Microsoft Dynamics® AX 2012

Securing data by dimension value by using XDS

White Paper

This white paper documents patterns and practices for creating and applying XDS security policies to filter financial data by dimension value.

Date: July 2012

http://www.microsoft.com/dynamics/ax

Author: Jeff Anderson, Senior Software Development Engineer

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

Table of Contents

Overview

In Microsoft Dynamics AX 2012, financial dimension data was changed from an array on tables to a foreign key. A concept of “entity backed” dimensions was also introduced, which allows some existing tables in the system to provide the domain of available values for a dimension. You can use the new functionality exposed by the Extensible Data Security (XDS) subsystem, along with the financial dimension data to create a security policy that only exposes financial information that has been filtered by a given dimension value.

The examples in this white paper describe how to filter financial data by department. However, the same patterns and practices could be used for any backing entity value.

The following list enumerates the primary steps that are documented in this white paper.

1. Determine which backing entity values are available for the current user.
2. Model queries to filter dimension queries to only include only these values.
3. Build XDS policies that apply these filters.
4. Constrain tables by using the XDS policies.

Scenario

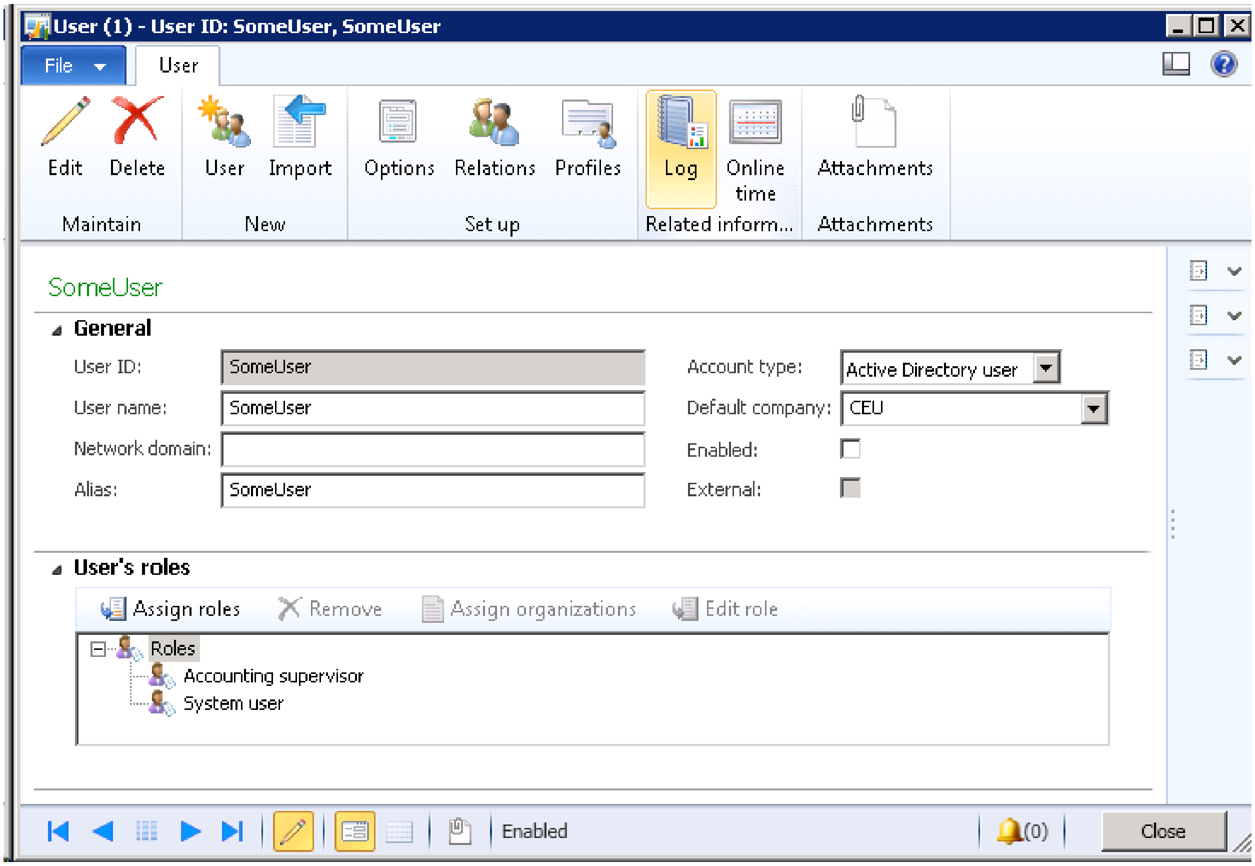
This white paper describes filtering financial data by department, which is stored in the OMOperatingUnit table with OMOperatingUnitType set to “Department”. The domain of values that may be used to specify the department dimension field is also exposed by the view OMDepartmentView. Departments are associated with users through the following relationships.

1. Users are related to zero or more workers through user relations.
2. Workers are assigned to zero or more positions.
3. Positions are assigned to zero or one departments.
4. Departments are organized in an organization hierarchy.

This white paper defines a user’s valid departments as all departments associated with a position the user is assigned to through a worker, as well as all departments which are descendants of those departments in a hierarchy. A more precise example is shown below for each of these steps.

1. Relate a user to a worker

Assign the user “SomeUser” to the “Accounting supervisor” role, which allows SomeUser to access many financial forms.



