Microsoft Dynamics® AX 2012

Implementing the Agreement Framework

White Paper

The Agreement Framework in Microsoft Dynamics AX 2012 replaces the data model and functionality that were provided by blanket orders in Microsoft Dynamics AX 2009. This new framework offers users of Microsoft Dynamics AX a broad set of tools for applying and following up on commercial agreements between the company and its customers and vendors.

December 2011

www.microsoft.com/dynamics/ax

Oleg Lisovoy, Software Development Engineer

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

Microsoft Dynamics®

Table of Contents

Introduction

The Agreement Framework in Microsoft Dynamics AX 2012 replaces the data model and functionality that were provided by blanket orders in Microsoft Dynamics AX 2009. In Microsoft Dynamics AX 2009, blanket orders were used to establish a contract between two parties for the sale or purchase of a specific quantity of a specific item at a special price.

This new framework offers users of Microsoft Dynamics AX a broad set of tools for applying and following up on commercial agreements between the company and its customers and vendors. Agreements may relate to the purchase or sale of a specific quantity or volume of a particular item. Agreements may also relate to a range of items within a specific category, to which special policies have been applied to set an agreed price for the trade of goods or services.

The new physical data model physical data model is based on the advanced technologies that are available in Microsoft Dynamics AX 2012 for data modeling. The physical data model completely separates Agreement Framework's data from general-purpose orders.

In Microsoft Dynamics AX 2012, the new Agreement Framework enables you to do the following:

- Set up a contractual commitment to sell or purchase a specific amount or quantity of a specific item, a specific category, or any combination of items and categories.
- Enforce price agreements for the purchases or sales that are subject to the agreement.
- Follow up on sales or purchases to see whether the contractual obligations have been fulfilled.
- Enable volume-based commitments.
- Manage category-based commitments.
- When you create orders, apply commitments that the system automatically suggests.
- Achieve a clean physical data model that does not include irrelevant data and is suitable for further extension and customization.

This document highlights the main features of the new Microsoft Dynamics AX 2012 physical data model and indicates various entry points for possible customization.

Background

In Microsoft Dynamics AX 2009, the blanket order functionality was integrated into general-purpose order subsystems. In the other words, blanket orders were considered another type of sales order or purchase order.

Because of this approach, the data structures that were used by blanket orders were not optimal. The approach also required a cumbersome code organization that had to maintain a class hierarchy behind the data tables, and constant conditional code deviation based on the type of data that was stored in a single data table.

As a result, blanket orders provided limited capability to enforce prices for items that were sold or purchased according a particular blanket order. In addition, the capability to follow up on a particular blanket order was almost nonexistent.

The blanket order functionality was also limited, because it handled only quantities of particular stocked items, and did not provide the capability to establish contractual obligations for the sale or purchase of a specific volume of goods or services within a particular category.

Agreement Framework physical data model

In Microsoft Dynamics AX 2012, the Agreement Framework's data model has been designed to be completely separate from the general-purpose order data model. Such a separation enables Agreement Framework entities to achieve an optimal data structure in which entities that are stored in the database do not have to have properties that are not related to agreements.

The Agreement Framework physical data model has also been designed to use the new Microsoft Dynamics AX 2012 table inheritance feature, and to implement all the data subtypes that the framework requires, and the behavioral deviations of those subtypes, as a table inheritance ladder. These design features encapsulate both data-related and functional specializations within a single type of object, Microsoft Dynamic AX data tables. As a result, a class hierarchy no longer has to be maintained behind the data tables that are used.

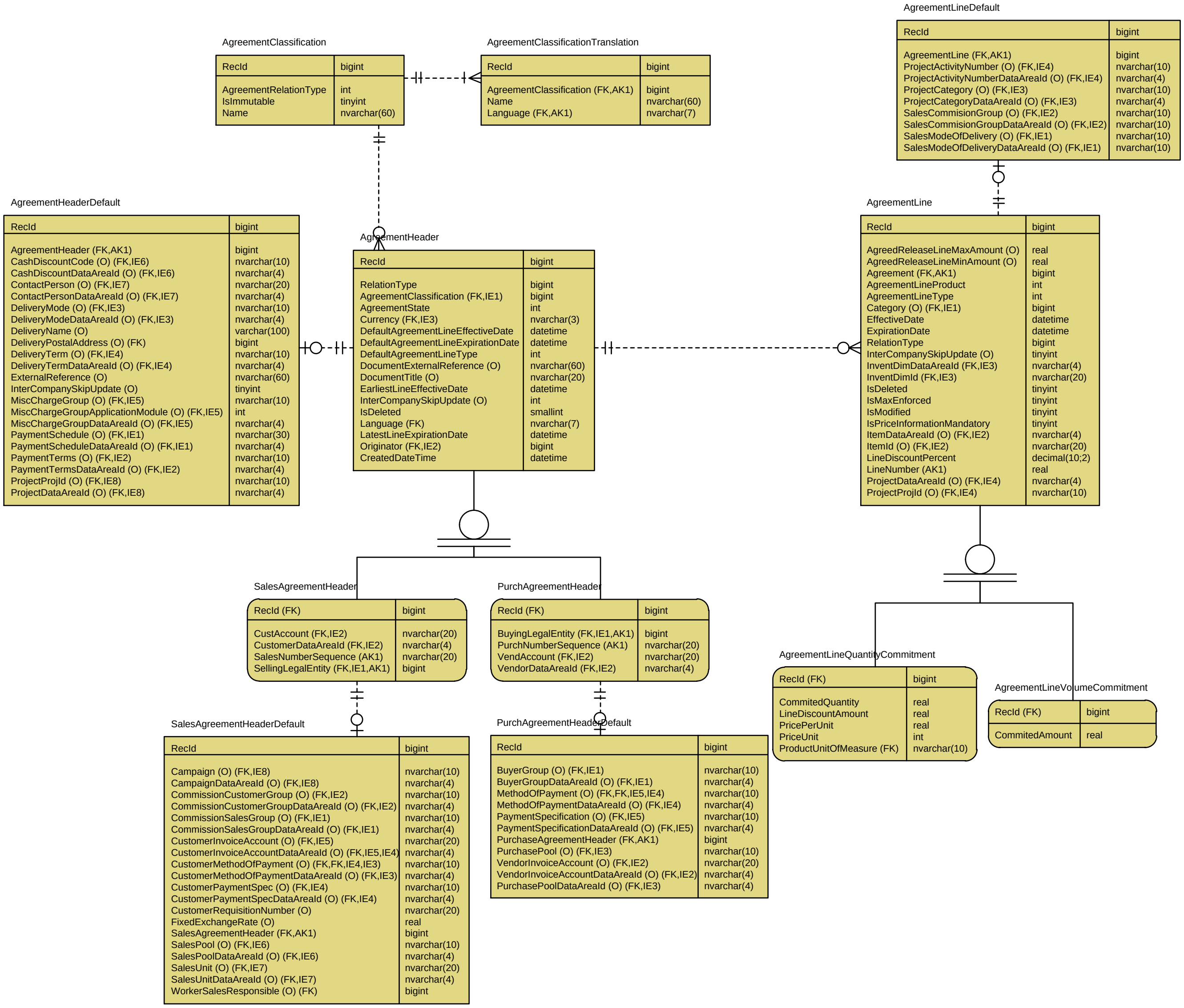
The Agreement Framework physical data model is based on the principles of data normalization and controlled data denormalization (in several well-justified scenarios, for performance reasons). Therefore, the Agreement Framework physical data model provides a logical and balanced way to store and access data.

A complete description of the physical data model of the new Agreement Framework in Microsoft Dynamics AX 2012 is beyond the scope of this paper. Instead, this document presents several fragments of the physical data model that depict the Agreement Framework's main functional subsystems.

Agreement Framework physical data model: Main entities

Figure 1 illustrates the main entities of the Agreement Framework physical data model.

Note: To help you establish the relationship between an entity in the physical data model and a table object in the Application Object Table (AOT) in Microsoft Dynamics AX, all the illustrations of physical data model fragments in this document use the actual names of Microsoft Dynamics AX table objects as entity names. Likewise, the actual names of Microsoft Dynamics AX table columns are used as entity properties.



