal. |
| Enterprise Portal has new User Control components. | The User Control components for Enterprise Portal pages limited the layout possibilities. | New user control components, such as AXColumn and AxMultiColumn, allow for more flexible layout. | Improved visual appearance of pages in Enterprise Portal improves user productivity. |
| Enterprise Portal can use optimized lookups. | Lookups load data at the time that they are | Lookups can be configured to load data at the time the page | The user experience is improved, because users |

displayed.

loads.

do not have to wait for the lookup to load.

List Page Framework

| What can you do? | Microsoft Dynamics AX 2009 | Microsoft Dynamics AX 2012 | Why is this important? |
|---|--|---|---|
| Create list pages for use in Enterprise Portal. | Separate list pages are created for Enterprise Portal. | A list page can be created and used in both the Microsoft Dynamics AX client and Enterprise Portal. | Saves development time when you are creating lists. |

Services Framework

| What can you do? | Microsoft Dynamics AX 2009 | Microsoft Dynamics AX 2012 | Why is this important? |
|--|---|--|---|
| Enterprise Portal uses Microsoft Dynamics AX services to access data and metadata. | Enterprise Portal used the .NET Business Connector to access data and metadata. | Enterprise Portal uses the Microsoft Dynamics AX services to access data and metadata. | Allows for more flexible deployment of Enterprise Portal. |

Search

| What can you do? | Microsoft Dynamics AX 2009 | Microsoft Dynamics AX 2012 | Why is this important? |
|---|---|---|--|
| Search for data in Microsoft Dynamics AX. | Data Crawler is used to implement search. | The SharePoint search infrastructure is used. | By using the search infrastructure of SharePoint, you can choose the appropriate |

search tool for your organization, such as Search Server Express or SharePoint Search Server 2010.

Pluggable Authentication

| What can you do? | Microsoft Dynamics AX 2009 | Microsoft Dynamics AX 2012 | Why is this important? |
|--|--|--|---|
| Enterprise Portal can use Windows Live ID to provide authentication for external-facing pages. | You must use Active Directory accounts for external users. | You can use Windows Live ID accounts for external users. | You do not have to have external users added to the Active Directory for your organization. |

Proxy Support

| What can you do? | Microsoft Dynamics AX 2009 | Microsoft Dynamics AX 2012 | Why is this important? |
|--|--|--|--|
| Code for Enterprise Portal can use the new proxy implementation. | Proxies are limited to Enterprise Portal and not easy to set up. | Proxies can be used throughout Microsoft Dynamics AX. This includes Enterprise Portal. | Proxies are easier to set up and use. They enable Enterprise Portal code to access Microsoft Dynamics AX resources, such as X++ classes. |

Cross-reference Support

| What can you do? | Microsoft Dynamics AX 2009 | Microsoft Dynamics AX 2012 | Why is this important? |
|---|--|--|--|
| You can use the cross-reference tool in the MorphX IDE to see how resources for Enterprise Portal are used. | The cross-reference tool does not support Enterprise Portal resources. | The cross-reference tool supports Enterprise Portal resources. | It is now easier to see how resources are used in Enterprise Portal. |

Are there any special considerations?

If you have created User Controls for Enterprise Portal pages in Microsoft Dynamics AX 2009, you will need to update them for use in Enterprise Portal for Microsoft Dynamics AX 2012.

What's New: Help System for Developers in Microsoft Dynamics AX 2012

Microsoft Dynamics AX 2012 provides a new approach to supplying Help documentation. The new Help system uses a Help server to store and manage documentation. The Help server is an IIS virtual application that runs inside your network. To find and view Help documentation, you use the Microsoft Dynamics AX Windows client to open the Help viewer. The Help viewer displays individual Help topics. In addition, the Help viewer enables you to search the documentation hosted by the Help server.

What is new or changed?

To add and update Help documentation, the Help system enables you to complete the following types of tasks:

- Create Custom Documentation
- Add Documentation to the Help Server
- Use Search
- Support for Multiple Publishers

Create Custom Documentation

| What can you do? | Microsoft Dynamics AX 2009 | Microsoft Dynamics AX 2012 | Why is this important? |
|--|---|---|---|
| Create Help documentation. | Changes to Help documentation were possible but cumbersome, often requiring recompiling and replacing .chm files. | Allows users, system administrators, and developers to create documentation. | Enables you to customize the documentation to the needs of your organization. |
| Use Microsoft Office Word to create documentation. | Feature not available. | You can use Word to create documentation. | Anyone who has Word can create Help documentation. |
| Use HTML or text editors to create documentation. | Feature not available. | You can use HTML to create documentation. | You can use any HTML or text editor to produce your documentation. |
| Use Microsoft Dynamics AX labels in documentation. | Including labels was possible but required the recompilation of .chm files. | You can insert labels from the client user interface into your HTML Help documentation. | Your documentation matches what the user sees on the client. |
| Hide topics from other publishers. | You could remove topics from the .chm file but you then had to | When you add new Help documentation you can also hide outdated documentation | The Help viewer always shows the most relevant |

recompile and redistribute that file.

that was supplied by other publishers.

Help documentation.

Add Documentation to the Help Server

| What can you do? | Microsoft Dynamics AX 2009 | Microsoft Dynamics AX 2012 | Why is this important? |
|--|--|--|---|
| Publish new or updated documentation. | Changes to Help topics required recompilation of .chm files. | You add or update documentation by putting files on the Help server. | You can add or update documentation to reflect the needs of your organization. You can also add ongoing updates from Microsoft and other publishers. |
| Distribute documentation updates to users. | You had to install an updated .chm file on each client computer. | Documentation updates are available to all client computers when the update is published on the Help server. | You can make sure all clients have access to the same documentation. |

Use Search

| What can you do? | Microsoft Dynamics AX 2009 | Microsoft Dynamics AX 2012 | Why is this important? |
|--|---|--|---|
| Add keywords that improve topic discoverability. | Changes were possible but required the recompilation of .chm files. | You can add keywords that improve the ranking of your topic in a list of search results. | Search results are more accurate and informative. |
| View rich search results. | Search results from the .chm files were a list of topic titles. | Search results contain topic titles, topic summaries, and links to additional sources of Help documentation. | You can use the summaries to find the documentation that your need. |

Support for Multiple Publishers

| What can you do? | Microsoft Dynamics AX 2009 | Microsoft Dynamics AX 2012 | Why is this important? |
|--|---|---|---|
| View Help documentation produced by more than one publisher. | Adding documentation to the .chm files was difficult so Help documentation was often located in separate files. | The Help viewer retrieves and displays documentation from all publishers. | The Help viewer includes documentation from Microsoft, partners, developers, and other publishers. For example, you can write an update to a procedure and have your update appear next to the help documentation provided by Microsoft. |
| Add table of contents entries. | Changes were possible but required the recompilation of .chm files. | Each publisher can add table of contents entries for new documentation | Your table of contents entries are added to the main table of contents that appear in the Help viewer. |

Are there any special considerations?

To enable the customization of Help documentation, you have to add a folder to the Help server where writers put documentation files. A writer that publishes documentation files has to have write or modify security permissions to the specified folder.

What's New: MorphX Development for Developers in Microsoft Dynamics AX 2012

This section lists the new and enhanced features in the MorphX development environment that are planned to be made available in Microsoft Dynamics AX 2012.

What is new or changed?