User (1) - User ID: SomeUser, SomeUser

SomeUser

General

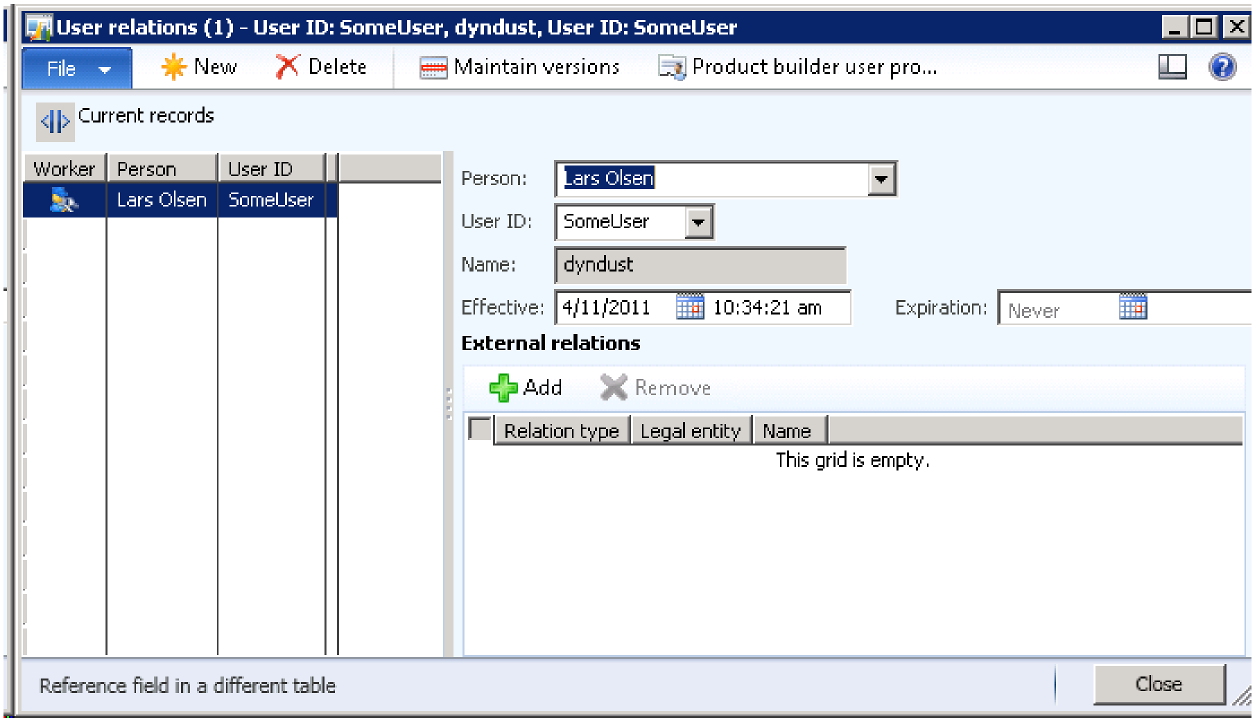
User ID:	SomeUser	Account type:	Active Directory user
User name:	SomeUser	Default company:	CEU
Network domain:		Enabled:	□
Alias:	SomeUser	External:	□

User's roles

- Roles
  - Accounting supervisor
  - System user

Figure 1: User form

Associate SomeUser to the worker Lars Olsen by selecting **System administration** > **Common** > **Users** > **Users** > **Set up** > **Relations**.


User relations (1) - User ID: SomeUser, dyndust, User ID: SomeUser
File
New
Delete
Maintain versions
Product builder user pro...
Current records
External relations
Add
Remove
Close

Worker	Person	User ID
	Lars Olsen	SomeUser

Person: Lars Olsen

User ID: SomeUser

Name: dyndust

Effective: 4/11/2011 10:34:21 am

Expiration: Never

□	Relation type	Legal entity	Name

This grid is empty.

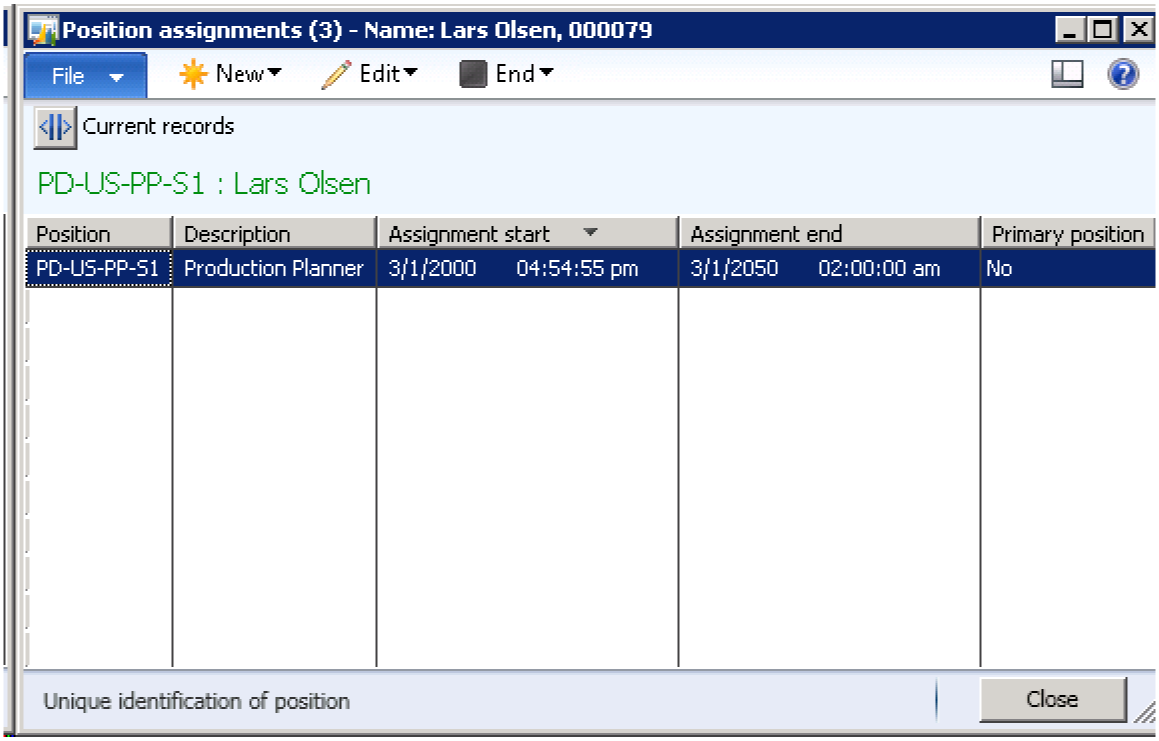
Reference field in a different table

Figure 2: User relations form

2. Assign a worker to a position

On the Workers form (**Human resources** > **Common** > **Workers** > **Workers**), select Lars Olsen, and then click **Worker position assignments**.

The **Position assignments** form opens and lists the positions to which a worker is assigned. In this example, Lars Olsen has been assigned the position PD-US-PP-S1. That means the user SomeUser is associated with position PP-US-PP-S1 through workers.



PD-US-PP-S1 : Lars Olsen

Position	Description	Assignment start	Assignment end	Primary position
PD-US-PP-S1	Production Planner	3/1/2000 04:54:55 pm	3/1/2050 02:00:00 am	No