AgreementClassification

RecId	bigint
AgreementRelationType	int
IsImmutable	tinyint
Name	nvarchar(60)

AgreementClassificationTranslation

RecId	bigint
AgreementClassification (FK,AK1)	bigint
Name	nvarchar(60)
Language (FK,AK1)	nvarchar(7)

AgreementLineDefault

RecId	bigint
AgreementLine (FK,AK1)	bigint
ProjectActivityNumber (O) (FK,IE4)	nvarchar(10)
ProjectActivityNumberDataAreaId (O) (FK,IE4)	nvarchar(4)
ProjectCategory (O) (FK,IE3)	nvarchar(10)
ProjectCategoryDataAreaId (O) (FK,IE3)	nvarchar(4)
SalesCommisionGroup (O) (FK,IE2)	nvarchar(10)
SalesCommisionGroupDataAreaId (O) (FK,IE2)	nvarchar(10)
SalesModeOfDelivery (O) (FK,IE1)	nvarchar(10)
SalesModeOfDeliveryDataAreaId (O) (FK,IE1)	nvarchar(10)

AgreementHeaderDefault

RecId	bigint
AgreementHeader (FK,AK1)	bigint
CashDiscountCode (O) (FK,IE6)	nvarchar(10)
CashDiscountDataAreaId (O) (FK,IE6)	nvarchar(4)
ContactPerson (O) (FK,IE7)	nvarchar(20)
ContactPersonDataAreaId (O) (FK,IE7)	nvarchar(4)
DeliveryMode (O) (FK,IE3)	nvarchar(10)
DeliveryModeDataAreaId (O) (FK,IE3)	nvarchar(4)
DeliveryName (O)	varchar(100)
DeliveryPostalAddress (O) (FK)	bigint
DeliveryTerm (O) (FK,IE4)	nvarchar(10)
DeliveryTermDataAreaId (O) (FK,IE4)	nvarchar(4)
ExternalReference (O)	nvarchar(60)
InterCompanySkipUpdate (O)	tinyint
MiscChargeGroup (O) (FK,IE5)	nvarchar(10)
MiscChargeGroupApplicationModule (O) (FK,IE5)	int
MiscChargeGroupDataAreaId (O) (FK,IE5)	nvarchar(4)
PaymentSchedule (O) (FK,IE1)	nvarchar(30)
PaymentScheduleDataAreaId (O) (FK,IE1)	nvarchar(4)
PaymentTerms (O) (FK,IE2)	nvarchar(10)
PaymentTermsDataAreaId (O) (FK,IE2)	nvarchar(4)
ProjectProjId (O) (FK,IE8)	nvarchar(10)
ProjectDataAreaId (O) (FK,IE8)	nvarchar(4)

AgreementHeader

RecId	bigint
RelationType	bigint
AgreementClassification (FK,IE1)	bigint
AgreementState	int
Currency (FK,IE3)	nvarchar(3)
DefaultAgreementLineEffectiveDate	datetime
DefaultAgreementLineExpirationDate	datetime
DefaultAgreementLineType	int
DocumentExternalReference (O)	nvarchar(60)
DocumentTitle (O)	nvarchar(20)
EarliestLineEffectiveDate	datetime
InterCompanySkipUpdate (O)	int
IsDeleted	smallint
Language (FK)	nvarchar(7)
LatestLineExpirationDate	datetime
Originator (FK,IE2)	bigint
CreatedDateTime	datetime

AgreementLine

RecId	bigint
AgreedReleaseLineMaxAmount (O)	real
AgreedReleaseLineMinAmount (O)	real
Agreement (FK,AK1)	bigint
AgreementLineProduct	int
AgreementLineType	int
Category (O) (FK,IE1)	bigint
EffectiveDate	datetime
ExpirationDate	datetime
RelationType	bigint
InterCompanySkipUpdate (O)	tinyint
InventDimDataAreaId (FK,IE3)	nvarchar(4)
InventDimId (FK,IE3)	nvarchar(20)
IsDeleted	tinyint
IsMaxEnforced	tinyint
IsModified	tinyint
IsPriceInformationMandatory	tinyint
ItemDataAreaId (O) (FK,IE2)	nvarchar(4)
ItemId (O) (FK,IE2)	nvarchar(20)
LineDiscountPercent	decimal(10;2)
LineNumber (AK1)	real
ProjectDataAreaId (O) (FK,IE4)	nvarchar(4)
ProjectProjId (O) (FK,IE4)	nvarchar(10)

SalesAgreementHeader

RecId (FK)	bigint
CustAccount (FK,IE2)	nvarchar(20)
CustomerDataAreaId (FK,IE2)	nvarchar(4)
SalesNumberSequence (AK1)	nvarchar(20)
SellingLegalEntity (FK,IE1,AK1)	bigint

PurchAgreementHeader

RecId (FK)	bigint
BuyingLegalEntity (FK,IE1,AK1)	bigint
PurchNumberSequence (AK1)	nvarchar(20)
VendAccount (FK,IE2)	nvarchar(20)
VendorDataAreaId (FK,IE2)	nvarchar(4)

AgreementLineQuantityCommitment

RecId (FK)	bigint
CommitedQuantity	real
LineDiscountAmount	real
PricePerUnit	real
PriceUnit	int
ProductUnitOfMeasure (FK)	nvarchar(10)

AgreementLineVolumeCommitment

RecId (FK)	bigint
CommitedAmount	real

SalesAgreementHeaderDefault

RecId	bigint
Campaign (O) (FK,IE8)	nvarchar(10)
CampaignDataAreaId (O) (FK,IE8)	nvarchar(4)
CommissionCustomerGroup (O) (FK,IE2)	nvarchar(10)
CommissionCustomerGroupDataAreaId (O) (FK,IE2)	nvarchar(4)
CommissionSalesGroup (O) (FK,IE1)	nvarchar(10)
CommissionSalesGroupDataAreaId (O) (FK,IE1)	nvarchar(4)
CustomerInvoiceAccount (O) (FK,IE5)	nvarchar(20)
CustomerInvoiceAccountDataAreaId (O) (FK,IE5,IE4)	nvarchar(4)
CustomerMethodOfPayment (O) (FK,FK,IE4,IE3)	nvarchar(10)
CustomerMethodOfPaymentDataAreaId (O) (FK,IE3)	nvarchar(4)
CustomerPaymentSpec (O) (FK,IE4)	nvarchar(10)
CustomerPaymentSpecDataAreaId (O) (FK,IE4)	nvarchar(4)
CustomerRequisitionNumber (O)	nvarchar(20)
FixedExchangeRate (O)	real
SalesAgreementHeader (FK,AK1)	bigint
SalesPool (O) (FK,IE6)	nvarchar(10)
SalesPoolDataAreaId (O) (FK,IE6)	nvarchar(4)
SalesUnit (O) (FK,IE7)	nvarchar(20)
SalesUnitDataAreaId (O) (FK,IE7)	nvarchar(4)
WorkerSalesResponsible (O) (FK)	bigint

PurchAgreementHeaderDefault

RecId	bigint
BuyerGroup (O) (FK,IE1)	nvarchar(10)
BuyerGroupDataAreaId (O) (FK,IE1)	nvarchar(4)
MethodOfPayment (O) (FK,FK,IE5,IE4)	nvarchar(10)
MethodOfPaymentDataAreaId (O) (FK,IE4)	nvarchar(4)
PaymentSpecification (O) (FK,IE5)	nvarchar(10)
PaymentSpecificationDataAreaId (O) (FK,IE5)	nvarchar(4)
PurchaseAgreementHeader (FK,AK1)	bigint
PurchasePool (O) (FK,IE3)	nvarchar(10)
VendorInvoiceAccount (O) (FK,IE2)	nvarchar(20)
VendorInvoiceAccountDataAreaId (O) (FK,IE2)	nvarchar(4)
PurchasePoolDataAreaId (O) (FK,IE3)	nvarchar(4)

Figure 1: Main entities of the Agreement Framework physical data model

Figure 1 illustrates the first important difference between the Agreement Framework physical data model in Microsoft Dynamics AX 2012 and the blanket order data model in Microsoft Dynamics AX 2009: entities that are related to the Agreement Framework do not reuse the Microsoft Dynamics AX general order tables (SalesTable, SalesLine, PurchTable, and PurchLine) but persist all their data in new dedicated tables. These new tables use existing Microsoft Dynamics AX naming conventions. Therefore, the name of each new table begins with the prefix Agreement, SalesAgreement, or PurchAgreement.