The following MorphX development features have been added or changed in Microsoft Dynamics AX 2012:

- Models and the Model Store
- Object IDs
- AxUtil Command-Line Utility and PowerShell Cmdlets
- Development Workspace
- Some Layers Have Been Renamed
- Installation-Specific IDs

Models and the Model Store

| What can you do? | Microsoft Dynamics AX 2009 | Microsoft Dynamics AX 2012 | Why is this important? |
|--|---|---|--|
| In earlier versions, application model data for Microsoft Dynamics AX was always stored in a proprietary database that was named the Application Object Database (AOD). Each layer of the application had a corresponding .aod file. Microsoft Dynamics AX 2012 introduces the model store. Use the SQL Server-based model store to work with business data, or customer data. | The model store did not exist in Microsoft Dynamics AX 2009. Application model data was stored in .aod files. You can load .aod files during version upgrade. You can also load .aod files by using the Tools menu. | The following model features are new: <ul style="list-style-type: none"> • A model is a set of model elements in a specific layer. • Each layer consists of one or more models. One of the models is generated by the system. For example, VAR Model is the model that is generated for the VAR layer. • Each element in a layer must belong to only one model. • Models can be exported to a file artifact that is called a model file. Model files have an .axmodel extension. A model file is like an .aod file from earlier versions, but the | Storing models in SQL Server increases quality, reliability, and performance. In addition, you can use the tools that are available in SQL Server for backup and administration. |

names and numbers are not limited. Models in the model store can be exported to model files and imported from model files. Model files can be signed, and the signature is verified when the model files are installed.

- Model files replace .aod files as installation artifacts.
 - Models in the SQL Server-based model store replace .aod files that were used at run time.
 - Development is performed in the current model in any given layer. All development work, such as creating a new class, becomes part of the current model in the current layer. You can change the current model by clicking the name of the current model on the status bar, similarly to the way that you change the current company.
 - Any element that is created in the current layer can easily be moved to another model in the same layer.
 - The Application Object Tree (AOT) shows you which model a particular element belongs to, in addition to the layer tags.
 - The additional folder capabilities for version control have been renamed models, and the capabilities have been extended. When you add an element to version control, the element is added to the version-
-

controlled model.

- You can generate a MorphX project that contains all of the application objects in the model. A new **Model management** submenu on the **Tools** menu contains many tools that you can use to work with models and the model store.
- The new SysModel* system tables provide a view of the metadata that is associated with models. These tables enable you to use model metadata in **select** statements in your X++ code. For example, you can use the SysModelElementLabel table to retrieve the string value for the label that is associated with a particular model element.

Object IDs

| What can you do? | Microsoft Dynamics AX 2009 | Microsoft Dynamics AX 2012 | Why is this important? |
|---|-------------------------------|------------------------------|--|
| In earlier versions of Microsoft Dynamics AX, the number of object IDs that was available for application elements was limited, both in the range of IDs and by layer. In Microsoft Dynamics AX 2012, the 16-bit object ID is upgraded to 32 bits. This change exponentially increases the number of object IDs. Therefore, you no longer have to worry that you may run out of object IDs. | Object IDs were 16 bits long. | Object IDs are 32 bits long. | Changing the length of object IDs from 16 bits to 32 bits exponentially increases the number of object IDs that are available. |

AxUtil Command-Line Utility and PowerShell Cmdlets

| What can you do? | Microsoft Dynamics AX 2009 | Microsoft Dynamics AX 2012 | Why is this important? |
|--|--------------------------------|---|---|
| Use AxUtil to import and export application model data and work with the SQL Server model store. | The feature was not available. | AxUtil is a command-line utility that you can use to import and export .axmodel files into the SQL Server model store. You can also use AxUtil to delete one or more models, create new empty models, and list all of the models in a layer. All of the capabilities of AxUtil are also exposed as PowerShell CmdLets. | Users can use these tools to work with models from outside the development environment. Therefore, there are more options for scripting deployment processes. |

Development Workspace

| What can you do? | Microsoft Dynamics AX 2009 | Microsoft Dynamics AX 2012 | Why is this important? |
|--|---|---|--|
| In Microsoft Dynamics AX 2012, the development experience is separated from the application experience as a new workspace. You now use the development environment when you write code in Microsoft Dynamics AX. | Development occurred directly in the Application Workspace. | The Developer Workspace contains all of the tools and features that a developer must have to create and customize a Microsoft Dynamics AX application. Changes that you save in the Development Workspace are always synchronized with the Application Workspace. You can still open application elements in the AOT. You can also open an Application Workspace from the Development Workspace to view your customizations. You can open Microsoft Dynamics AX directly from a Development Workspace by | The new Development Workspace provides a more streamlined environment for writing code. The menus have also been customized to make it easier to find the tools and commands that you use when you write code. |

using the Ax32.exe
command-line flag
/development.

Some Layers Have Been Renamed

| What can you do? | Microsoft Dynamics AX 2009 | Microsoft Dynamics AX 2012 | Why is this important? | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|----------------------------|--|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|---|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|--|
| <p>In Microsoft Dynamics AX 2012, 16 layers are supported. Because multiple solutions can be installed in parallel into the same layer, some layers have been renamed to emphasize the new usage.</p> | <p>Old layer names:</p> | <p>New layer names:</p> | <p>The new layer names more accurately describe the usage of the layers.</p> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | <table border="1"> <tr><td>USP</td></tr> <tr><td>USR</td></tr> <tr><td>CUP</td></tr> <tr><td>CUS</td></tr> <tr><td>VAP</td></tr> <tr><td>VAR</td></tr> <tr><td>BUP</td></tr> <tr><td>BUS</td></tr> <tr><td>SL3</td></tr> <tr><td>SL2</td></tr> <tr><td>SL1</td></tr> <tr><td>HFX</td></tr> <tr><td>GLP</td></tr> <tr><td>GLS</td></tr> <tr><td>SYP</td></tr> <tr><td>SYS</td></tr> </table> | USP | USR | CUP | CUS | VAP | VAR | BUP | BUS | SL3 | SL2 | SL1 | HFX | GLP | GLS | SYP | SYS | <table border="1"> <tr><td>USP</td></tr> <tr><td>USR</td></tr> <tr><td>CUP</td></tr> <tr><td>CUS</td></tr> <tr><td>VAP</td></tr> <tr><td>VAR</td></tr> <tr><td>ISP</td></tr> <tr><td>ISV</td></tr> <tr><td>SLP</td></tr> <tr><td>SLN</td></tr> <tr><td>FPP</td></tr> <tr><td>FPK</td></tr> <tr><td>GLP</td></tr> <tr><td>GLS</td></tr> <tr><td>SYP</td></tr> <tr><td>SYS</td></tr> </table> | USP | USR | CUP | CUS | VAP | VAR | ISP | ISV | SLP | SLN | FPP | FPK | GLP | GLS | SYP | SYS | |
| USP | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| USR | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| CUP | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| CUS | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| VAP | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| VAR | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| BUP | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| BUS | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| SL3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| SL2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| SL1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| HFX | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| GLP | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| GLS | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| SYP | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| SYS | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| USP | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| USR | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| CUP | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| CUS | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| VAP | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| VAR | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ISP | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ISV | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| SLP | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| SLN | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| FPP | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| FPK | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| GLP | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| GLS | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| SYP | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| SYS | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Installation-Specific IDs

| What can you do? | Microsoft Dynamics AX 2009 | Microsoft Dynamics AX 2012 | Why is this important? |
|---|---|--|--|
| Object IDs, such as table IDs, field IDs, and class IDs, are now installation specific. Therefore, you no longer have to worry about conflicting object IDs in different installations. | Objects IDs were assigned when a model element was created. | <p>When a new model element is saved, imported, or installed, a unique ID is assigned to the model element at that installation site.</p> <p>For example, when a new class is added by a developer and saved to the model store, the class is assigned a class ID. However, when the same class is imported into another installation at a customer site, the class ID may be different from the ID that was assigned in the first installation site.</p> <p>The new object IDs that are assigned for Microsoft Dynamics AX 2012 installations have a larger range than the previous object IDs and will not conflict with any of the earlier versions of Microsoft Dynamics AX.</p> <p>In an upgrade scenario, object IDs are preserved, because they are assigned to the new LegacyId property on the application objects.</p> | <p>Because of installation-specific IDs, conflicts are avoided, because an ID is not assigned until installation time.</p> <p>Because the assignment of object IDs is handled at the installation site, Team Server no longer has to manage IDs. Team Server is no longer installed, and the setup of version control is no longer dependent on Team Server.</p> |

What's New: MorphX Tools for Developers in Microsoft Dynamics AX 2012

This section describes the new and enhanced features that are planned to be made available in the development tools for MorphX in Microsoft Dynamics AX 2012.