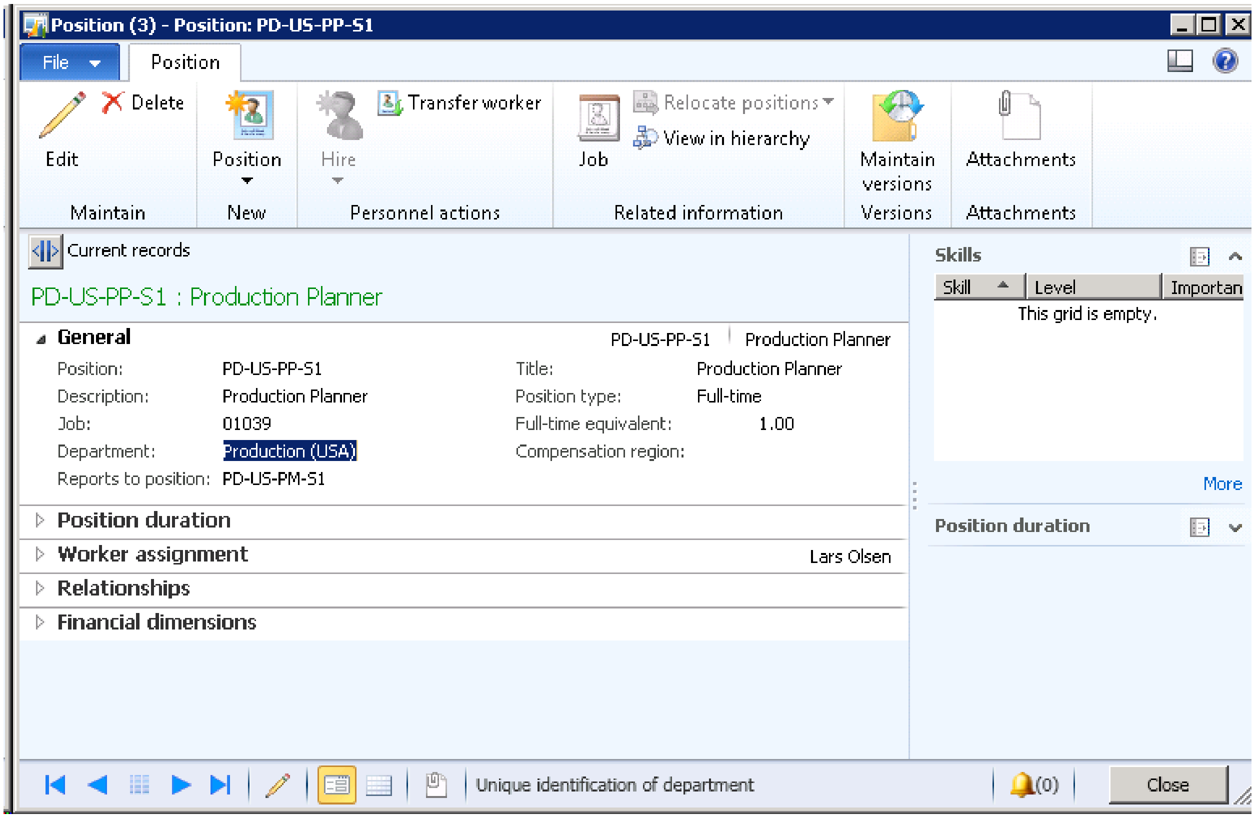
Unique identification of position

Figure 3: Position assignments form

3. Assign a position to a department

The **Position** column on the **Position assignments** form displays a list of position IDs that are links. On the **Position assignments** form, click the position ID link.

On the **Position** form, in the General FastTab, the department for the position is shown. In this example, PD-US-PP-S1 is associated with the Production department. That means SomeUser is associated with the Production department through workers and position assignments.



PD-US-PP-S1 : Production Planner

General — PD-US-PP-S1 | Production Planner

Position:	PD-US-PP-S1	Title:	Production Planner
Description:	Production Planner	Position type:	Full-time
Job:	01039	Full-time equivalent:	1.00
Department:	Production (USA)	Compensation region:	
Reports to position:	PD-US-PM-S1		

Position duration

Worker assignment — Lars Olsen

Relationships

Financial dimensions

Skills

Skill	Level	Importan
This grid is empty.		

More

Position duration

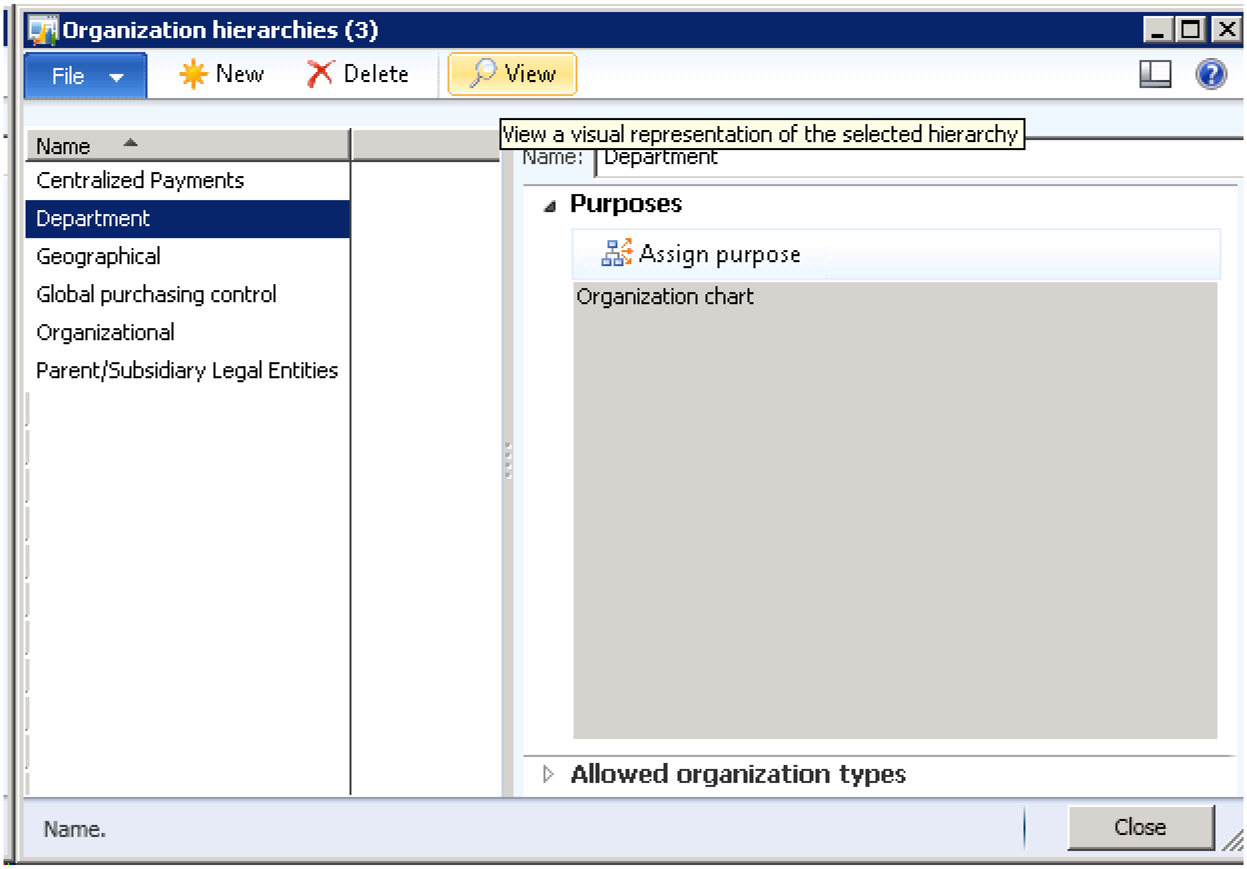
Unique identification of department

Figure 4: Position form

4. Organize a department in a hierarchy

You can organize departments in a hierarchy in which one department is subordinate to another department. Select **Organization administration** > **Setup** > **Organization** > **Organization hierarchies**. Select the “Department” hierarchy, and then click **View**.