The second important difference in the new Agreement Framework physical data model is that many of its entities use the new Microsoft Dynamics AX 2012 table inheritance feature. This feature allows inherited tables to extend their base table data and behavior so that they inherit from their base table. For example, in **Error! Reference source not found.**, the SalesAgreementHeader and urchaseAgreementHeader tables that represent agreements in the **Sales and marketing** and **Procurement and sourcing** modules, respectively, are inherited from (extend) the same base table, which is named AgreementHeader.

Although a detailed explanation of the Microsoft Dynamics AX 2012 table inheritance feature is beyond the scope of this paper, some of the main features are as follows:

- Microsoft Dynamics AX 2012 table inheritance resembles the "classic" OOP inheritance concept. This concept enables child entities of a parent entity to *specialize,* or extend, of a common (shared) base property set that is inherited from the parent entity. Therefore, all properties of the AgreementHeader entity (for example, the Currency, AgreementState, and DefaultAgreementLineEffectiveDate columns) are applicable to both the SalesAgreementHeader and PurchaseAgreementHeader entities. However, the **CustAccount** property is applicable only to agreement headers in **Sales and marketing**, and is therefore defined only by the SalesAgreementHeader entity.
- Microsoft Dynamics AX 2012 table inheritance also enables the specialization of base table behavior*.* Therefore, every method of the base table can be locally overwritten directly in the inherited table object. (This capability resembles the implementation of virtual methods by child class in languages that support OOP.) This feature not only enables the base behavior to be changed in inherited objects, but also removes the need to maintain a class hierarchy behind a table to adjust the behavior of the table that depends on the type of a current record. (This pattern was widely used in previous versions of Microsoft Dynamics AX to support behavioral deviations for records of different types that were stored in a single table.)

The third major difference from the blanket order data model in Microsoft Dynamics AX 2009 is the implementation of a soft deletion pattern for several Microsoft Dynamics AX 2012 Agreement Framework entities. Soft deletion prevents the physical deletion, from the database, of records that represent entities that have been logically deleted by a Microsoft Dynamics AX user.

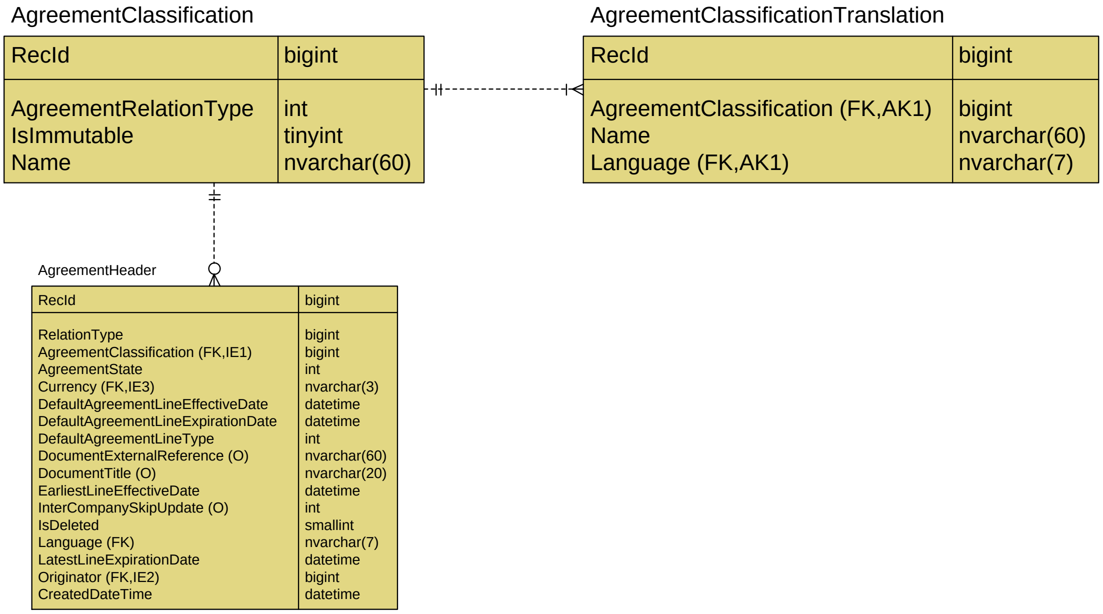
Usually, entities that implement a soft deletion pattern contain an **IsDeleted** property in their property (column) lists. For example, the AgreementHeader and AgreementLine entities both support the soft deletion functionality, and both contain the **IsDeleted** property. Therefore, if a Microsoft Dynamics AX 2012 user deletes a specific line from a particular agreement, the corresponding physical record is not actually deleted from a database. Instead, the line is just marked as deleted (that is, the **IsDeleted** field is set to **Yes**). As a result, the system can recognize records that have been deleted and, for example, omit them from forms that display all lines for the agreement.

When you query data directly from the database or design queries for your application, it is important to be aware of entities that implement the soft deletion functionality, and to include a proper condition for the **IsDeleted** field of a record. You can then adjust the record set that is returned by your query to suit your business needs.

Finally, note that the Agreement Framework in Microsoft Dynamics AX 2012 widely uses the classic Header-Lines pattern in exactly the same way as blanket orders used it in Microsoft Dynamics AX 2009. Therefore, several agreement lines in Microsoft Dynamics AX 2012 can be associated with a single agreement header. (Likewise, in Microsoft Dynamics AX 2009, several blanket order lines could be associated with a single blanket order header.) This is the only similarity between the new Agreement Framework physical data model in Microsoft Dynamics AX 2012 and the blanket order data model in Microsoft Dynamics AX 2009.

Agreement Framework physical data model: Agreement classification entities

Figure 1 illustrates the agreement classification and agreement classification translation entities of the Agreement Framework physical data model.



AgreementClassification

RecId	bigint
AgreementRelationType IsImmutable Name	int tinyint nvarchar(60)

AgreementClassificationTranslation

RecId	bigint

AgreementClassification (FK,AK1)
Name
Language (FK,AK1)
	bigint
	nvarchar(60)
	nvarchar(7)

AgreementHeader

RecId	bigint
RelationType AgreementClassification (FK,IE1) AgreementState Currency (FK,IE3) DefaultAgreementLineEffectiveDate DefaultAgreementLineExpirationDate DefaultAgreementLineType DocumentExternalReference (O) DocumentTitle (O) EarliestLineEffectiveDate InterCompanySkipUpdate (O) IsDeleted Language (FK) LatestLineExpirationDate Originator (FK,IE2) CreatedDateTime	bigint bigint int nvarchar(3) datetime datetime int nvarchar(60) nvarchar(20) datetime int smallint nvarchar(7) datetime bigint datetime

Figure 1 Agreement classification and agreement classification translation entities of the Agreement Framework physical data model

Anticipating a need among our customers and partners a more granular classification of agreements in the Agreement Framework, we introduced the concept of agreement classification in Microsoft Dynamics AX 2012. Agreement classifications provide another dimension, in addition to the AgreementHeader subtypes, that can be used to describe agreements.

As Figure 2 shows, every agreement in Microsoft Dynamics AX is associated with a corresponding agreement classification. This association is mandatory, and agreements of particular subtypes can be grouped by using various agreement classifications that are predefined in the system.

Therefore, a Microsoft Dynamics AX 2012 installation can be customized to incorporate several different types of sales agreements, based on your business needs, such as sales service agreements and sales general agreements.

At this time, the only property that an agreement classification entity has is **Name**. However, the properties can easily be extended through customizations. By associating properties with the AgreementClassificationTranslation entity, users of Agreement Framework can localize the agreement classification's **Name** property for the various languages that are used in the Microsoft Dynamics AX 2012 installation.