What is new or changed?

The following tools have been added or enhanced in Microsoft Dynamics AX 2012:

- X++ Editor
- Type Hierarchy Browser and Type Hierarchy Context

- Unit Test Framework
- Tools Used With Models and the Model Store
- Code Upgrade Tools

X++ Editor

| What can you do? | Microsoft Dynamics AX 2009 | Microsoft Dynamics AX 2012 | Why is this important? |
|--|--|---|---|
| Develop Microsoft Dynamics AX applications in the X++ code editor. | The code editor was unique to Microsoft Dynamics AX. | <p>The code editor is now based on the code editor for Visual Studio, and includes many of the same features and shortcuts. The following new and changed features are available:</p> <ul style="list-style-type: none"> • The method list displays all methods in the class. You can scroll through the list by using the mouse, and the size of the list is resized. • Basic editing has improved. For example, you can comment or uncomment code, move code by using a drag-and-drop operation, swap characters, delete a whole line, and change text in many other basic ways. • Code navigation is more similar to code navigation in Visual Studio. For example, you can use the Home key, navigate line ends, and go to the next or previous word as you do in Visual Studio. • It is easier to select code by using the margin or by using block selection, and word selection is | The new X++ editor adds features and is more familiar to developers who have worked with Visual Studio. |

improved.

- Code rendering is improved. For example, the value of a label is displayed when you rest the mouse pointer on the label, and an indicator in the margin tracks your changes. Additionally, line numbers are displayed, the font is improved, there is more syntax coloring, and you can view whitespace.
- You can search for code incrementally. Search uses .NET regular expressions.
- IntelliSense has been improved. IntelliSense now includes word completion, tooltips that display the XML documentation comments for a method or class, and help for multiline parameters.
- Errors in code are underlined by using a wavy line. When you rest the mouse pointer on the wavy line, the error message for the compiler is displayed. Code breakpoints can be toggled in the margin.

Type Hierarchy Browser and Type Hierarchy Context

| What can you do? | Microsoft Dynamics AX 2009 | Microsoft Dynamics AX 2012 | Why is this important? |
|-------------------------------|--|--|------------------------------------|
| View detailed | You used the Application | Use the Type Hierarchy Browser | The Type Hierarchy |

| | | | |
|--|------------------------|---|---|
| <p>information about elements in the Application Object Tree (AOT), and information about the hierarchy of those elements.</p> | <p>Hierarchy Tree.</p> | <p>to see detailed information about an element and its components, such as methods, fields, or properties, and the element from which the element inherits. In the form, you can view information about classes, tables, maps, views, extended data types, and enumerations.</p> | <p>Browser and Type Hierarchy Context provide more functionality than the Application Hierarchy Tree.</p> |
| | | <p>Use the Type Hierarchy Context to display a pane that shows the hierarchy details for a class, table, map, view, extended data type, or enumeration that is selected in the Application Object Tree (AOT). Click any element in the tree to open the Type Hierarchy Browser for that object.</p> | |

Unit Test Framework

| What can you do? | Microsoft Dynamics AX 2009 | Microsoft Dynamics AX 2012 | Why is this important? |
|--|---|--|--|
| <p>Attach a predefined attribute to a test class or test method to provide more information.</p> | <p>The feature was not available.</p> | <p>Use predefined attributes to specify that a test is active, or that it is a check-in test. You can also use predefined attributes to provide the target element that you are testing.</p> | <p>A test can be self-describing with regard to organization and when the test is run. For example, to run only the check-in tests, you can attach an attribute to the test that indicates that the test is a check-in test.</p> |
| <p>Add custom attributes to further categorize test cases.</p> | <p>You categorized test cases by creating them and adding them to a</p> | <p>Add and remove attributes from test cases to provide detail. For example, to satisfy</p> | <p>A test can be self-describing with regard to organization and when</p> |

Microsoft Dynamics AX

| | | | |
|--|---|--|---|
| | test project. | requirements for integration-level testing, you can attach an attribute to all tests that cross module boundaries. | the test is run. |
| At a glance, know whether a test case passed or failed. | You viewed the test detail by clicking the Details button on the Unit Test toolbar. Alternatively, you displayed results by using a listener. | In a test project, an icon appears next to each test case to indicate whether the test case passed or failed. | At a glance, you can understand how well a given project is running. |
| Rest the mouse pointer on a failed test case to display the exception that occurred. | You accessed the exception detail in a listener. | In a test project, rest the mouse pointer on a failed test case to display the exception that occurred. | At a glance, you can know the exceptions that were introduced for a project. |
| Use test metrics to measure performance. | The feature was not available. | Click the Details button on the Unit Test toolbar to view how long the test case ran. | You can use the duration metrics to help improve performance. |
| Use test metrics to ensure the integrity of your starting test data. | The feature was not available. | Click the Details button on the Unit Test toolbar to view the number of records that were updated in the database. | You can use the metrics that indicate the number of records that were updated to troubleshoot data integrity issues. When you run successive test cases, the integrity of the data that you start with for each test is very important. |

Tools Used With Models and the Model Store

| What can you do? | Microsoft Dynamics AX 2009 | Microsoft Dynamics AX 2012 | Why is this important? |
|--|--------------------------------|---|--|
| Select the model that you want to work with. | The feature was not available. | Use the Change Current Model tool to select the model that you want to work with. | You can now change the current model. |
| Use the Create Project from Model form. | The feature was not available. | Use this form to create a project from the contents of a model. | You can now create a project that contains only the contents of a model. |

| | | | |
|---------------------------------|--------------------------------|---|--|
| Use the Create Model Tool form. | The feature was not available. | Use this form to create a new model. | You can now create a new model. |
| Use the Import AOD File form. | The feature was not available. | Use this form to load an .aod file from an earlier version of Microsoft Dynamics AX into the model store. | You can now load an .aod file from an earlier version of Microsoft Dynamics AX into the model store. |

Code Upgrade Tools

| What can you do? | Microsoft Dynamics AX 2009 | Microsoft Dynamics AX 2012 | Why is this important? |
|---|--------------------------------|---|---|
| Detect conflicts when code is upgraded. | The feature was not available. | Use the Detect Code Upgrade Conflicts tool to analyze a system, and then create projects that contain the application objects that must be upgraded manually. | You can now see how much manual work is required to upgrade from an earlier version of Microsoft Dynamics AX. |
| Migrate Extended Data Type (EDT) relations. | The feature was not available. | Use the EDT Relation Migration tool to move the previous relations that are defined under the EDT nodes in the AOT to the table nodes. | In earlier versions of Microsoft Dynamics AX, the relations of tables to extended data types were stored in two locations: relations in tables, and relations in extended data types. In the current version of Microsoft Dynamics AX, relations in extended data types are obsolete. Instead, we recommend that developers migrate their EDT relations to table relations. The EDT Relation Migration tool makes it easy to transfer relations from extended data types to tables. |

What's New: Reporting for Developers in Microsoft Dynamics AX 2012

SQL Server Reporting Services is now the primary reporting platform for Microsoft Dynamics AX. The default, out-of-the-box reports that are planned to be made available with Microsoft Dynamics AX now run on the Reporting Services platform.