The **Hierarchy designer** form opens.



View a visual representation of the selected hierarchy

Name
Centralized Payments
Department
Geographical
Global purchasing control
Organizational
Parent/Subsidiary Legal Entities

Name: Department

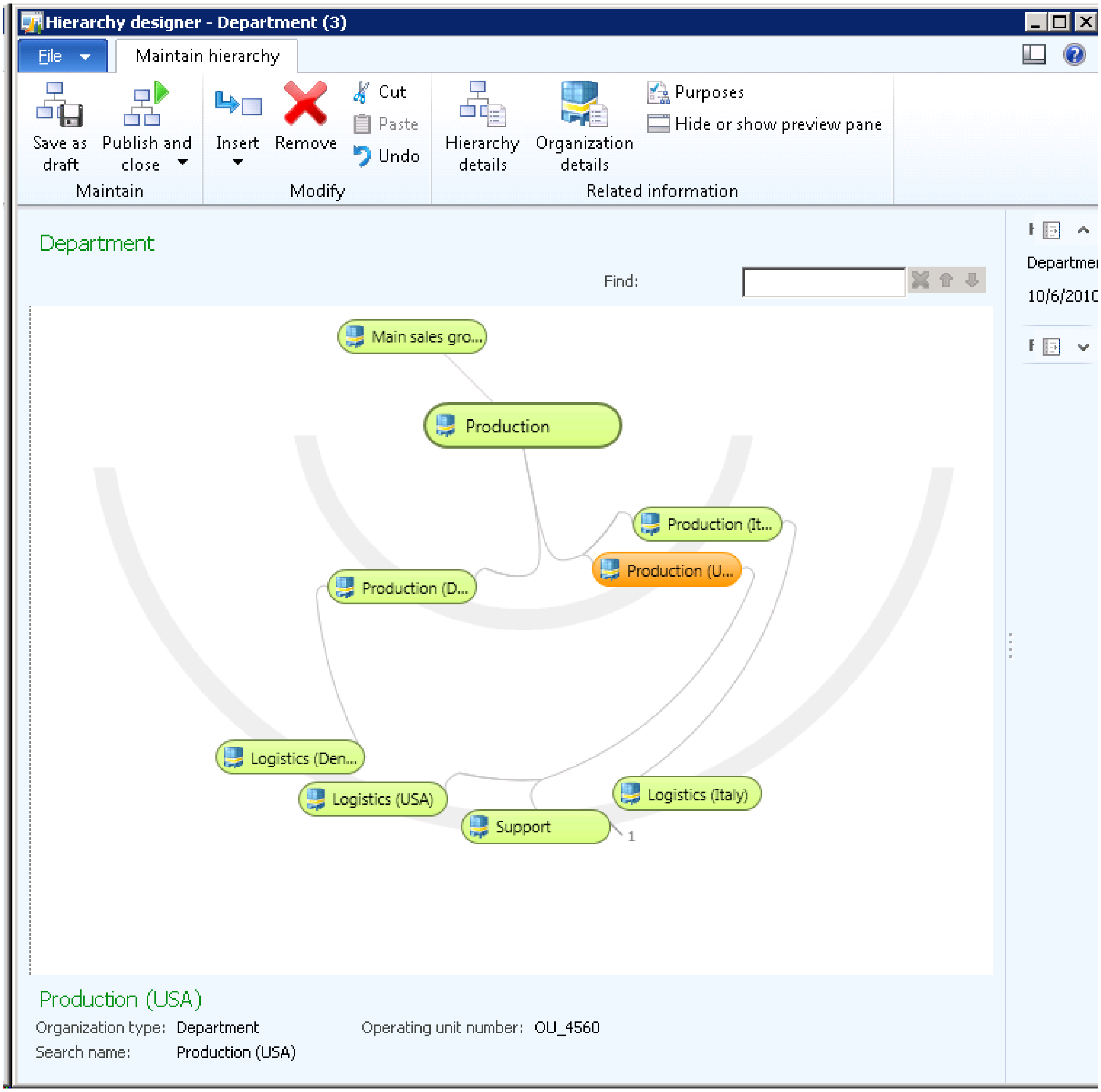
Organization chart

Name.

Figure 5: Organization hierarchies form

In this example, “Production (USA)” is defined as a child of the Production department, and a parent of “Logistics (USA)” and “Support”. Support has one child, “Customer Support”. Based on this hierarchy, the user SomeUser has access to the following four departments. All other departments should not be visible to SomeUser.

- Production (USA) is defined in the position assignments from the worker.
- Logistics (USA) is a descendent of Production (USA).
- Support is a descendent of Production (USA).
- Customer Support is a descendent of Production (USA).



Production (USA)

Organization type: Department

Search name: Production (USA)

Operating unit number: OU_4560

Figure 6: Hierarchy designer form

Determine available values

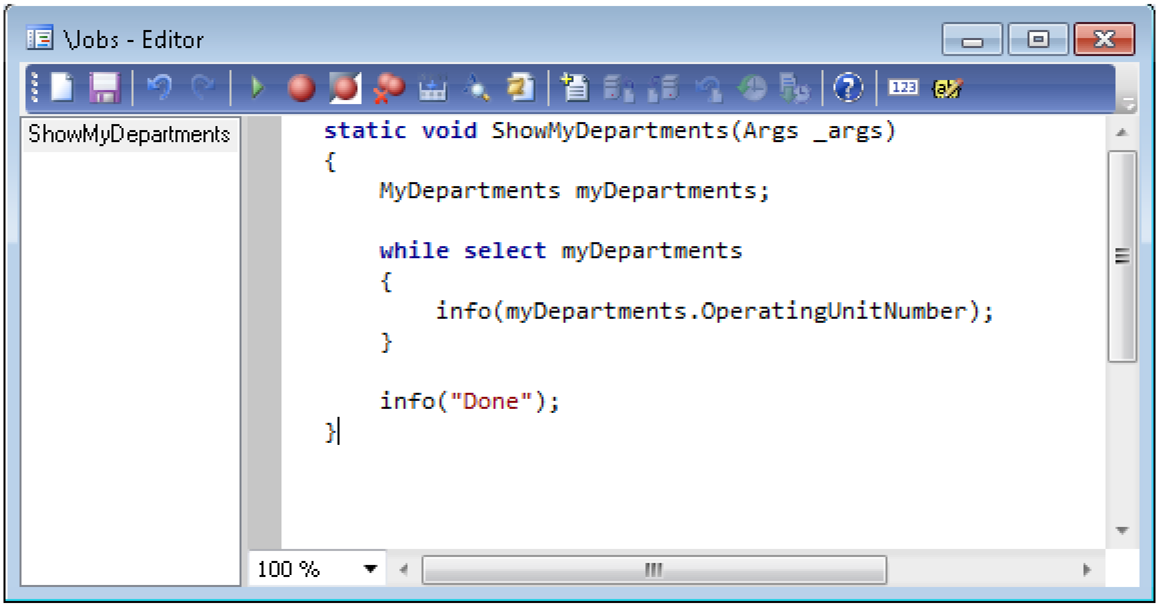
This white paper uses the construct “XDS MyConstructs” to provide the set of available departments to XDS. This construct is used because department assignment is very static data, yet calculating to which departments a user is assigned is fairly expensive. This construct already exists in the basic Microsoft Dynamics AX 2012 installation and is called MyDepartments.