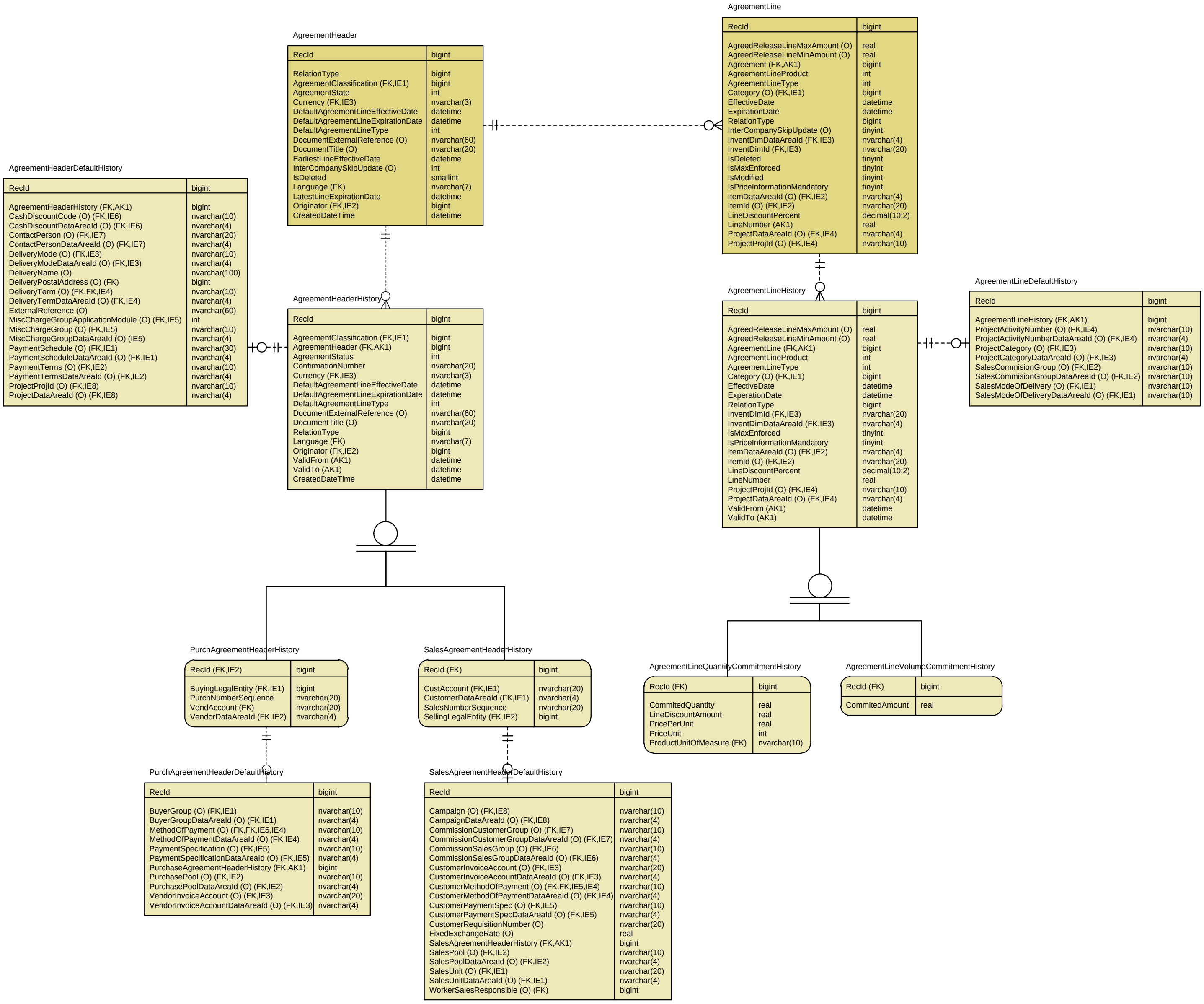
You can customize the Microsoft Dynamics AX 2012 Agreement Framework in many other ways by basing the customizations on the framework's agreement classification concept.

Agreement Framework physical data model: History entities

The Agreement Framework in Microsoft Dynamics AX 2012 enables users to create a snapshot of the current agreement state at any given time. This process is called confirmation of agreements. The system stores these snapshots, which are called the agreement history, as separate records in a database. Therefore, the persisted state for a particular agreement can be reconstructed by the system and accessed by the user for later analysis.

To enable this functionality, the Agreement Framework physical data model defines a number of history entities, which are shown in Figure 3.

Note: The name of every Agreement Framework history entity is composed of the name of the base entity and the suffix History. For example, for an AgreementHeader entity, the history snapshots available in the system are stored in a corresponding AgreementHeaderHistory table.



AgreementHeader

RecId	bigint
RelationType	bigint
AgreementClassification (FK,IE1)	bigint
AgreementState	int
Currency (FK,IE3)	nvarchar(3)
DefaultAgreementLineEffectiveDate	datetime
DefaultAgreementLineExpirationDate	datetime
DefaultAgreementLineType	int
DocumentExternalReference (O)	nvarchar(60)
DocumentTitle (O)	nvarchar(20)
EarliestLineEffectiveDate	datetime
InterCompanySkipUpdate (O)	int
IsDeleted	smallint
Language (FK)	nvarchar(7)
LatestLineExpirationDate	datetime
Originator (FK,IE2)	bigint
CreatedDateTime	datetime

AgreementLine

RecId	bigint
AgreedReleaseLineMaxAmount (O)	real
AgreedReleaseLineMinAmount (O)	real
Agreement (FK,AK1)	bigint
AgreementLineProduct	int
AgreementLineType	int
Category (O) (FK,IE1)	bigint
EffectiveDate	datetime
ExpirationDate	datetime
RelationType	bigint
InterCompanySkipUpdate (O)	tinyint
InventDimDataAreaId (FK,IE3)	nvarchar(4)
InventDimId (FK,IE3)	nvarchar(20)
IsDeleted	tinyint
IsMaxEnforced	tinyint
IsModified	tinyint
IsPriceInformationMandatory	tinyint
ItemDataAreaId (O) (FK,IE2)	nvarchar(4)
ItemId (O) (FK,IE2)	nvarchar(20)
LineDiscountPercent	decimal(10;2)
LineNumber (AK1)	real
ProjectDataAreaId (O) (FK,IE4)	nvarchar(4)
ProjectProjId (O) (FK,IE4)	nvarchar(10)

AgreementHeaderDefaultHistory

RecId	bigint
AgreementHeaderHistory (FK,AK1)	bigint
CashDiscountCode (O) (FK,IE6)	nvarchar(10)
CashDiscountDataAreaId (O) (FK,IE6)	nvarchar(4)
ContactPerson (O) (FK,IE7)	nvarchar(20)
ContactPersonDataAreaId (O) (FK,IE7)	nvarchar(4)
DeliveryMode (O) (FK,IE3)	nvarchar(10)
DeliveryModeDataAreaId (O) (FK,IE3)	nvarchar(4)
DeliveryName (O)	nvarchar(100)
DeliveryPostalAddress (O) (FK)	bigint
DeliveryTerm (O) (FK,FK,IE4)	nvarchar(10)
DeliveryTermDataAreaId (O) (FK,IE4)	nvarchar(4)
ExternalReference (O)	nvarchar(60)
MiscChargeGroupApplicationModule (O) (FK,IE5)	int
MiscChargeGroup (O) (FK,IE5)	nvarchar(10)
MiscChargeGroupDataAreaId (O) (IE5)	nvarchar(4)
PaymentSchedule (O) (FK,IE1)	nvarchar(30)
PaymentScheduleDataAreaId (O) (FK,IE1)	nvarchar(4)
PaymentTerms (O) (FK,IE2)	nvarchar(10)
PaymentTermsDataAreaId (O) (FK,IE2)	nvarchar(4)
ProjectProjId (O) (FK,IE8)	nvarchar(10)
ProjectDataAreaId (O) (FK,IE8)	nvarchar(4)

AgreementHeaderHistory

RecId	bigint
AgreementClassification (FK,IE1)	bigint
AgreementHeader (FK,AK1)	bigint
AgreementStatus	int
ConfirmationNumber	nvarchar(20)
Currency (FK,IE3)	nvarchar(3)
DefaultAgreementLineEffectiveDate	datetime
DefaultAgreementLineExpirationDate	datetime
DefaultAgreementLineType	int
DocumentExternalReference (O)	nvarchar(60)
DocumentTitle (O)	nvarchar(20)
RelationType	bigint
Language (FK)	nvarchar(7)
Originator (FK,IE2)	bigint
ValidFrom (AK1)	datetime
ValidTo (AK1)	datetime
CreatedDateTime	datetime

AgreementLineHistory

RecId	bigint
AgreedReleaseLineMaxAmount (O)	real
AgreedReleaseLineMinAmount (O)	real
AgreementLine (FK,AK1)	bigint
AgreementLineProduct	int
AgreementLineType	int
Category (O) (FK,IE1)	bigint
EffectiveDate	datetime
ExperationDate	datetime
RelationType	bigint
InventDimId (FK,IE3)	nvarchar(20)
InventDimDataAreaId (FK,IE3)	nvarchar(4)
IsMaxEnforced	tinyint
IsPriceInformationMandatory	tinyint
ItemDataAreaId (O) (FK,IE2)	nvarchar(4)
ItemId (O) (FK,IE2)	nvarchar(20)
LineDiscountPercent	decimal(10;2)
LineNumber	real
ProjectProjId (O) (FK,IE4)	nvarchar(10)
ProjectDataAreaId (O) (FK,IE4)	nvarchar(4)
ValidFrom (AK1)	datetime
ValidTo (AK1)	datetime