What is new or changed?

Reporting Services is a server-based reporting platform that provides comprehensive reporting functionality for a variety of data sources. Reporting Services includes a complete set of tools for you to create, manage, and deliver reports, as well as APIs that enable you to integrate or extend data and report processing in custom applications. The new report development environment takes advantage of extended SQL Server tools and components. The report development environment is fully integrated into Microsoft Visual Studio.

Reporting Improvements

The following table describes the improvements that have been made to the Microsoft Dynamics AX reporting framework.

| What can you do? | Microsoft Dynamics AX 2009 | Microsoft Dynamics AX 2012 | Why is this important? |
|---------------------------|---|--|--|
| Access data for a report. | Used Microsoft Dynamics AX queries, data from analysis cubes, external SQL Server databases, data methods that access data from Microsoft Dynamics AX or other sources, and managed code for data access. | Uses the data access methods available in Microsoft Dynamics AX 2009 and the report data provider (RDP) framework to define business logic for a report. | When you use Microsoft Dynamics AX queries, the reporting framework was enhanced to support display methods, field groups, and dimensions. When you add a field to a field group, the field is reflected throughout the application. This means that reports automatically add the field. Display methods are treated like fields in the query, even though the display method is retrieving the data from a different table. Support for dimensions means that you can create a report design using |

| | | | |
|---|--|---|---|
| | | | <p>dimension attributes in the data set and dimension attributes can be used as report parameters.</p> <p>A report data provider class lets you write X++ code to define your report business logic.</p> |
| Use labels in a report. | Each language locale required a .resx file to store localized strings. | Use AX labels in your reports that can be dynamically resolved at runtime. | Using AX labels in reports, you no longer have to create a .resx file for each language which improves the time to deploy reports. Only one report definition per report is needed. |
| Use data on your report that is based on specific time periods. | Feature not available. | Create a report that has data that is based on specific time periods. The reporting framework supports date effective data. | You do not have to handle date effective data explicitly for Microsoft Dynamics AX query based datasets. |
| Use an improved set of developer tools. | Used X++ Reporting Framework and Visual Studio 2008. | Use Visual Studio 2010 integration, take advantage of enhancements to auto-design reports, and create Dundas charts. | <p>In line with the long term goal to align with the Microsoft technology stack.</p> <p>Brings Reporting Services features to Microsoft Dynamics AX developers like rich charting and interactive reports. The following list describes additional improvements in the reporting tools:</p> <ul style="list-style-type: none"> • The Preview option is now enabled for all nodes under the design node. • A drop-down was added to display the possible parameter |

types.

- Microsoft PowerPivot and other Open Data Protocol (OData) Clients are now supported.
- SQL Server Reporting Services sub reports in precision design are now supported.
- Report Data Customization Extension (RDCE) uses Windows event viewer logging.
- New expressive charting capabilities for precision design reports.

| | | | |
|--|---|---|---|
| Process business logic for data on a report. | Used a report data method and managed code. | Use the report data provider (RDP) framework that supports complex business logic in X++ code. | The report data provider framework can support logic to handle data on a report like calculating an amount based on a specific field. |
| Use services to access data and business logic for a report. | Used .NET Business Connector, which has a weakly-typed programming model. | Create a custom service and access the service from a report. Queries and RDP use services to access data for a report. | Standards-based service interfaces enable you to integrate with many platforms. Services are strongly typed meaning that issues are found at design time instead of run time. A service was added by the reporting framework to consume a report data provider class from an external application. Services enable you to quickly expose existing |

| | | | |
|---|---|--|--|
| | | | X++ business logic without additional coding. |
| Work with large datasets for a report. | Used a report data method to return a data table. | Create reports that use large datasets. Added functionality now binds to data methods that return data in a page or stream of data. | Only the first row of a dataset must be executed in order to determine the report schema. This makes it easier to work with large datasets. Now you can write business logic that returns data in a page or a stream of data. You no longer need to fill the data table before you return it which was using memory and resources on the Reporting Server. |
| Use Microsoft Dynamics AX metadata to auto-format a report. | Defined the report format explicitly. | Create reports where the auto-formatting of the report is based on the formatting specified for the extended data types. | Display width, date/time and numeric formatting are set. All style and colors were updated to be correct. You can use metadata to handle whether a field or table should be visible on a report. |
| Define secure reports based on the role of the user. | Reports were secured using security keys. | Secure your reports by using role-based security and extensible data security (XDS) policies. When you save a report to the AOT, the securable objects will be added to the base permission set under the report design node. When a menu item is created and linked to a report design, the minimum permissions to execute the report are derived from the base | Report security is based on the user role and you can specify with more precision which data is displayed on a report. |

| | | | |
|--|---|---|--|
| | | permission set. XDS policies can be created to limit records that a user in a particular role can access. | |
| Create an unlimited number of dimensions attributes. | Limited to three default dimensions and seven user-definable dimension attributes. | Use the query picker, report model, and report generation that were updated to reflect improvements to the dimension framework. | Create an unlimited number of dimension attributes using the dimension framework. Additional flexibility is enabled by the ability to combine dimension attributes. |
| Reap the benefits of queries using a surrogate foreign key (SFK). | Report queries sourced data from tables that sometimes contain natural keys and the text for the fields was not always understandable. | Use queries that are using SFKs instead of natural keys. | Some of the benefits include improved readability, reduced database size, decreased time to upgrade, and fewer data integrity issues. A lookup was added to the ReportRunUI control for SFK parameters to let you filter on a SFK relationship in a report. |
| Access cross-reference information for a Reporting Services report. | Feature not available. | Use the cross-reference tool with Reporting Services reports. | You can see what data the report is using and what AOT objects are using the report. |
| Define a report that can be flipped from right to left based on the language that is set for the report. | Right to left flip of a report design was not supported. A new precision design was needed where all the report items were flipped manually which resulted in two designs for the same report. | Create a report that will flip right to left at runtime, based on the language that is set and properties set on the report. | You no longer need to maintain two report designs for reports that require a right to left flip. |
| AX_CompanyName report parameter has been updated to reflect changes in | At report design time, the company name report parameter was | The AX_CompanyName parameter is only added when the query is company | You can add organization parameters to your report such as Operating |

| | | | |
|--|---|---|---|
| Microsoft Dynamics AX 2012. | <p>added to each report.</p> <p>The AX_CompanyName report parameter was unique because it could not be deleted and it was not required to be initialized.</p> | <p>specific.</p> <p>You can initialize the AX_CompanyName parameter using built-in methods.</p> | <p>Unit or Department.</p> <p>The AX_CompanyName parameter was created by the Microsoft Dynamics AX framework for queries that are company specific.</p> <p>This change reflects the organization hierarchy product wide feature.</p> |
| Use the improved drill-through functionality on reports. | Feature not available. | <p>The following list describes the changes that were made to improve the drill-through functionality on reports.</p> <ul style="list-style-type: none"> • The client no longer uses table IDs, instead it uses table and field names to determine which menu item the report link should use. • The link style is no longer underlined and the color was changed. • The drill-through now opens in a new client window. | <p>Using table and field names reduces ambiguity and errors.</p> <p>The updated link style is consistent with expected link style. This improves the look and usability of reports that have links.</p> <p>Launching a new window for a linked report is consistent with expected behavior and improves the usability of reports that have links.</p> |

Parity with Existing Functionality

The following table describes the features that were added so that the Reporting Services reports have the same capabilities as the legacy X++ reports.

| Feature | Description |
|-------------------------------------|---|
| Modify queries used by reports | Allows you to filter and sort data on reports. |
| Control report destination settings | The PrintDestinationSettings class was added to allow you to programmatically change print settings such as saving to a file. |

Microsoft Dynamics AX

Create auto-reports

An auto-report is a report that is generated by clicking Print on a form. The data on the form is displayed on the report. Auto-reports are generated by using the Reporting Services framework for Microsoft Dynamics AX.

