Microsoft Dynamics® AX 2012

Implementing InventTrans Refactoring for Microsoft Dynamics AX 2012 Applications

White Paper

This document discusses the refactoring of the InventTrans table and common design patterns. It also discusses how to implement these changes when developing new features or updating existing features.

http://microsoft.com/dynamics/ax

Date: January, 2011

Author: Kim Moerup, Senior Developer, Inventory Management

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

Table of Contents

Overview **3**

Audience **3**

Document purpose **3**

Data model changes **4**

Fields deleted from the InventTrans table 5

Developing new features 6

Upgrading existing features 6

Design patterns **6**

Relationships between inventory transaction originating tables and the InventTrans table 6

Modifying custom code 7

The Direction field 7

The TransType and TransRefId fields 7

The ItemRouteId and ItemBomId fields 7

Conclusion **8**

Overview

One of the largest tables in Microsoft Dynamics® AX has always been the InventTrans table. Over time, this table has kept growing because of too much redundant data. Microsoft Dynamics AX 2012 begins to change this trend by starting the process of refactoring parts of the table.

The refactoring has the following purposes:

- To reduce the amount of data stored (disk space)
- To refactor parts of the table to avoid redundant data and the inherent risk of inconsistent data

In previous versions of Microsoft Dynamics AX, fields named InventTransId were used to identify sets of inventory transactions in the InventTrans table and to identify the origin of inventory transactions in originating tables, such as the SalesLine, PurchLine, ProdTable, or InventJournalTrans table. Despite its name, InventTransId was not the ID for inventory transactions. The best way to describe the field is that it identified the set of inventory transactions related to a given inventory transaction originating record.

In Microsoft Dynamics AX 2012, we introduce a new table named InventTransOrigin for which RecId is the primary key, but for which InventTransId is the alternate or natural key. The new table stores data that is common to all the transactions related to the same inventory transaction originating record.

Audience

This document is intended for developers building new applications for Microsoft Dynamics AX 2012, as well as for developers updating their existing application code and data.

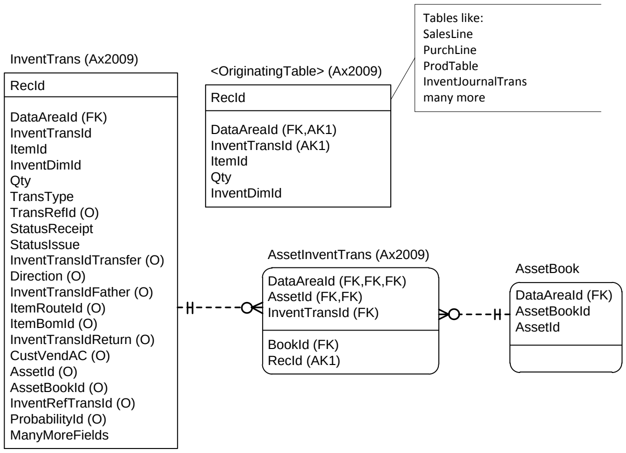
Document purpose

This document highlights the new design pattern used to implement relationships between the InventTrans table and inventory transaction originating tables. When detailing the new pattern, this document also describes the existing pattern that is being replaced and how developers should approach refactoring their existing code.

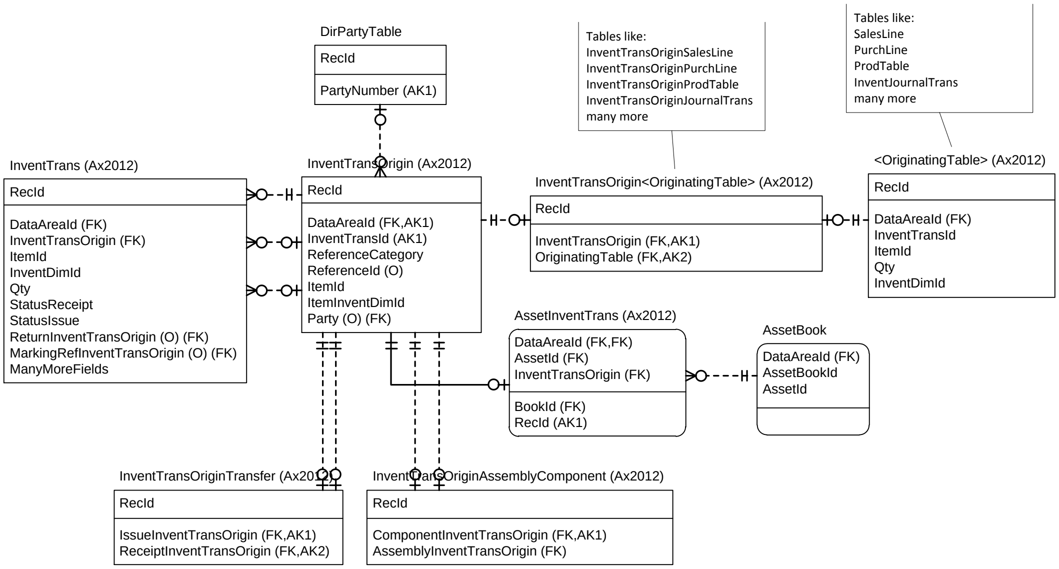
Data model changes

Microsoft Dynamics AX 2012 changed the data model of the **InventTrans** table as follows.