AgreementLineDefaultHistory

RecId	bigint
AgreementLineHistory (FK,AK1)	bigint
ProjectActivityNumber (O) (FK,IE4)	nvarchar(10)
ProjectActivityNumberDataAreaId (O) (FK,IE4)	nvarchar(4)
ProjectCategory (O) (FK,IE3)	nvarchar(10)
ProjectCategoryDataAreaId (O) (FK,IE3)	nvarchar(4)
SalesCommisionGroup (O) (FK,IE2)	nvarchar(10)
SalesCommisionGroupDataAreaId (O) (FK,IE2)	nvarchar(10)
SalesModeOfDelivery (O) (FK,IE1)	nvarchar(10)
SalesModeOfDeliveryDataAreaId (O) (FK,IE1)	nvarchar(10)

PurchAgreementHeaderHistory

RecId (FK,IE2)	bigint
BuyingLegalEntity (FK,IE1)	bigint
PurchNumberSequence	nvarchar(20)
VendAccount (FK)	nvarchar(20)
VendorDataAreaId (FK,IE2)	nvarchar(4)

SalesAgreementHeaderHistory

RecId (FK)	bigint
CustAccount (FK,IE1)	nvarchar(20)
CustomerDataAreaId (FK,IE1)	nvarchar(4)
SalesNumberSequence	nvarchar(20)
SellingLegalEntity (FK,IE2)	bigint

AgreementLineQuantityCommitmentHistory

RecId (FK)	bigint
CommitedQuantity	real
LineDiscountAmount	real
PricePerUnit	real
PriceUnit	int
ProductUnitOfMeasure (FK)	nvarchar(10)

AgreementLineVolumeCommitmentHistory

RecId (FK)	bigint
CommitedAmount	real

PurchAgreementHeaderDefaultHistory

RecId	bigint
BuyerGroup (O) (FK,IE1)	nvarchar(10)
BuyerGroupDataAreaId (O) (FK,IE1)	nvarchar(4)
MethodOfPayment (O) (FK,FK,IE5,IE4)	nvarchar(10)
MethodOfPaymentDataAreaId (O) (FK,IE4)	nvarchar(4)
PaymentSpecification (O) (FK,IE5)	nvarchar(10)
PaymentSpecificationDataAreaId (O) (FK,IE5)	nvarchar(4)
PurchaseAgreementHeaderHistory (FK,AK1)	bigint
PurchasePool (O) (FK,IE2)	nvarchar(10)
PurchasePoolDataAreaId (O) (FK,IE2)	nvarchar(4)
VendorInvoiceAccount (O) (FK,IE3)	nvarchar(20)
VendorInvoiceAccountDataAreaId (O) (FK,IE3)	nvarchar(4)

SalesAgreementHeaderDefaultHistory

RecId	bigint
Campaign (O) (FK,IE8)	nvarchar(10)
CampaignDataAreaId (O) (FK,IE8)	nvarchar(4)
CommissionCustomerGroup (O) (FK,IE7)	nvarchar(10)
CommissionCustomerGroupDataAreaId (O) (FK,IE7)	nvarchar(4)
CommissionSalesGroup (O) (FK,IE6)	nvarchar(10)
CommissionSalesGroupDataAreaId (O) (FK,IE6)	nvarchar(4)
CustomerInvoiceAccount (O) (FK,IE3)	nvarchar(20)
CustomerInvoiceAccountDataAreaId (O) (FK,IE3)	nvarchar(4)
CustomerMethodOfPayment (O) (FK,FK,IE5,IE4)	nvarchar(10)
CustomerMethodOfPaymentDataAreaId (O) (FK,IE4)	nvarchar(4)
CustomerPaymentSpec (O) (FK,IE5)	nvarchar(10)
CustomerPaymentSpecDataAreaId (O) (FK,IE5)	nvarchar(4)
CustomerRequisitionNumber (O)	nvarchar(20)
FixedExchangeRate (O)	real
SalesAgreementHeaderHistory (FK,AK1)	bigint
SalesPool (O) (FK,IE2)	nvarchar(10)
SalesPoolDataAreaId (O) (FK,IE2)	nvarchar(4)
SalesUnit (O) (FK,IE1)	nvarchar(20)
SalesUnitDataAreaId (O) (FK,IE1)	nvarchar(4)
WorkerSalesResponsible (O) (FK)	bigint

Figure 2 History entities of the Agreement Framework physical data model

There are several important points to note in connection with the history entities of the Agreement Framework physical data model.

First, explicit relations between history and base entities are established for only a few entities: AgreementHeader, AgreementHeaderHistory, AgreementLine, and AgreementLineHistory. All other similar relations were deemed redundant for the data model, and therefore were not implemented.

It is also important to note that, in spite of obvious similarities between the physical data model fragment shown in Figure 3 and the physical data model fragment for the Agreement Framework main entities shown in Figure 1, these two fragments are significantly different in the following way: History entities for agreement headers and lines do not directly implement the "Header-Lines" pattern. As the figures show, there is no relation between AgreementHeaderHistory and AgreementLineHistory entities.

This unusual implementation of history entities for agreement headers and lines is caused by the performance optimization that is applied to the Agreement Framework physical data model. Logically, each instance of the agreement header history contains several instances of the agreement line history. However, for performance reasons, AgreementLineHistory instances can be shared among several AgreementHeaderHistory instances. Therefore, the explicit relationship between the two entities cannot be established by the physical data model.

The following example illustrates why this sharing is important for performance. An agreement contains 100 lines, and between two consecutive snapshots that are taken for this agreement, only one line has been changed. If history records cannot be shared among several header history records (that is, between several snapshots), two identical sets of 99 agreement line history records for unchanged lines must be stored (one set per snapshot). This behavior would significantly affect the performance of the system.

Instead, when a second snapshot is taken for an agreement that contains only one changed line, only one instance of the agreement history line is stored. This instance represents the changed line. The other 99 records are not duplicated, but they are specifically marked to indicate that they now belong to several snapshots (history headers). This behavior is enabled by several properties that are defined by both main and history entities. In this example, the AgreementLine entity contains an **isModified** property, which is used to mark agreement lines that have been modified since the last agreement confirmation was performed by a user. Additionally, the AgreementHeaderHistory and AgreementLineHistory entities contain the **IsModified**, **ValidFrom**, and **ValidTo** properties, which are used to define the validity period for the history records. These properties are controlled by Agreement Framework, which sets or modifies the properties every time history records are created (at the moment the user confirms the agreement in the system).

Every time the state for a particular version that is persisted as a snapshot for an agreement must be reconstructed by the system, the Agreement Framework analyzes the interval between the **ValidFrom** and **ValidTo** timestamps of the AgreementHeaderHistory record that corresponds to the required version of a selected agreement. It then selects only the AgreementLineHistory records that belong to this header, and that were valid during the whole validity period of the header version. As a result, the Agreement Framework can accurately reconstruct and present to the user any history version of an agreement, but only has to persist a minimal data set in the database.

Agreement Framework physical data model: Release relation entities

Another major reason for the creation of the Agreement Framework was the need to represent contractual obligations for sales and purchases in the system. Similarly to blanket orders in Microsoft Dynamics AX 2009, a major part of Microsoft Dynamics AX 2012 Agreement Framework functionality deals with the creation of release orders when sales orders and purchase orders are "released from an agreement." To facilitate the release of orders from agreements, and to provide follow-up information (agreement fulfillment numbers) for agreements that are registered in the system, the Agreement Framework physical data model defines the following release relation entities, as shown in Figure 4: AgreementReleaseHeaderMatch, AgreementLineReleasedLine, and its "history shadow," AgreementLineReleasedLineHistory.