Select **AOT** > **Data Dictionary** > **Tables** > **MyDepartments**.

Note: The table type is TempDB, which allows XDS policies to join to this table from a database level.

It has a method called xds() with a refresh frequency of PerSession, which means that the user’s department assignments are computed once per session. When the user logs off and logs on again, the department assignments will be recomputed.

The way this method computes available departments follows the association to system user described in the preceding sections. If the following job is run for a given user, only those departments associated to the user through workers, position assignments, and organization hierarchies will be returned.



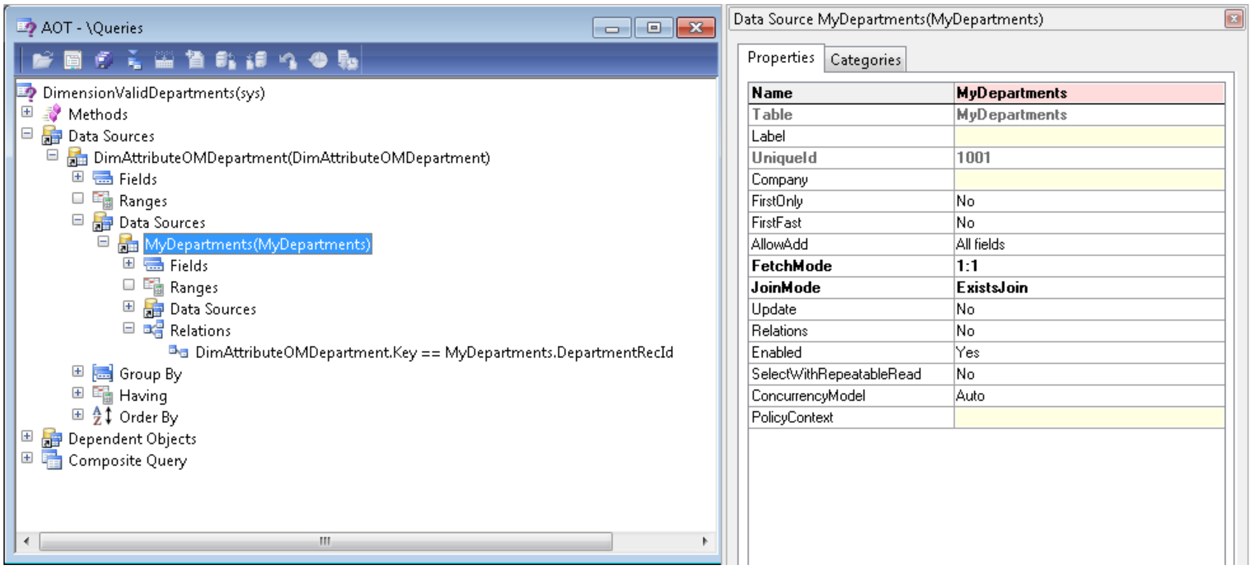
```
static void ShowMyDepartments(Args _args)
{
    MyDepartments myDepartments;

    while select myDepartments
    {
        info(myDepartments.OperatingUnitNumber);
    }

    info("Done");
}
```

Figure 7: ShowMyDepartments job

Modeling queries to filter dimension values

The financial dimensions subsystem uses a set of views named DimAttributeXXX to retrieve the domain of available values for a given entity-backed dimension. Because we are filtering by department in our example, we need to model a query to return the valid DimAttributeOMDepartment records.

The "Key" field in this view represents the RecId of the department, so an "exists join" can be used to join this view to the MyDepartments construct as shown below. The result of this view will be all DimAttributeOMDepartment dimension values to which the user has access, based on the filtering defined earlier in this white paper.



```
AOT - \Queries
DimensionValidDepartments(sys)
  Methods
  Data Sources
    DimAttributeOMDepartment(DimAttributeOMDepartment)
      Fields
      Ranges
      Data Sources
        MyDepartments(MyDepartments)
          Fields
          Ranges
          Data Sources
          Relations
            DimAttributeOMDepartment.Key == MyDepartments.DepartmentRecId
      Group By
      Having
      Order By
  Dependent Objects
  Composite Query
```

Data Source MyDepartments(MyDepartments)

Name	**MyDepartments**
Table	MyDepartments
Label	
UniqueId	1001
Company	
FirstOnly	No
FirstFast	No
AllowAdd	All fields
FetchMode	**1:1**
JoinMode	**ExistsJoin**
Update	No
Relations	No
Enabled	Yes
SelectWithRepeatableRead	No
ConcurrencyModel	Auto
PolicyContext	