Microsoft Dynamics AX 2009 physical data model



Microsoft Dynamics AX 2012 physical data model


DirPartyTable
RecId
PartyNumber (AK1)
Tables like:
InventTransOriginSalesLine
InventTransOriginPurchLine
InventTransOriginProdTable
InventTransOriginJournalTrans
many more
Tables like:
SalesLine
PurchLine
ProdTable
InventJournalTrans
many more
InventTrans (Ax2012)
RecId
DataAreaId (FK)
InventTransOrigin (FK)
ItemId
InventDimId
Qty
StatusReceipt
StatusIssue
ReturnInventTransOrigin (O) (FK)
MarkingRefInventTransOrigin (O) (FK)
ManyMoreFields
InventTransOrigin (Ax2012)
RecId
DataAreaId (FK,AK1)
InventTransId (AK1)
ReferenceCategory
ReferenceId (O)
ItemId
ItemInventDimId
Party (O) (FK)
InventTransOrigin<OriginatingTable> (Ax2012)
RecId
InventTransOrigin (FK,AK1)
OriginatingTable (FK,AK2)
<OriginatingTable> (Ax2012)
RecId
DataAreaId (FK)
InventTransId
ItemId
Qty
InventDimId
AssetInventTrans (Ax2012)
DataAreaId (FK,FK)
AssetId (FK)
InventTransOrigin (FK)
BookId (FK)
RecId (AK1)
AssetBook
DataAreaId (FK)
AssetBookId
AssetId
InventTransOriginTransfer (Ax2012)
RecId
IssueInventTransOrigin (FK,AK1)
ReceiptInventTransOrigin (FK,AK2)
InventTransOriginAssemblyComponent (Ax2012)
RecId
ComponentInventTransOrigin (FK,AK1)
AssemblyInventTransOrigin (FK)

Fields deleted from the InventTrans table

The following table lists all the fields deleted from the InventTrans table. The table also specifies the tables to which some fields have been moved.

InventTrans field	Refactoring
InventTransId	This field has been moved to the new InventTransOrigin table.
TransType	This field has been moved to the new InventTransOrigin table and is renamed ReferenceCategory.
TransRefId	This field has been moved to the new InventTransOrigin table and is renamed ReferenceId.
InventTransIdTransfer	The functionality of this bidirectional reference is now modeled in the new InventTransOriginTransfer table, which identifies the issue and receipt records of the InventTransOrigin table that originated the transfer. For a transfer, there are a set of transactions related to the issuing of items and another set of transactions related to the receipt of the items. This new table identifies these two sets of transactions related to a single transfer. The data model now enforces that all transactions related to an inventory transfer are part of the same transfer record. In Microsoft Dynamics AX 2009 this was also the case, but it was only enforced by code and not by the data model.
Direction	This field was redundant (can be deduced from the Qty field) and has been removed.
InventTransIdFather	The functionality of this reference is now modeled in the new InventTransOriginAssemblyComponent table, which establishes the relationship between the final assembly and its components. The data model now enforces that all transactions related to an originating record are part of the same assembly. In Microsoft Dynamics AX 2009, this was also the case, but it was only enforced by code and not by the data model itself.
ItemRouteId	This field has been deleted.
ItemBomId	This field has been deleted.
InventTransIdReturn	This field has been replaced by the new ReturnInventTransOrigin field, which has been added to the InventTrans table. The functionality remains unchanged.
CustVendAC	This field has been replaced by the Party field in the new InventTransOrigin table. The field has a relationship to the DirParty table.
AssetId	This field has been removed because the table AssetInventTrans already exists and its InventTransId field has been replaced with a reference to the new InventTransOrigin table.
AssetBookId	This field has been removed because the tablee AssetInventTrans already exists and its InventTransId field has been replaced with a reference to the new InventTransOrigin table.
InventRefTransId	This field has been replaced by the new MarkingRefInventTransOrigin field, which has been added to the InventTrans table. The functionality remains unchanged.
ProbabilityId	This field has been deleted.

Developing new features

Developers developing new features for Microsoft Dynamics AX 2012 that references the **InventTrans** table should examine the following sections of this document for changes.