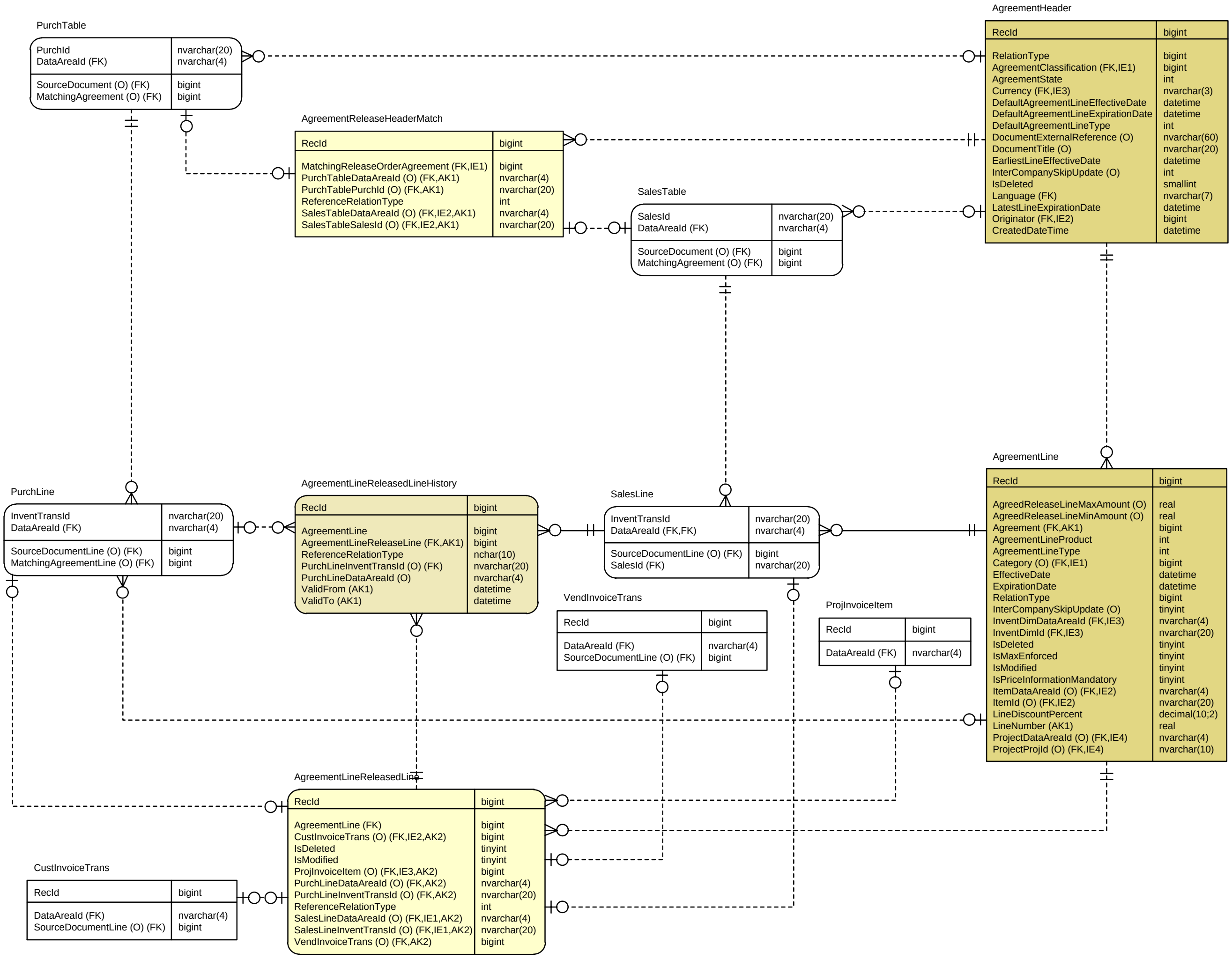

PurchTable
PurchId nvarchar(20)
DataAreaId (FK) nvarchar(4)
SourceDocument (O) (FK) bigint
MatchingAgreement (O) (FK) bigint
AgreementReleaseHeaderMatch
RecId bigint
MatchingReleaseOrderAgreement (FK,IE1) bigint
PurchTableDataAreaId (O) (FK,AK1) nvarchar(4)
PurchTablePurchId (O) (FK,AK1) nvarchar(20)
ReferenceRelationType int
SalesTableDataAreaId (O) (FK,IE2,AK1) nvarchar(4)
SalesTableSalesId (O) (FK,IE2,AK1) nvarchar(20)
SalesTable
SalesId nvarchar(20)
DataAreaId (FK) nvarchar(4)
SourceDocument (O) (FK) bigint
MatchingAgreement (O) (FK) bigint
AgreementHeader
RecId bigint
RelationType bigint
AgreementClassification (FK,IE1) bigint
AgreementState int
Currency (FK,IE3) nvarchar(3)
DefaultAgreementLineEffectiveDate datetime
DefaultAgreementLineExpirationDate datetime
DefaultAgreementLineType int
DocumentExternalReference (O) nvarchar(60)
DocumentTitle (O) nvarchar(20)
EarliestLineEffectiveDate datetime
InterCompanySkipUpdate (O) int
IsDeleted smallint
Language (FK) nvarchar(7)
LatestLineExpirationDate datetime
Originator (FK,IE2) bigint
CreatedDateTime datetime
PurchLine
InventTransId nvarchar(20)
DataAreaId (FK) nvarchar(4)
SourceDocumentLine (O) (FK) bigint
MatchingAgreementLine (O) (FK) bigint
AgreementLineReleasedLineHistory
RecId bigint
AgreementLine bigint
AgreementLineReleaseLine (FK,AK1) bigint
ReferenceRelationType nchar(10)
PurchLineInventTransId (O) (FK) nvarchar(20)
PurchLineDataAreaId (O) nvarchar(4)
ValidFrom (AK1) datetime
ValidTo (AK1) datetime
SalesLine
InventTransId nvarchar(20)
DataAreaId (FK,FK) nvarchar(4)
SourceDocumentLine (O) (FK) bigint
SalesId (FK) nvarchar(20)
VendInvoiceTrans
RecId bigint
DataAreaId (FK) nvarchar(4)
SourceDocumentLine (O) (FK) bigint
ProjInvoiceItem
RecId bigint
DataAreaId (FK) nvarchar(4)
AgreementLine
RecId bigint
AgreedReleaseLineMaxAmount (O) real
AgreedReleaseLineMinAmount (O) real
Agreement (FK,AK1) bigint
AgreementLineProduct int
AgreementLineType int
Category (O) (FK,IE1) bigint
EffectiveDate datetime
ExpirationDate datetime
RelationType bigint
InterCompanySkipUpdate (O) tinyint
InventDimDataAreaId (FK,IE3) nvarchar(4)
InventDimId (FK,IE3) nvarchar(20)
IsDeleted tinyint
IsMaxEnforced tinyint
IsModified tinyint
IsPriceInformationMandatory tinyint
ItemDataAreaId (O) (FK,IE2) nvarchar(4)
ItemId (O) (FK,IE2) nvarchar(20)
LineDiscountPercent decimal(10;2)
LineNumber (AK1) real
ProjectDataAreaId (O) (FK,IE4) nvarchar(4)
ProjectProjId (O) (FK,IE4) nvarchar(10)
AgreementLineReleasedLine
RecId bigint
AgreementLine (FK) bigint
CustInvoiceTrans (O) (FK,IE2,AK2) bigint
IsDeleted tinyint
IsModified tinyint
ProjInvoiceItem (O) (FK,IE3,AK2) bigint
PurchLineDataAreaId (O) (FK,AK2) nvarchar(4)
PurchLineInventTransId (O) (FK,AK2) nvarchar(20)
ReferenceRelationType int
SalesLineDataAreaId (O) (FK,IE1,AK2) nvarchar(4)
SalesLineInventTransId (O) (FK,IE1,AK2) nvarchar(20)
VendInvoiceTrans (O) (FK,AK2) bigint
CustInvoiceTrans
RecId bigint
DataAreaId (FK) nvarchar(4)
SourceDocumentLine (O) (FK) bigint

Figure 3 Release relation entities of the Agreement Framework physical data model

When a sales or purchase order contains at least one line that is released from an agreement, the order is referred as a release order. For each release order, a corresponding record exists in the AgreementReleaseHeaderMatch. This record defines the explicit relation between the release order header (SalesTable or PurchTable record) and the AgreementHeader in the system.

A similar approach is used to establish the connection between the release line and agreement line: For each line of the release order that is released from an agreement (release line), Agreement

Framework creates a record in the AgreementLineReleasedLine table. This record explicitly establishes a relationship between the release line (SalesLine or PurchLine record) and the corresponding AgreementLine.