Figure 8: Example of the "Key" field

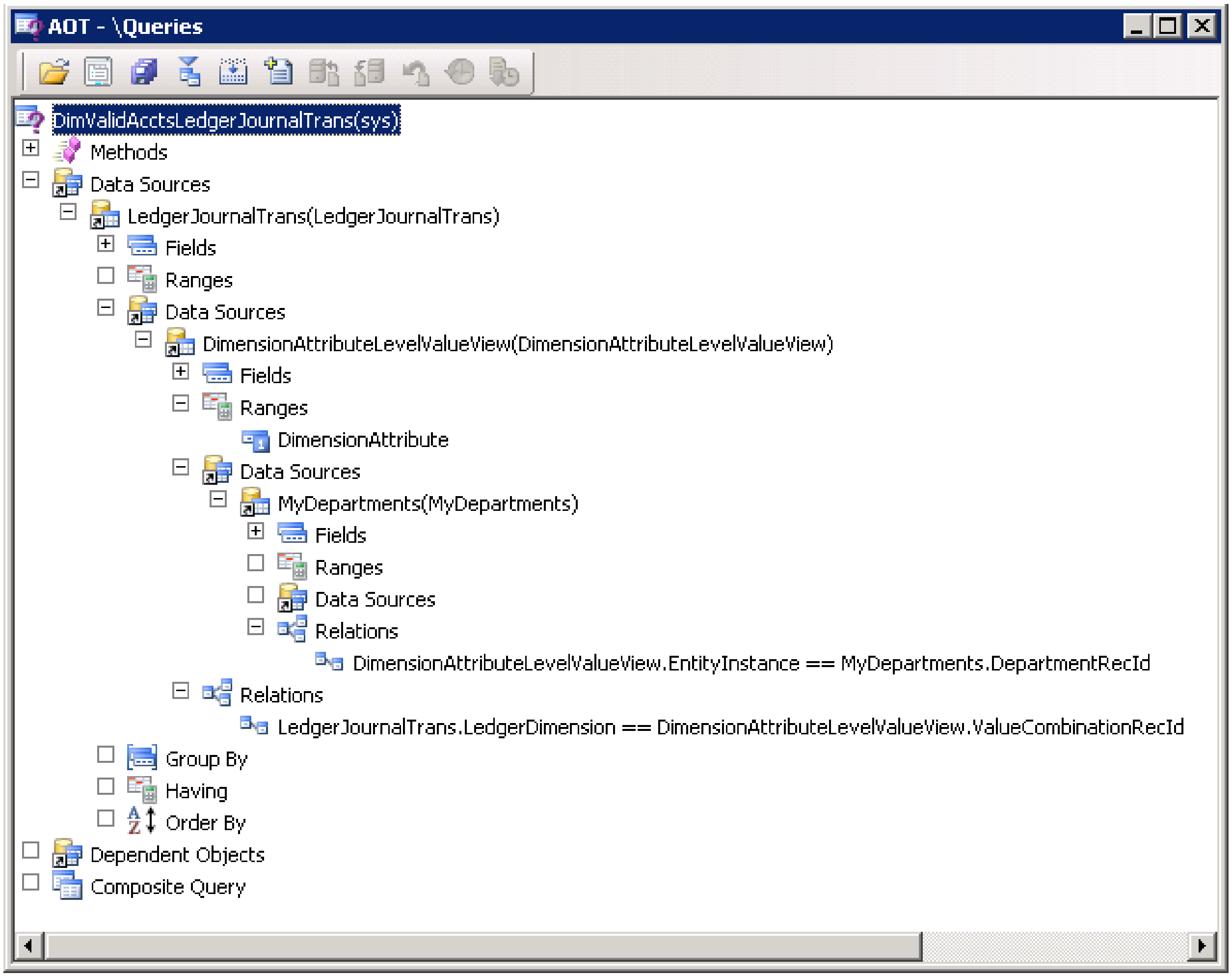
However, no tables hold a foreign key directly to the DimAttributeOMDepartment view. Instead, they hold foreign keys to DimensionAttributeValueCombination and DimensionAttributeValueSet, which are related to the DimAttributeOMDepartment view through a number of joins.

You will also need to model similar queries for tables DimensionAttributeValueCombination and DimensionAttributeValueSet.

The records in the DimensionAttributeValueCombination table are ledger accounts and default accounts. Tables such as LedgerJournalTrans hold a foreign key to this table, and logic on forms provides the interaction necessary to show the constituent dimension segments.

Note: The LedgerJournalTrans table should not be joined to the MyDepartments construct with an "exists join". Using an "exists join" would filter out combinations that do not have a department segment, which is incorrect. Instead, it should be joined by using two "not exists" joins. The query that contains the join structure returns all combinations that *do not* have a department segment that the user *does not* have rights to view. Those combinations with no department segment are fine to include, as are those that have a valid department segment.

In the following query, the JoinMode property should be set to NoExistsJoin for both DimensionAttributeLevelValueView and MyDepartments.



```
DimValidAcctsLedgerJournalTrans(sys)
⊞ Methods
⊟ Data Sources
    ⊟ LedgerJournalTrans(LedgerJournalTrans)
        ⊞ Fields
        □ Ranges
        ⊟ Data Sources
            ⊟ DimensionAttributeLevelValueView(DimensionAttributeLevelValueView)
                ⊞ Fields
                ⊟ Ranges
                      DimensionAttribute
                ⊟ Data Sources
                    ⊟ MyDepartments(MyDepartments)
                        ⊞ Fields
                        □ Ranges
                        □ Data Sources
                        ⊟ Relations
                              DimensionAttributeLevelValueView.EntityInstance == MyDepartments.DepartmentRecId
                ⊟ Relations
                      LedgerJournalTrans.LedgerDimension == DimensionAttributeLevelValueView.ValueCombinationRecId
        □ Group By
        □ Having
        □ Order By
□ Dependent Objects
□ Composite Query
```

Figure 9: Example of a query

The top data source in the query is the table that is being filtered – LedgerJournalTrans in this example. Therefore, a different query will need to be created for each of the tables that must be filtered. The reason for creating separate queries instead of using free-form expressions on query policies is performance. In order to use free-form query policy expressions, an "OR" clause must be used. This has been found in volume dataset testing to result in bad SQL query plans that cannot be fixed through indexing alone. By creating separate queries and policies for each table, you avoid these bad query plans and maintain acceptable performance.

Another important piece of this query is the range on DimensionAttributeLevelValueView.DimensionAttribute. It is possible to set up multiple dimension attributes backed by the same underlying entity that are used for different purposes. For example, you can have one dimension attribute called "department" and another called "trading partner", both backed by the department entity. In this case, the combinations should filter by valid "department" but allow the user to see other "trading partner" values that are not related to the user's departments.

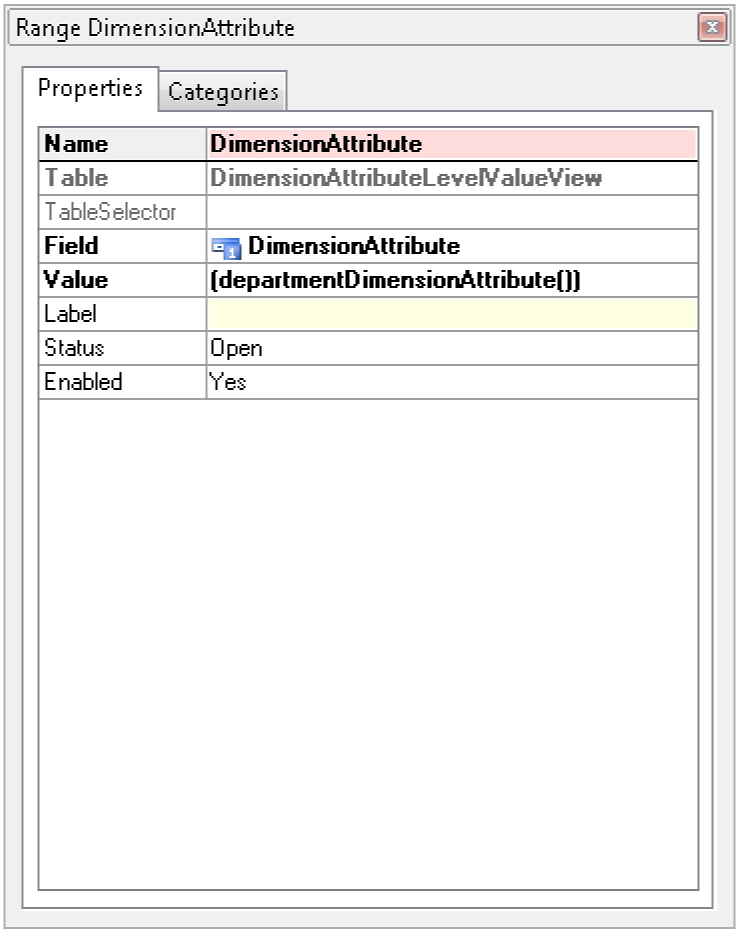
To support this scenario, the dimension values need to be filtered to only those in which DimensionAttribute is set to the RecId of the proper dimension attribute. The easiest way to do this is to add a new functional predicate to the SysQueryRangeUtil class.