Print reports in a batch

You can use the Reporting Services framework to print a batch of reports, print a report on a recurring basis, and schedule a report to be printed during off peak hours.

Are there any special considerations?

Reports will not be upgraded from Microsoft Dynamics AX 2009 to Microsoft Dynamics AX 2012. Microsoft Dynamics AX 2012 provides hundreds of default, out-of-the-box reports that you can deploy and customize.

When upgrading, existing Reporting Services and X++ reports will be copied to the Microsoft Dynamics AX 2012 system, but they will not be upgraded. We recommend that you use a Reporting Services report that is provided with Microsoft Dynamics AX 2012 as a template and customize it to meet your needs.

What's New: Security for Developers in Microsoft Dynamics AX 2012

This section describes the new and enhanced security framework that is available in Microsoft Dynamics AX 2012.

What is new or changed?

Microsoft Dynamics AX 2012 has a new security framework, which is called role-based security. The system administrator and the developer each manage parts of the new security system. The developer creates and defines duties. The system administrator assigns duties and users to roles.

Many elements in the Application Object Tree (AOT) have properties that relate to role-based security. However, for the developer, the central location of role-based security is under the **Security** node.

Role-based security has mechanisms that help secure both columns and rows in tables. You can help secure the following elements:

- Tables and their fields – You can often help secure these elements by limiting access through menu items and forms that interact with the tables.
- Table records – You can help secure these elements based on the data values in the records.



Important:

Security keys are no longer used.

These mechanisms are described in the following sections:

- Security Permissions on Data Sources and Fields
- Security Policies on Table Records by Data Values

Security Permissions on Data Sources and Fields

| What can you do? | Microsoft Dynamics AX 2009 | Microsoft Dynamics AX 2012 | Why is this important? |
|---|--------------------------------|--|--|
| Create security permissions on individual AOT elements. | The feature was not supported. | <p>The system generates sets of permission specifications on forms and other AOT elements. The permission specifications are inferred by the system to match the details of each form. The developer can modify the permission specifications on any given form.</p> <p>The developer selects the set of permission specifications that applies to each menu item that refers to the form. The</p> | <p>The new security system takes less work to maintain as your business evolves. Security constructs can be reused more easily.</p> <p>Data that the user does not have permissions to see is never sent from Application Object Server (AOS) to the client, and appropriate security does not rely on forms for</p> |

developer uses the AOT to activate the selected permission specifications. The developer can combine many active permissions into one privilege. Active permissions are created in the AOT, under the Security > Permissions node. One privilege can correspond to the various actions that are required to perform one particular business task.

Either the system administrator or the developer can assign a privilege to a user role. All users who are assigned to the role gain that privilege.

Security Policies on Table Records by Data Values

| What can you do? | Microsoft Dynamics AX 2009 | Microsoft Dynamics AX 2012 | Why is this important? |
|--------------------|----------------------------|----------------------------------|------------------------|
| Implement security | The record level security | Security policies for table rows | Security policies can |

policies for table records.

(RLS) feature was used to filter access to table records. RLS security constructs applied only within individual tables. RLS was applied only if the developer explicitly called RLS to enforce security. Therefore, security enforcement could be accidentally omitted.

behave like a where clause in SQL. A security policy is based on a query under the Queries node in the AOT. The details of the policy are specified under the Data Sources > Ranges node. Therefore, foreign key relations between tables can be part of a security policy. Security policies are assigned to user roles.

restrict access to table rows, based on foreign key relationships between tables. One security policy can replace several RLS specifications.

After a security policy is created in the AOT and assigned to a user role, the system enforces the policy. Therefore, developers do not have to add calls to a security system in their code.

The RLS feature will be removed from a future version of the product.

What's New: Services & AIF for Developers in Microsoft Dynamics AX 2012

This page lists the new and enhanced developer features for Services and Application Integration Framework (AIF) that are planned to be made available in Microsoft Dynamics AX 2012.

Services and AIF both support integration with external systems. Services is the WS-* standards-based programming model that enables you to expose Microsoft Dynamics AX functionality through service interfaces. AIF is the framework that contains integration-specific features such as integration ports, data policies, pipelines, transforms, and so on.

What is new or changed?

This feature has changed considerably since Microsoft Dynamics AX 2009. Many new features have been added to support application development and integration with Microsoft Dynamics AX through services.

New features have been added to enable X++ business logic to be exposed as services and for X++ to consume services. In addition to programming model changes, AIF has been updated to enable developers to create transformations of non-XML data. The AIF system services enable you to retrieve user or metadata information and also to run queries.

- Services Enhancements
- Service Groups
- Support for Data Model Changes
- Business Operation Framework
- Support for Non-XML Files
- SOAP Headers
- Consume External Services in X++
- Search Message Log Information
- System Services

Services Enhancements

| What can you do? | Microsoft Dynamics AX 2009 | Microsoft Dynamics AX 2012 | Why is this important? |
|--|---|--|---|
| X++ business logic can easily be exposed as a service. | X++ classes had to implement a service interface in order to be exposed as a service. | X++ classes and members can be exposed as services and service operations by using attributes. | Enables you to easily expose X++ business logic as a service to external systems. This means that existing X++ code investments can be re-used. |
| Services support data | Only classes that | Any .NET or X++ native type | Enables you to more |

| | | | |
|--|--|--|---|
| contracts of any type. | implemented the AifSerializable interface could be used as service data contracts. | (such as a str or an int) can be used as a data contract. Any class that uses the AIF data contract attributes can be used as a data contract. | easily develop services because you can use all native types and complex types as data contracts. |
| Support for X++ types in data contracts. | Feature not available. | Data contracts can now use X++ container types and strongly typed X++ collections such as a Set or an Array. | Support for complex X++ types means that more complex data scenarios can supported in services. |

Service Groups

| What can you do? | Microsoft Dynamics AX 2009 | Microsoft Dynamics AX 2012 | Why is this important? |
|---|--|--|--|
| The ability to group services by using service groups was added to the AOT. | Services were managed at the individual service level. | A service group is a collection of services that are frequently consumed and managed together. All services in a service group are published in a single Web Services Definition Language (WSDL) file. | You only have to add a reference to a single WSDL to gain access to the service proxies for all the services in that service group. Service groups enable type sharing for types common across various services in a service group. For example, if two services in a service group take a parameter of type Address, that type is shared and can be seamlessly passed between the two services. |

Support for Data Model Changes

| What can you do? | Microsoft Dynamics AX 2009 | Microsoft Dynamics AX 2012 | Why is this important? |
|----------------------|----------------------------|---------------------------------|-------------------------|
| Extensive data model | Feature not available. | The document services framework | AIF fully supports data |

Microsoft Dynamics AX

changes are supported by AIF.

supports the data model changes implemented in Microsoft Dynamics AX 2012. The document services framework handles the following scenarios to produce a usable data contract schema:

model features to make data contracts easier to use.