Note: Release orders in Microsoft Dynamics AX 2012 can mix general and release lines, but all release lines for one release order must relate to the agreement lines of the same agreement. In the other words, a single release order cannot contain release lines that originate from different agreements.

In contrast to the release from blanket order process in Microsoft Dynamics ACX 2009, the relationship between the release line and the agreement line in Agreement Framework in Microsoft Dynamics AX 2012 is not permanent. Some modifications that can be made to the release line may violate conditions of the corresponding agreement line. However, the user can overwrite the agreement terms (for example, the released quantity, price, or discount) and drop the relationship between the release line and the agreement line. As a result, the release line is no longer covered by the agreement conditions. In this case, the line is no longer referred as a release line.

Dropped relations (records in the AgreementLineReleasedLine table) are not physically deleted from a database. Instead, the lines are marked as deleted (soft deletion pattern). Agreement Framework uses this approach, because it facilitates proper history reconstruction for any agreement in the system.

To support all business scenarios, Agreement Framework has to track release relations at a line level for the following journals: VendInvoiceTrans, CustInvoiceTrans, and ProjInvoiceItem. Every time a record in these journals is created from the release line, a corresponding instance of the AgreementLineReleasedLine is created to define the relationship between the created journal record and the agreement line.

By searching through release relation records for a particular agreement line, the Agreement Framework can calculate fulfillment indicators such as remaining quantity, released quantity, delivered quantity, and invoiced quantity for a particular agreement line. As a result, the Agreement Framework can dynamically provide information about contract execution. To avoid performance bottlenecks during mass processing of orders, the Agreement Framework does not store these indicators anywhere in the system, but calculates them every time the user wants to access them.

Agreement Framework physical entities: Customization entry points

As previously described, the Agreement Framework uses the Microsoft Dynamics AX 2012 table inheritance functionality. As a result, the Agreement Framework does not have to create and maintain a class hierarchy behind the table structure to support the behavioral deviations in table functionality for the various data subtypes that it uses. Therefore, before they customize the Agreement Framework functionality that Microsoft Dynamics AX provides out-of-the-box, developers should consider applying both their data adjustments and data behavior modifications directly to the Microsoft Dynamics AX table objects. The following table summarizes the new physical entities that have been introduced by the Agreement Framework.

Entity (table) name	General description	Data customization suggestions	Behavior customization suggestions
AgreementClassification	A table that defines all known agreement classifications in Microsoft Dynamics AX 2012.	All extra properties to the AgreementClassification entity in a customized solution should be included in this table.	All extra behaviors for the AgreementClassification entity that are required by the customized solution should be implemented by this table.
AgreementClassificationTranslation	A table that defines all known translations for agreement classifications that are implemented by the system.	All extra properties that are specific to a customized version of the AgreementClassificationTranslation entity should be added to this table.	If behavior for AgreementClassificationTranslation entity is added by the customized solution, the corresponding methods must be added to this table.
AgreementHeader	A base table (root of a table inheritance tree) that contains all the properties that are common to sale and purchase agreement headers.	If a customized solution implements properties that are applicable to all AgreementHeader subtypes, these properties should be included in this table.	All customizations to existing functionality, and new "virtual" methods that are applicable to all AgreementHeader subtypes that are defined by customized solutions should be placed in this table.
SalesAgreementHeader	A derived table that extends the AgreementHeader entity. It contains all the fields that are specific to sales agreement headers.	If a customized solution has to introduce new properties for the headers of its sales agreements, these new properties should be included in this table.	All customizations to the behavior of the agreement headers in the Sales and marketing module should be implemented in this table.
PurchAgreementHeader	A derived table that extends the AgreementHeader entity. It contains all the fields that are specific to purchase agreement headers.	If a customized solution has to introduce new properties for the headers of its purchase agreements, these new properties should be included in this table	All customizations to the behavior of the agreement headers in the Procurement and sourcing module should be implemented in this table.

Entity (table) name	General description	Data customization suggestions	Behavior customization suggestions
AgreementHeaderDefault	A table that contains release order defaulting policies (fields that are used for release order header initialization) that are common to all subtypes of release orders.	If a customized solution adds new fields to both the sales order and purchase order headers, and it relies on Agreement Framework for initialization of these new fields during the release from agreement process, similar header fields should a be added to this table.	All release order header defaulting policies that support the functionality that is required for the customization should be placed in this table[1].
SalesAgreementHeaderDefault	A table that contains sales release order defaulting policies (fields that are used for sales release order header initialization).	If a sales order header is customized with new sales order–specific fields, and these fields should be initialized when a sales order is created as a result of the release from a sales agreement, similar header fields should be added to this defaulting table.	All sales release order header–specific defaulting policies that support functionality that is required for the customization should be placed in this table.
PurchAgreementHeaderDefault	A table that contains purchase release order defaulting policies (fields that are used for purchase release order header initialization).	If a purchase order header is customized with new purchase order–specific fields, and these fields should be initialized when a purchase order is created as a result of the release from a purchase agreement, similar header fields should be added to this defaulting table.	All purchase release order header–specific defaulting policies that support the functionality that is required for the customization should be placed in this table.
AgreementLine	The base table (root of a table inheritance tree) that contains all the properties that are common to all subtypes of the agreement lines that are defined in the system.	If a customized solution implements properties that are applicable to all AgreementLine subtypes, these properties should be included in this table.	All customizations to existing functionality, and new "virtual" methods that are applicable to all AgreementLine subtypes that are defined by a customized solution should be placed in this table.

[1] For performance reasons, the Agreement Framework does not store records for defaulting entities (tables that have the suffix Default in their name) unless at least one record field has been assigned a non-default value. Therefore, after you customize the data structure, it is important to adjust the functionality that checks whether a record contains a non-default value according to the new record layout. In the other words, all new properties that are added to the defaulting entities should be included in the logic of methods, such as the **checkDefaultValues**() method of the AgreementHeaderDefault table.

Entity (table) name	General description	Data customization suggestions	Behavior customization suggestions
AgreementLineQuantityCommitment	A derived table that extends the AgreementLine entity and that contains all the fields that are specific to quantity-based commitments (agreement lines).	If a customized solution has to introduce new properties for quantity-based agreement lines, these new properties should be included in this table.	All customizations to the behavior of quantity-based commitment lines should be implemented in this table.
AgreementLineVolumeCommitment	A derived table that extends the AgreementLine entity and that contains all the fields that are specific to volume-based commitments (agreement lines).	If a customized solution has to introduce new properties for volume-based agreement lines, these new properties should be included in this table.	All customizations to the behavior of volume-based commitment lines should be implemented in this table.
AgreementLineDefault	A table that contains optional release order line defaulting policies (non-mandatory fields that are used for release order line initialization).	If a customized solution adds new fields to both sales order or purchase order lines, and relies on Agreement Framework for initialization of these new fields during the release from agreement process, similar order line fields should a be added to this table.	All release order line defaulting policies that support the functionality that is required for the customization should be placed in this table.
AgreementReleaseHeaderMatch	A table that contains records that describe the relations between agreement and release order headers.	If a customized solution extends multiple entities that can be matched with an agreement header (in addition to sales order and purchase order headers), corresponding fields that describe these relations should be added to this table.	All required customizations to the behavior of the relations between agreement and release order headers should be implemented by this table.

Entity (table) name	General description	Data customization suggestions	Behavior customization suggestions
AgreementLineReleasedLine	A table that contains records that describe the relations between agreement and release order lines.	If a customized solution extends multiple entities that can be associated with agreement lines (in addition to sales order and purchase order lines, and journal lines that are stored in the following Microsoft Dynamics AX 2012 tables: CustInvoiceTrans, VendInvoiceTrans, and ProjInvoiceItem), corresponding fields that describe these relations should be added to this table.	All required customizations to the logic that is uses to establish or drop associations between agreement lines and release order (journal) lines should be implemented by this table.
AgreementHeaderHistory	A base table (root of a table inheritance tree) that contains all the fields that are used to create a snapshot (historical state) for AgreementHeader records. Fields that are stored by this table are common to sales and purchase agreement history headers.	All modifications to the data composition that are made for the AgreementHeader table should be reflected in the customization of this table.	All customizations to the implemented behavior of the AgreementHeaderHistory entity, and new "virtual" methods that are applicable to all AgreementHeaderHistory subtypes that are defined by the customized solution should be placed in this table.
SalesAgreementHeaderHistory	A derived table that extends the AgreementHeaderHistory entity. It contains all the SalesAgreementHeader-specific fields that are used to create a snapshot (historical state) for a SalesAgreementHeader record.	If a customized solution introduces new properties for the headers of its sales agreements, corresponding new properties should be included in this table.	All customizations to the existing functionality of the SalesAgreementHeaderHistory entity, and all required subtype-specific implementations for new "virtual" methods that are introduced by customization of the AgreementHeaderHistory entity, should be implemented by this table.