The following code finds the RecId based on a DimensionAttribute invariant name. This code can be replaced with any other method of determining which dimension is interesting for filtering.

```
/// <summary>
///    Gets the ID of the dimension attribute for departments.
/// </summary>
/// <returns>
///    The ID of the dimension attribute for departments.
/// </returns>
public static RecId departmentDimensionAttribute()
{
    return DimensionAttribute::findByName("DEPT").RecId;
}
```

Figure 10: Code example

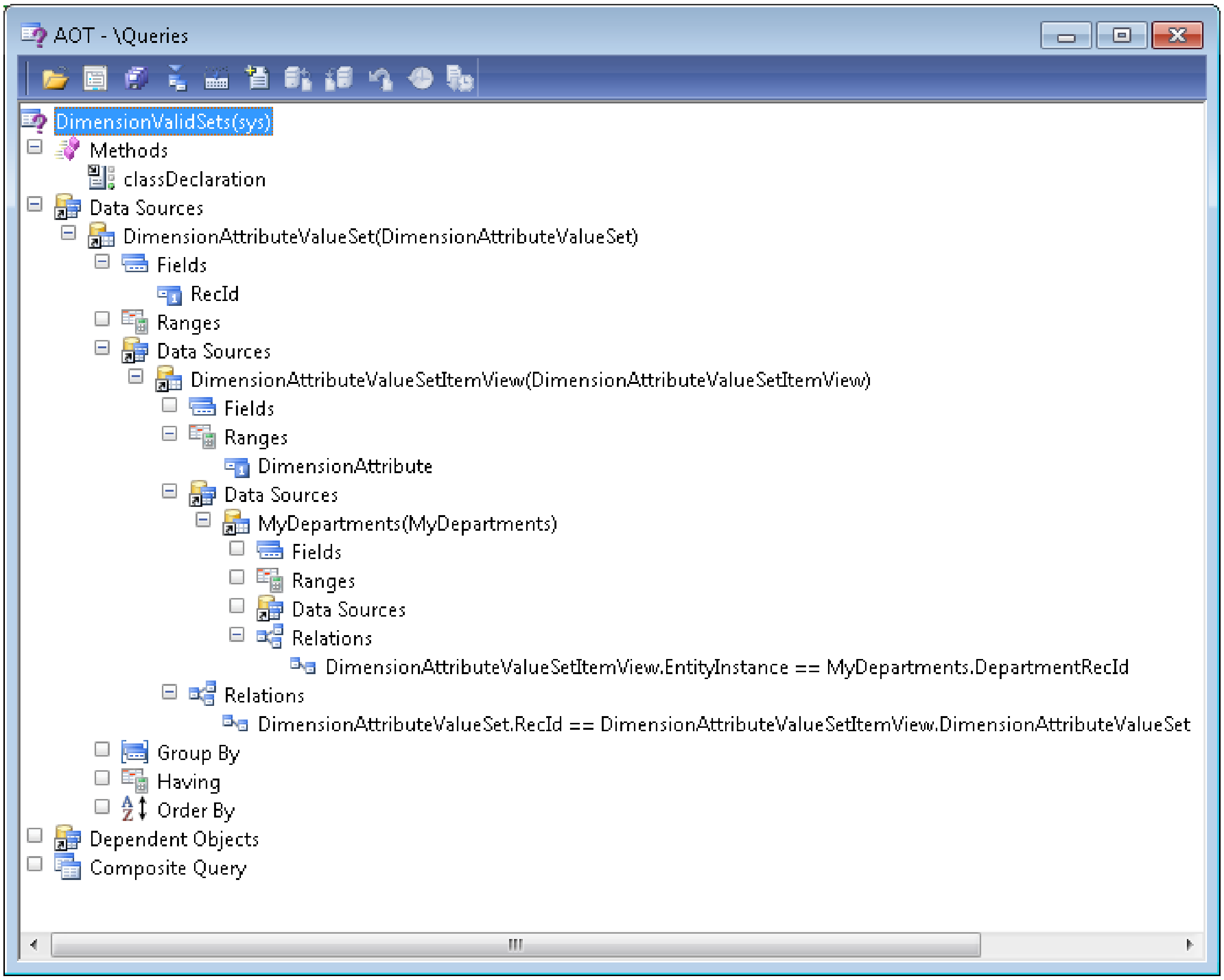
After this functional predicate has been created, it can be referenced as the value for a range on the DimensionAttribute field.



Name	DimensionAttribute
Table	DimensionAttributeLevelValueView
TableSelector	
Field	DimensionAttribute
Value	(departmentDimensionAttribute())
Label	
Status	Open
Enabled	Yes

Figure 11: Example of the field value DimensionAttribute

A similar query then needs to be modeled for the DimensionAttributeValueSet table.