- Replaces a surrogate foreign key (SFK) with the natural key for a specific table relationship. This means that the service contract has user-friendly, readable fields instead of obscure SFKs.
 - Handles all queries that contain the Global Address Book (GAB) views as one of their data sources.
 - Supports query data source tables that implement table inheritance.
 - Supports query data sources that contain tables with date effective fields.
 - Handles query data sources that contain tables with dimension columns. The framework supports serialization and deserialization of the dimension columns across the service boundary.
 - The wizards used to create and update document services support these data model changes.
-

Business Operation Framework

| What can you do? | Microsoft Dynamics AX 2009 | Microsoft Dynamics AX 2012 | Why is this important? |
|--|---|---|--|
| Business Operation Framework (BOF) replaces the RunBase framework. | Only the RunBase framework was available. User interface, business logic code, and data contract code were all contained in a single class. | BOF uses services to call from the client to the server so application "chattiness" is reduced. The BOF classes separate the user interface, business logic, and data contract logic. | You can use BOF to write applications that have reduced client to server communication. Because BOF uses services, there is no calling from the client back to the server, and an open connection from the server to the client cannot be maintained. The user interface for input parameters can be auto-generated. |

Support for Non-XML Files

| What can you do? | Microsoft Dynamics AX 2009 | Microsoft Dynamics AX 2012 | Why is this important? |
|--|----------------------------|--|--|
| AIF supports the import and export of data in non-XML formats. | Feature not available. | Data that is not XML, for example, a comma-delimited file, can now be imported into and exported from Microsoft Dynamics AX. This feature provides the ability for developers to create custom transformations that are used to transform non-XML data into a format that can be consumed by AIF. Custom transformations can be implemented by using an XSLT or a .NET DLL that implements the ITransform interface. | AIF can work with non-XML data, and you can handle data transformations within Microsoft Dynamics AX |

SOAP Headers

| What can you do? | Microsoft Dynamics AX 2009 | Microsoft Dynamics AX 2012 | Why is this important? |
|--|--|--|---|
| You can pass data into a service by using the SOAP header. | Developers had to manually add custom code for SOAP headers. | <p>SOAP headers have been extended. Now there is a strongly-typed parameter that is part of the service contract.</p> <p>You can add SOAP headers when you call a service to pass additional data into the service. The following SOAP headers are supported:</p> <ul style="list-style-type: none">• Company – The calling user’s default company.• Company Time Zone – The calling user’s default company time zone.• Currency – The currency associated with the user’s default company.• Language – The calling user’s default language.• Logon As Guest – Any client calling a service can pass this header and gain access to Microsoft Dynamics AX with guest permissions. The guest account must be enabled in Microsoft Dynamics AX for the call to be successful.• Logon As User – If the client calling the service is executing under the BC.NET proxy user account, you can use this header to pass a Microsoft Dynamics AX username. The service session will then execute as the specified user.• Preferred Calendar – The calling user’s preferred calendar. | Extended SOAP headers enable developers to provide more contextual information to a service when the client calls it. |

- User Time Zone – The calling user’s time zone.

Consume External Services in X++

| What can you do? | Microsoft Dynamics AX 2009 | Microsoft Dynamics AX 2012 | Why is this important? |
|---|---|--|--|
| Consume external web services from X++ without adding a service reference in the AOT. | You could consume an external Web service from X++, but you first had to add a service reference in the AOT. The ability to consume external services only worked in limited scenarios. For example, server authentication was not supported. | The Service References node has been removed from the AOT. You add a service reference in a .NET project and then add that project to the model store. Then you can access the external service from X++ code. | There is now full support for web services that have complex schemas in addition to support for web services authentication. The comprehensive deployment functionality in Visual Studio means that your service references are automatically deployed when you configure them to do so. |

Search Message Log Information

| What can you do? | Microsoft Dynamics AX 2009 | Microsoft Dynamics AX 2012 | Why is this important? |
|---|----------------------------|--|---|
| You can search all AIF log information. | Feature not available. | A new table links message and exception log data. You can do a full-text search for all records in the following AIF forms: <ul style="list-style-type: none"> • Exceptions • History • Queue manager | Troubleshooting integrations is easier because you can search all the AIF log data. |

System Services

| What can you do? | Microsoft Dynamics AX 2009 | Microsoft Dynamics AX 2012 | Why is this important? |
|--|----------------------------|--|--|
| Use system services to interact with and retrieve Microsoft Dynamics AX information. | Feature not available. | <p>System services are Windows Communication Foundation (WCF) services that are included with Microsoft Dynamics AX. The system services include the following:</p> <ul style="list-style-type: none"> • Metadata service – Enables you to query for data about the structure of your installation. You can retrieve metadata for labels, menus, tables, data types, enums, services, and so on. The metadata service should be used when you want to return information about the structures in Microsoft Dynamics AX to a client application. • Query service – Enables you to issue a query for data without using an AIF document service or creating a custom service. The query service returns data in a dataset and implements a paging mechanism so that you can manage queries that return large amounts of data. When you call the query service, you can specify that the query is run in one of three different ways: static query, user-defined query, or dynamic query. • User session service – Enables you to retrieve information about the | The system services open up Microsoft Dynamics AX development to other platforms because they enable you to retrieve system information via services. System services make it easier to develop applications that work with Microsoft Dynamics AX. |

calling user such as the user's default language, default company, default company time zone, and so on.

Are there any special considerations?

AIF integrations in Microsoft Dynamics AX 2009 will be upgraded to Microsoft Dynamics AX 2012. However, there are some manual upgrade actions that you may have to perform. The following points should be considered for any upgrade.

- The Microsoft Dynamics AX data upgrade process attempts a conversion of all AIF artifacts that correspond to artifacts in Microsoft Dynamics AX 2012. For example: the conversion of endpoints to integration ports. However, some artifacts cannot be upgraded because there is no equivalence in Microsoft Dynamics AX 2012, for example, endpoint constraints. After the upgrade is complete, you should verify the configuration of your integration ports and properties.
- There is no code upgrade for services code. All existing services must be manually updated with the required SysEntryPoint attribute and other access-level attributes so that they are deployed successfully.

What's New: Visual Studio Tools for Developers in Microsoft Dynamics AX 2012

This page lists the new and enhanced features for Visual Studio Tools that are planned to be made available in Microsoft Dynamics AX 2012.

Microsoft Dynamics AX now provides an integrated development experience with Visual Studio. Visual Studio Tools for Microsoft Dynamics AX (Visual Studio Tools) is a set of tools and functionality that support development for Microsoft Dynamics AX in .NET managed code. Visual Studio Tools contains tools and functionality from both the MorphX and Visual Studio development environments.

What is new or changed?

This feature has changed considerably since Microsoft Dynamics AX 2009. In addition to the new features that have been added, support for managed code has been integrated into Microsoft Dynamics AX in various ways.

New features have been added to the tools for modeling business application elements in Visual Studio. You can now add Visual Studio projects to Microsoft Dynamics AX and they are stored in the same model store as X++ code. You can view Visual Studio projects from either the Application Explorer in Visual Studio or the Application Object Tree (AOT) in MorphX. Reporting projects created in Visual Studio are stored in granular and customizable elements in the AOT. Source control is available for application projects from either the AOT or Visual Studio. The following sections describe the components that comprise Visual Studio Tools.