Entity (table) name	General description	Data customization suggestions	Behavior customization suggestions
PurchAgreementHeaderHistory	A derived table that extends the AgreementHeaderHistory entity. It contains all the properties that are required to recreate the historical state of a particular record in the PurchAgreementHeader table.	If a customized solution introduces new properties for the headers of purchase agreements, corresponding new properties should be included in this table.	All customizations to the existing functionality of the PurchAgreementHeaderHistory entity, and all required subtype-specific implementations for new "virtual" methods that are introduced by customization of the AgreementHeaderHistory entity should be implemented by this table.
AgreementHeaderDefaultHistory	A table that contains the fields that comprise a snapshot of the release order defaulting policies (fields that are common to both sales order and purchase release order headers that are initialized by Agreement Framework during the release from agreement process).	Any customization that is introduced for the AgreementHeaderDefault entity should be reflected by corresponding changes to the column set of this table.	Entity-specific getters/setters and, if required, other behavioral methods that are common to all AgreementHeaderHistory subtypes should be implemented in this table.
SalesAgreementHeaderDefaultHistory	A table that contains the fields that comprise a snapshot of the sales release order defaulting policies (fields that are specific to sales release order headers that are initialized by Agreement Framework during the release from agreement process).	Any customization that is introduced for the SalesAgreementHeaderDefault entity should be reflected by corresponding changes to the column set of this table.	Entity-specific getters/setters and, if required, other behavioral methods that are specific to the SalesAgreementHeaderHistory subtype should be implemented in this table.
PurchAgreementHeaderDefaultHistory	A table that contains the fields that comprise a snapshot of the purchase release order defaulting policies (fields that are specific to the purchase release order headers that are initialized by Agreement Framework during the release from agreement process).	Any customization that is introduced for the PurchaseAgreementHeaderDefault entity should be reflected by corresponding changes to the column set of this table.	Entity-specific getters/setters and, if required, other behavioral methods that are specific to the PurchaseAgreementHeaderHistory subtype should be implemented in this table.

Entity (table) name	General description	Data customization suggestions	Behavior customization suggestions
AgreementLineHistory	A base table (root of a table inheritance tree) that contains all the fields that are used to create a snapshot (historical state) for AgreementLine records. Fields that are stored by this table are common to all AgreementLineHistory subtypes that are defined in the system.	If a customized solution implements properties that are applicable to all AgreementLine subtypes, these properties should be included in this table.	All customizations to existing functionality, and new "virtual" methods that are applicable to all AgreementLineHistory subtypes that are defined by a customized solution should be placed in this table.
AgreementLineQuantityCommitmentHistory	A derived table that extends the AgreementLineHistory entity. It contains all the quantity commitment–specific fields that are used to create a snapshot (historical state) for AgreementLineQuantityCommitment records.	If a customized solution has to introduce new properties for quantity-based agreement lines, corresponding new properties should be included in this table.	All customizations to the behavior of quantity-based commitment history lines should be implemented in this entity.
AgreementLineVolumeCommitmentHistory	A derived table that extends the AgreementLineHistory entity. It contains all volume commitment–specific fields that are used to create a snapshot (historical state) for AgreementLineVolumeCommitment records.	If a customized solution has to introduce new properties for volume-based agreement lines, corresponding new properties should be included in this table.	All customizations to the behavior of volume-based commitment history lines should be implemented in this entity.
AgreementLineDefaultHistory	A table that contains the fields that comprise a snapshot of the release order lines defaulting policies (optional fields that are common to both sales order and purchase release order lines that are initialized by Agreement Framework during the release from agreement process).	All new properties that are introduced by the customization of the AgreementLineDefault entity should have corresponding fields in this table to facilitate proper snapshot creation and further reconstruction.	All new behavior that customizations require from the AgreementLineDefaultHistory entity must be implemented by the methods of this table.

Entity (table) name	General description	Data customization suggestions	Behavior customization suggestions
AgreementLineReleasedLineHistory	A table that contains fields that are used to create a snapshot (historical state) of records that describe the relations between agreement order and release order lines.	If a customized solution introduces new data fields for the AgreementLineReleasedLine entity, corresponding changes should be propagated to the field set of this table to facilitate proper snapshot creation and further state reconstruction of AgreementReleaseLineRelation records.	All required customizations to the logic for historical snapshots for AgreementLineReleasedLine records should be implemented by this entity. For example, variations of the find() and initFromXXX() methods, an entity's getter/setter set, and so on.

Agreement Framework UI customization references

Numerous forms have been developed for the Agreement Framework, to facilitate an efficiently balanced and comprehensive visual user interface (UI) for the Agreement Framework physical data and functionality.

Agreement Framework form name	General description	Comments
AgreementClassification	A simple list page that provides a graphical interface for the creation, update, and deletion of agreement classifications in the system.	
AgreementClassificationTranslation	A simple list page that can be used to manage translations for agreement classifications that are defined in the system.	
AgreementConfirmRunForm	A form that is used for agreement confirmation functionality.	
AgreementLine	A card-style form that displays information about agreement lines that are associated with release order lines.	
AgreementLinePrompt	A form that facilitates the process of creating an association between release order lines and agreement lines when the release from the agreement is initiated from a sales order or purchase order.	

Agreement Framework form name	General description	Comments
PurchAgreement	A transaction details form that provides functionality and interfaces for purchase agreements.	This form's functionality (especially the logic that enables and disables menu commands in the form) relies extensively on the service that is provided by the PurchAgreementForm class. Therefore, be sure to align your customizations to this form's functionality with patterns and performance optimizations that are implemented by the PurchAgreementForm class.
PurchAgreementGenerateReleaseOrder	A form that is used to create a release purchase order.	If additional customizations to the release from purchase agreement process are implemented, consider customizing the following: This form The underlying TmpPurchLine temporary table The PurchAgreementGenerateReleaseOrder service class
PurchAgreementHistory	A simple list details form that provides the whole range of functionality for the purchase agreement history (snapshots).	
PurchAgreementInvoiceJournal	A simple list page that displays invoice journal lines that are associated with purchase agreement lines.	
PurchAgreementListPage	A list page that provides a UI for managing purchase agreements. This form is an entry point to the purchase agreement–related functionality cluster that is provided by Microsoft Dynamics AX 2012.	When customizing purchase agreements, consider placing your customizations in the following locations: • This form • The corresponding LP interaction class, PurchAgreementListPageInteraction
PurchCreateOrderFromVendorWithAgreement	A form that facilitates the creation of a purchase order directly from a vendor form that displays the purchase agreements that have been created in the system.	

Agreement Framework form name	General description	Comments
SalesAgreement	A transaction details form that provides sales agreement–related functionality and interfaces.	This form's functionality (especially the logic that enables and disables menu commands in the form) relies extensively on services that are provided by the SalesAgreementForm class. Therefore, be sure to align your customizations to this form's functionality with patterns and performance optimization approaches that are implemented by the SalesAgreementForm class.
SalesAgreementGenerateReleaseOrder	A form that is used to create the release sales order.	If additional customizations to the release from sales agreement process are implemented, consider customizing the following: • This form • The underlying SalesCreatetReleaseOrderLineTmp temporary table • The underlying SalesCreateReleaseOrderTableTmp temporary table • The SalesAgreementGenerateReleaseOrder service class
SalesAgreementHistory	A simple list details form that provides the whole range of functionality for the sales agreement history (snapshots).	
SalesAgreementInvoiceJournal	A simple list page that displays invoice journal lines that are associated with sales agreement lines.	
SalesAgreementListPage	A list page that provides a UI for sales agreements. This form is an entry point to the sales agreement–related functionality cluster that is provided by Microsoft Dynamics AX 2012.	When customizing sales agreements, consider making your customizations in the following locations: • This form • The corresponding LP interaction class, SalesAgreementListPageInteraction
SalesCreateOrderFromCustomerWthAgreement	A form that facilitates the creation of a sales order directly from a customer form that displays the sales agreements that have been created in the system.	

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.