- Design Patterns
- Data Model Changes

Upgrading existing features

Developers upgrading existing features should first attempt to identify all the references to the defined design patterns and then follow the instructions in the related sections to upgrade their features. The upgrade can be done in any sequence, but the following steps are required:

- Identify the pattern that your feature uses today.
- Add the fields from the new data model to your data model. Prefix the deleted existing InventTransId fields with "DEL_."
- Create a data upgrade script to populate the new fields from the old fields. Note that parts of the data upgrade (validation parts) are done on the source system (the old version being upgraded from). This is done to shorten the downtime during the upgrade.
- Update the user interface to use the new controls appropriate for the pattern defined. The new controls will leverage the new foreign keys that you added to your data model.
- Update the references and business logic in your X++ classes, table methods, and all other code to use the new design patterns defined in the Design patterns section of this document.

Design patterns

This section discusses how to implement refactoring for Microsoft Dynamics AX 2012. If you need to refer to the physical model for the tables, see the Data model changes earlier in this document.

Relationships between inventory transaction originating tables and the InventTrans table

The split of the InventTrans table affects code that uses the relationships between the inventory transaction originating tables and the inventory transaction table. The InventTransId field has been removed from the InventTrans table and has been replaced by the InventTransOrigin field to reference the InventTransOrigin table. Some of the inventory transaction originating tables retain the InventTransId field, but the field cannot be used to reference the InventTrans table. All inventory transaction originating records must first be referenced by a record in the InventTransOrigin table that references the InventTrans table. The following code example shows how to do this.

```
while select inventTransOriginSalesLine
    where  inventTransOriginSalesLine.SalesLineDataAreaId == salesLine.dataAreaId
        && inventTransOriginSalesLine.InventTransId       == salesLine.InventTransId
    join inventTrans
    where  inventTrans.InventTransOrigin == inventTransOriginSalesLine.InventTransOrigin
{
    info(inventTrans.toString());
}
```

This change also affects other objects that used the InventTransId field to find, collect, or summarize data from the InventTrans table.

Existing tables such as SalesLine still have an InventTransId field. However, we strongly recommend that you use the InventTrans <OriginatingTable> (InventTransSalesLine) table to reference the

InventTransOrigin table instead of using the InventTransId field of the <OrginatingTable> (SalesLine) table.Eventually we will be able to remove the InventTransId fields from the inventory transaction originating tables, and if you rely on those InventTransId fields, you will have to update your code again.

Modifying custom code

To modify your custom code, you must find all the places where your code uses the InventTransId field. For each occurrence, you must determine whether the usage refers to the inventory transaction originating table itself or to the inventory transactions. If the InventTransId field is being used to refer to the InventTrans table, the code must be changed to first retrieve the InventTransOriginId for the inventory transaction originating record and then to use that InventTransOriginId to reference the InventTrans table records. You can do this in one of two ways:

- By joining through the appropriate InventTransOrigin<OriginatingTable> table.

  –or–

- By retrieving the InventTransOriginId from the appropriate InventTransOrigin<OriginatingTable> table by making the following call.

  ```
  InventTransOrigin<OriginatingTable>::findInventTransOriginId(…)
  ```

The Direction field

The Direction field has been removed. To replace it, a new display method (also called direction) has been added, that you can use in most cases. If your code requires that you have an actual field representing the direction, you might be able to use the code in the init method on the marking form (InventMarking). This form uses a new table named InventTransDirection that holds exactly two records—one for each direction value that can be joined to the InventTrans table.

The TransType and TransRefId fields

In previous versions of Microsoft Dynamics AX, relationships were defined between the InventTrans table and the various inventory transaction originating tables. The relationships were defined by using the TransType and TransRefId fields. However, these relationships do not enforce data integrity, and the fixed-field part of the TransType field cannot be translated into terms understood by BI tools.

In Microsoft Dynamics AX 2012, these two fields have been moved to the InventTransOrigin table and have been renamed ReferenceCategory and ReferenceId. However, the relationships have intentionally not been moved. We consider them "for information only" fields, and therefore they should not be considered as part of any relationship. For example, you can delete an inventory journal after it has been posted. In that case, the transactions still hold the original journal ID. But because the journal has been deleted, the transactions and the journal do not constitute a relationship.

In forms showing inventory transactions, it should still be possible to navigate to the inventory transaction originating forms if the records still exist. To enable this functionality, you must now add code. Similarly, it should be possible to show the transactions related to a given inventory transaction origin. The code responsible for these two scenarios is located in the InventTransFormHelper class.

If you add new inventory transaction origin tables, you will most likely have to update the methods in this class to handle your inventory transaction originating table as well.

The ItemRouteId and ItemBomId fields

Inventory operations in Microsoft Dynamics AX have never been able to perform picking or other warehousing operations of items produced using specific bills of materials (BOMs) or routes. This information should not be located in the InventTrans table. No replacement fields have been added for these fields. If you have to upgrade code that references these fields, you will have to go back to the original inventory transaction originating table, such as SalesLine, to obtain the information.

Conclusion

This white paper has shown you how to implement the changes to the InventTrans table when developing new features or updating existing features. Using the new design pattern, you can implement relationships between the InventTrans table and inventory transaction originating tables resulting in a reduced amount of data stored (disk space) to avoid redundant data and the inherent risk of inconsistent data.

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.