- Application Explorer
- Managed Code Business Logic
- Proxies
- Managed Code Event Handlers
- Deployment
- Cross-Reference Tool
- Code Upgrade Tools

Application Explorer

| What can you do? | Microsoft Dynamics AX 2009 | Microsoft Dynamics AX 2012 | Why is this important? |
|---|--|--|---|
| The Application Explorer displays a view of the AOT within the Visual Studio IDE. | Model View enabled developers to view, create, edit, and delete models in a reporting project. | <ul style="list-style-type: none">• View, open for edit, and delete code elements in the model store whether they are managed code or X++ elements• Edit reports and Enterprise Portal Web controls | Enables you to see all objects in the model store from the Visual Studio IDE. |

- Add managed code projects to the model store
- Edit managed projects that have been added to the AOT (model store)
- View X++ code

Managed Code Business Logic

| What can you do? | Microsoft Dynamics AX 2009 | Microsoft Dynamics AX 2012 | Why is this important? |
|--|----------------------------|--|--|
| You can write business logic in .NET languages and manage this code in Microsoft Dynamics AX just as you would with X++ classes. | Feature not available. | <ul style="list-style-type: none"> <input type="checkbox"/> You can save .NET class library projects to the model store. <input type="checkbox"/> You can customize these managed code projects using the various layers. <input type="checkbox"/> You can build managed code projects in Visual Studio and they are automatically deployed if configured to do so. | Developing code that integrates with Microsoft Dynamics AX is now possible in Visual Studio. The full development and deployment scenario is supported for managed code. |

Proxies

| What can you do? | Microsoft Dynamics AX 2009 | Microsoft Dynamics AX 2012 | Why is this important? |
|---|----------------------------|---|--|
| Allows you to add an X++ class or table to your project in Visual Studio so it can be accessed by managed code. | Feature not available. | When you add an X++ class or table to a project by using the Application Explorer, a proxy for that class is created internally by the system. After the proxy is created, that type is then available as a strong type and features like IntelliSense are available. | You can easily add X++ objects to your Visual Studio project and write managed code that accesses those objects. |

Microsoft Dynamics AX

Managed Code Event Handlers

| What can you do? | Microsoft Dynamics AX 2009 | Microsoft Dynamics AX 2012 | Why is this important? |
|---|----------------------------|--|---|
| You can create a managed code event handler for an instance method of an X++ class. | Feature not available. | The event handler subscribes to the X++ event and can be called before the method begins or after the method completes. You can handle events raised only by a class in the Classes node in the AOT. | You can now create event handlers in managed code as well as X++. This means you can write .NET code to handle class events. |
| You can create a managed code event handler for an X++ delegate. | Feature not available. | The event handler subscribes to the X++ delegate, and the code runs when the delegate fires. | You can now create event handlers in managed code as well as X++. This means you can write .NET code to handle delegate events. |

Deployment

| What can you do? | Microsoft Dynamics AX 2009 | Microsoft Dynamics AX 2012 | Why is this important? |
|---|----------------------------|---|---------------------------------------|
| The deployment functionality enables you to automatically deploy managed assemblies (DLLs) that you create in Visual Studio to Microsoft Dynamics AX. | Feature not available. | Managed code assemblies are deployed to the location that you specify: client, server, Reporting Services or Enterprise Portal. You can specify where the assembly is deployed by setting project properties in Visual Studio. After you deploy an assembly, you can then see the managed code classes via IntelliSense and call the managed code from X++. | Deploys to the locations you specify. |

Cross-Reference Tool

| What can you do? | Microsoft Dynamics AX 2009 | Microsoft Dynamics AX 2012 | Why is this important? |
|---|----------------------------|--|---|
| The Cross-reference Tool in Microsoft Dynamics AX lets you see the relationships between elements and now supports managed code that has been added to the model store. | Feature not available. | In order to see cross-references for a Visual Studio project, you must first add the project to the model store by using the Application Explorer. After you add a project to the model store, the project and its cross-reference information are updated every time you build the project. | Managed code is now supported by the Cross-reference Tool in MorphX. This means you can see the same cross-reference information for both X++ and managed code. For example, when you access the Cross-reference Tool from a report, you can see what the report data source. |

Code Upgrade Tools

| What can you do? | Microsoft Dynamics AX 2009 | Microsoft Dynamics AX 2012 | Why is this important? |
|--|----------------------------|--|---|
| Microsoft Dynamics AX supports layer-based | Feature not available. | The conflict detection and resolution tools enable | The code upgrade tools that are included with |

Microsoft Dynamics AX

customizations for managed code. Therefore, the tools that enable developers to find and resolve conflicts that result from those customizations also support managed code. The code upgrade tools enable developers to detect and resolve conflicts between two versions of Microsoft Dynamics AX.

developers to find conflicts and compare customizations in different layers. These tools are available in the MorphX IDE and include the following:

- Detect code upgrade conflicts tool - Reports conflicts in both X++ and managed code that has been added to the model store.
- Compare tool - Enables you to compare both X++ code and managed code in two layers and resolve any conflicts.

Microsoft Dynamics AX support managed code development. Therefore, the same code upgrade process can be used for code whether it originates in MorphX or in Visual Studio.

These tools are used to detect and resolve code conflicts when upgrading to a new version of Microsoft Dynamics AX, installing customizations, or installing new modules developed by a VAR or partner.

What's New: Workflow for Developers in Microsoft Dynamics AX 2012

This section lists the new and enhanced features that are planned to be made available for Workflow in Microsoft Dynamics AX 2012.

What is new or changed?

Workflow has changed considerably since Microsoft Dynamics AX 2009. The following features have been added or changed:

- Advanced Workflow Controls
- Line Item Workflow
- Automated Workflow
- Workflow Editor
- Workflow Templates Are Now Workflow Types

Advanced Workflow Controls

| What can you do? | Microsoft Dynamics AX 2009 | Microsoft Dynamics AX 2012 | Why is this important? | | | | |
|--|--|---|------------------------|-------------|-------|--|--|
| New workflow control elements are available. These elements provide additional capabilities for flow control to increase the automation of a workflow. | The feature was not available. | The following table describes the new workflow control elements. <table border="1" data-bbox="711 1186 1161 1881"> <thead> <tr> <th>Workflow element</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>Start</td> <td>Represents one or more starting points in a workflow, so that the intent of the workflow at these points is clear. This workflow element is used at the start of a workflow, or at intermediate starting points in a workflow, such as in a parallel workflow.</td> </tr> </tbody> </table> | Workflow element | Description | Start | Represents one or more starting points in a workflow, so that the intent of the workflow at these points is clear. This workflow element is used at the start of a workflow, or at intermediate starting points in a workflow, such as in a parallel workflow. | By using advanced workflow controls, you can create more complex workflows than you could in Microsoft Dynamics AX 2009. |
| Workflow element | Description | | | | | | |
| Start | Represents one or more starting points in a workflow, so that the intent of the workflow at these points is clear. This workflow element is used at the start of a workflow, or at intermediate starting points in a workflow, such as in a parallel workflow. | | | | | | |

| | |
|----------|---|
| Sequence | Represents a workflow in which there is a dependency between two other workflow elements. |
|----------|---|

| | |
|----------------------------------|--|
| Exclusive fork yes/no gateway | Provides a step where the user decides which branch of a workflow is followed. This element resembles the current workflow task element, except that the user can choose between two decision-based outcomes, and workflow branching is based on the selected outcome. |
|----------------------------------|--|

| | |
|---|---|
| Exclusive fork data-based gateway | Provides a step where an evaluation of data determines which branch of a workflow is followed. When a workflow reaches this step, data from the database or the user is evaluated, and workflow branching is based on the evaluation of the data. |
|---|---|

| | |
|-----------------|------------------------------------|
| Parallel branch | Provides a way to split a workflow |
|-----------------|------------------------------------|

into two or more parallel workflow branches. These branches are run separately, and then joined before the workflow continues to the next workflow element.

| | |
|---------------------|--|
| Intermediate branch | Provides a way to initiate a new workflow when a business event occurs. For example, a new workflow can be initiated when an order is placed, an invoice is received, or a payment is made. |
| End | Represents one or more termination points in a workflow, so that the intent of the workflow at these points is clear. This workflow element is used at the end of a workflow, or at intermediate end points in a workflow, such as in a parallel workflow. |

Microsoft Dynamics AX

Line Item Workflow

| What can you do? | Microsoft Dynamics AX 2009 | Microsoft Dynamics AX 2012 | Why is this important? |
|---|--------------------------------|--|---|
| You can now create a line item workflow that is configured and instantiated for different types of line item records that are in a header-to-line relationship. For example, a header can be a sales order that has sales order lines, or a purchase order that has purchase order lines. | The feature was not available. | Each line item that is associated with the header has its own workflow processing, depending on the conditions that are set up for that workflow. When workflow processing is completed for all of the individual line items, the overall workflow can continue. | You can create a separate workflow for each line item that is associated with a header. |

Automated Workflow

| What can you do? | Microsoft Dynamics AX 2009 | Microsoft Dynamics AX 2012 | Why is this important? |
|--|--------------------------------|--|--|
| You can now create an automated workflow element. An automated workflow element is a type of workflow element that does not involve human interaction. | The feature was not available. | An automated workflow element runs business logic as part of a workflow that is implemented to automate a business process. For example, automated tasks can execute a credit check, check inventory, run a report, perform calculations, update records, or start another workflow. | You can now create workflows that do not require human intervention. |

Workflow Editor

| What can you do? | Microsoft Dynamics AX 2009 | Microsoft Dynamics AX 2012 | Why is this important? |
|---|---|---|--|
| You can now create a workflow by using the Workflow Editor. | In Microsoft Dynamics AX 2009, workflows were configured by using the | The Workflow Editor is a managed code user interface that is used to graphically define | It is now much simpler to create a workflow by using a drag-and-drop |

Workflow Configuration user interface. and configure workflows in Microsoft Dynamics AX. The editor is based on the Windows Workflow Foundation Workflow Editor for the flow chart model. user interface.

Workflow Templates Are Now Workflow Types

| What can you do? | Microsoft Dynamics AX 2009 | Microsoft Dynamics AX 2012 | Why is this important? |
|---|---|--|--|
| Workflow templates have been renamed workflow types. Workflow types are building blocks that can be used to create customized workflows that enforce business policies. Workflow types are defined in the Application Object Tree (AOT) at design time. The metadata from workflow types is used by the customer to create a workflow configuration. For example, workflow tasks can have metadata that indicates the class that is used when the due date of the task is calculated. | In Microsoft Dynamics AX 2009, workflow types were known as workflow templates. | Workflow templates have been renamed workflow types. | All mentions of workflow templates have been replaced by workflow types. |

What's New: X++ for Developers in Microsoft Dynamics AX 2012

This section describes the new and enhanced X++ features that are planned to be made available in Microsoft Dynamics AX 2012.

What is new or changed?

This section provides details about the improvements that were made in X++.

Microsoft Dynamics AX

- .NET Proxies to X++ Classes
- X++ Attributes
- New Expression Operators in X++ for Inheritance
- Events in X++
- X++ Compiled to .NET CIL is Faster

.NET Proxies to X++ Classes

| What can you do? | Microsoft Dynamics AX 2009 | Microsoft Dynamics AX 2012 | Why is this important? |
|--|--------------------------------|---|---|
| Create proxy classes to represent .NET interop to X++. | The feature was not supported. | <p>A .NET proxy is a .NET managed class that represents a class or table that exists in Microsoft Dynamics AX. Your managed program can call proxy methods, just as your X++ code can call methods on a table or class.</p> <p>The C# compiler detects errors in your calls to the proxy at compile time. You can use this early-bound programming model to fix errors in your code before the code is run. The internal logic of the proxy class uses the late-bound programming model of .NET Business Connector.</p> <p>However, the tools that generate the source code of the proxy have been rigorously tested.</p> | Proxy classes enable you to write managed classes that interoperate with your X++ classes, to form a unified application. |

X++ Attributes

| What can you do? | Microsoft Dynamics AX 2009 | Microsoft Dynamics AX 2012 | Why is this important? |
|---|--------------------------------|---|--|
| Use X++ attribute classes for metadata. | The feature was not supported. | X++ classes and methods can now have metadata in the form of attribute classes. An attribute class is any non-abstract class that inherits from the SysAttribute class. Detailed metadata is assigned through | By using attributes, you can associate declarative information with X++ code. After an attribute is associated with a program entity, the attribute can be queried |

the constructor of the attribute class. at run time and used in many ways.

Tables and interfaces can also have attribute classes.

New Expression Operators in X++ for Inheritance

| What can you do? | Microsoft Dynamics AX 2009 | Microsoft Dynamics AX 2012 | Why is this important? |
|--|---|---|---|
| Use the is operator to test whether X++ objects are a type or a subtype of a specific class. | The feature was not supported. | The X++ language now has the expression operator is. You use the is operator to test whether an object is a type or a subtype of a specific class or table. | When the is operator is used, your code can test whether a particular downcast assignment would be valid. |
| Use the as operator to make downcast assignments explicit. | The feature was not supported. In Microsoft Dynamics AX 2009, downcast assignments were implicit. Flawed X++ code sometimes assigned an object to a variable type that was outside the inheritance hierarchy of the object. This incorrect assignment sometimes caused confusing errors during run time. | You can use the as operator to make downcast assignments explicit. | When the as operator is used to make a downcast assignment explicit, the incorrect assignment of the object is prevented during run time. The null value is assigned instead. Therefore, it is easier to diagnose the run-time error. |

Events in X++

| What can you do? | Microsoft Dynamics AX 2009 | Microsoft Dynamics AX 2012 | Why is this important? |
|---|--------------------------------|---|--|
| Implement events and event handling in X++. | The feature was not supported. | The X++ language now has the delegate keyword. When program conditions meet the programmer's criteria for the event, the X++ code can call the delegate. The delegate then calls all the event handler | The event programming model can reduce the chance that your customizations are affected by upgrades to your program or to Microsoft Dynamics AX. |

methods that were subscribed to the delegate.

The members of a class can include both methods and delegates.

You can subscribe methods to a delegate by dragging the methods onto the node for the delegate in the Application Object Tree (AOT).

Alternatively, you can subscribe methods to the delegate in X++ code by using the += operator together with the **eventHandler** keyword.

An X++ event can be handled by event handler methods that are written in either X++ or managed languages such as C#.

Pre-method and post-method events are generated by the system.

The feature was not supported.

The methods on your X++ class each raise an event immediately before they start. The methods raise another event immediately after they end. These two events offer opportunities for custom code to intervene in the program flow.

The pre-method event handler can modify the parameter values before the values are passed to the method. Before the caller of the method receives the value that is returned by the method, the post-method event handler can modify the return value.

The **CalledWhen** property of each event handler in the AOT Properties window makes it easy to choose between a subscription to the **Pre** or **Post**

The pre-method and post-method events are stable points at which custom code can run.

event of the method.

X++ Compiled to .NET CIL is Faster

| What can you do? | Microsoft Dynamics AX 2009 | Microsoft Dynamics AX 2012 | Why is this important? |
|--------------------------|--------------------------------|---|---|
| Speed up X++ batch jobs. | The feature was not supported. | <p>Some X++ batch jobs can now complete in much less time. All X++ code that runs on the AX32.exe client still runs as interpreted p-code. However, all batch jobs now run as Microsoft.NET Framework CIL, which is often much faster than p-code.</p> <p>The X++ developer must now complete the extra step of compiling X++ p-code to CIL. To compile p-code to CIL, right-click AOT, and then click Add-ins > Incremental CIL generation from X++.</p> <p>Next, if the installation uses multiple instances of Application Object Server (AOS), all AOS instances must be stopped and then restarted. This causes the AOS instances to load the updated assembly.</p> <p>Finally, a service or a batch job must run the X++ code that was compiled to CIL. Services and batch jobs now run only as CIL.</p> | Some batch jobs run much faster as CIL than they did as p-code. |