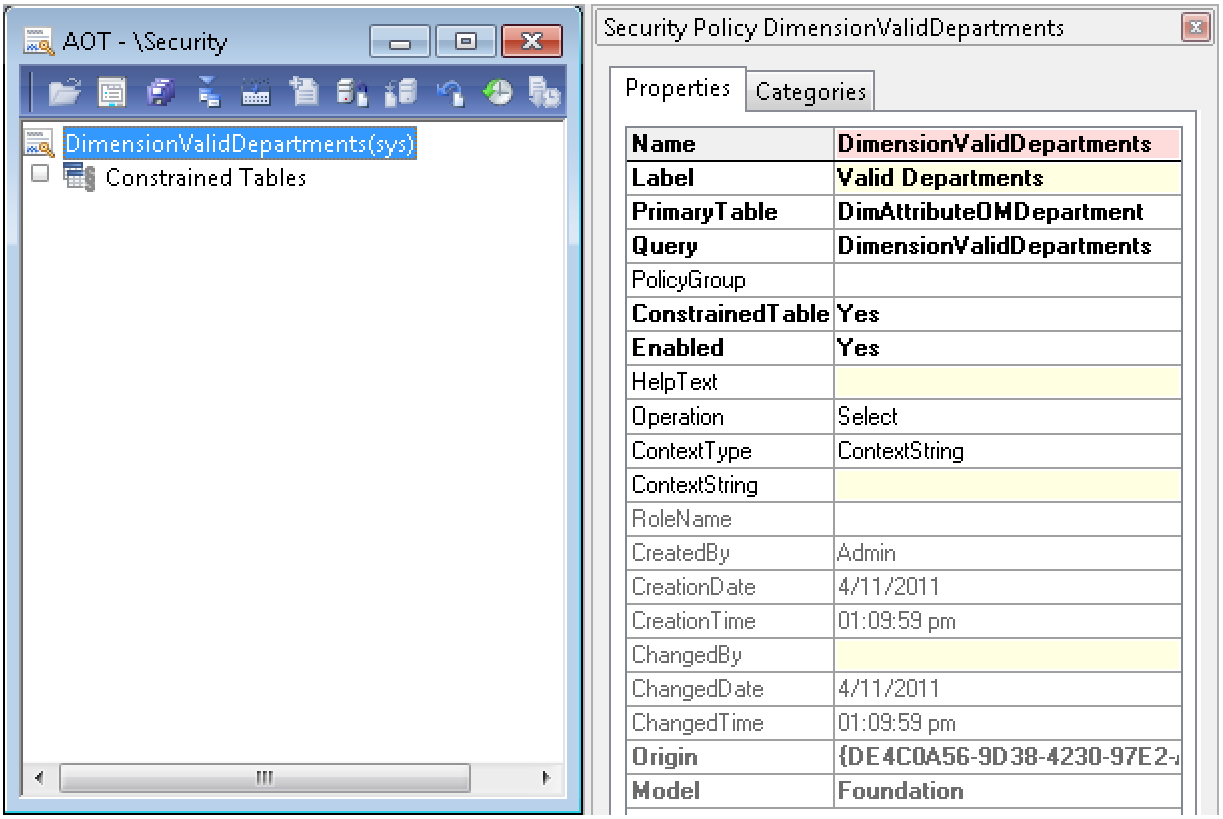
AOT - \Queries

```
DimensionValidSets(sys)
  Methods
    classDeclaration
  Data Sources
    DimensionAttributeValueSet(DimensionAttributeValueSet)
      Fields
        RecId
      Ranges
      Data Sources
        DimensionAttributeValueSetItemView(DimensionAttributeValueSetItemView)
          Fields
          Ranges
            DimensionAttribute
          Data Sources
            MyDepartments(MyDepartments)
              Fields
              Ranges
              Data Sources
              Relations
                DimensionAttributeValueSetItemView.EntityInstance == MyDepartments.DepartmentRecId
          Relations
            DimensionAttributeValueSet.RecId == DimensionAttributeValueSetItemView.DimensionAttributeValueSet
      Group By
      Having
      Order By
  Dependent Objects
  Composite Query
```

Figure 12: Example of a query

Security policies

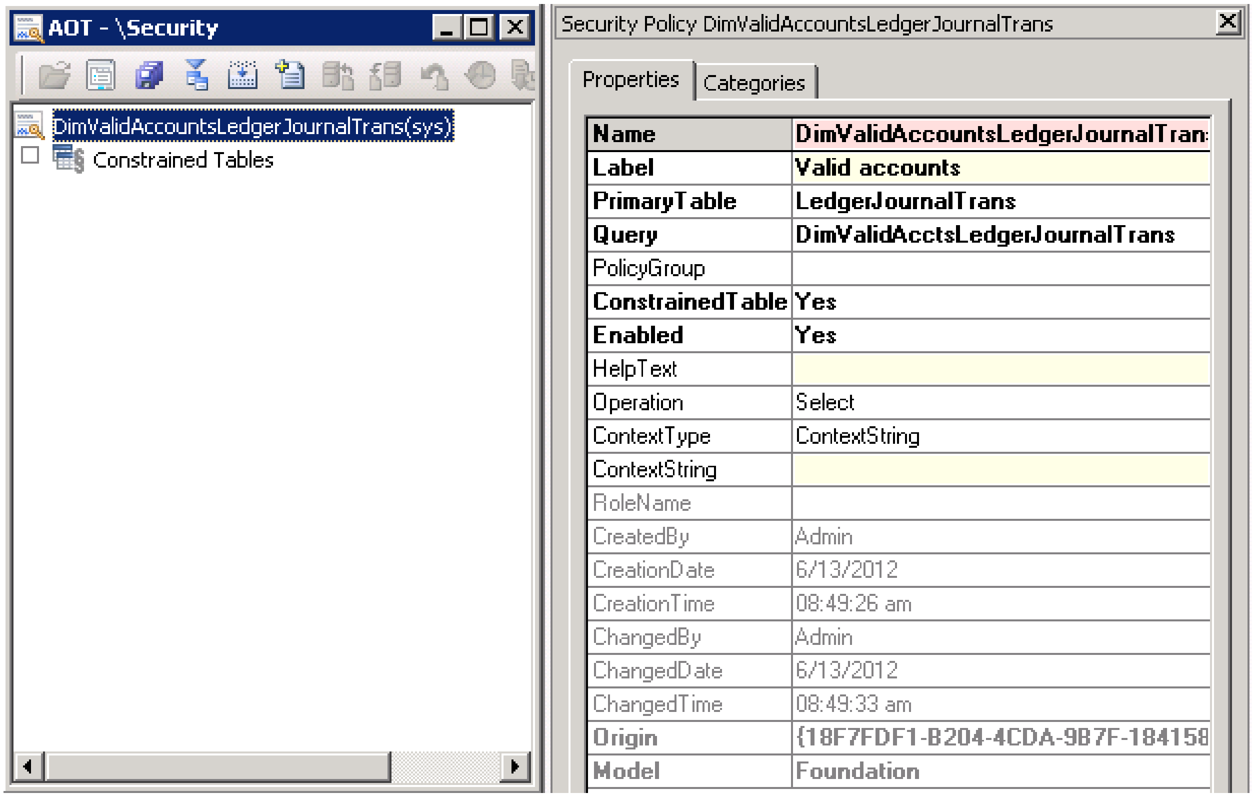
The first security policy that is needed filters the valid department values in tables places such as dimension lookups. It is a very simple policy associated with the DimensionValidDepartments query that was described earlier in this white paper, in which the ConstrainedTable is set to true. There are no additional constrained tables that need to be added because nothing holds a foreign key directly to the DimAttributeOMDepartment table.


AOT - \Security
DimensionValidDepartments(sys)
Constrained Tables
Security Policy DimensionValidDepartments
Properties
Categories

Name	**DimensionValidDepartments**
Label	**Valid Departments**
PrimaryTable	**DimAttributeOMDepartment**
Query	**DimensionValidDepartments**
PolicyGroup	
ConstrainedTable	**Yes**
Enabled	**Yes**
HelpText	
Operation	Select
ContextType	ContextString
ContextString	
RoleName	
CreatedBy	Admin
CreationDate	4/11/2011
CreationTime	01:09:59 pm
ChangedBy	
ChangedDate	4/11/2011
ChangedTime	01:09:59 pm
Origin	**{DE4C0A56-9D38-4230-97E2-**
Model	**Foundation**

Figure 13: Example of a security policy

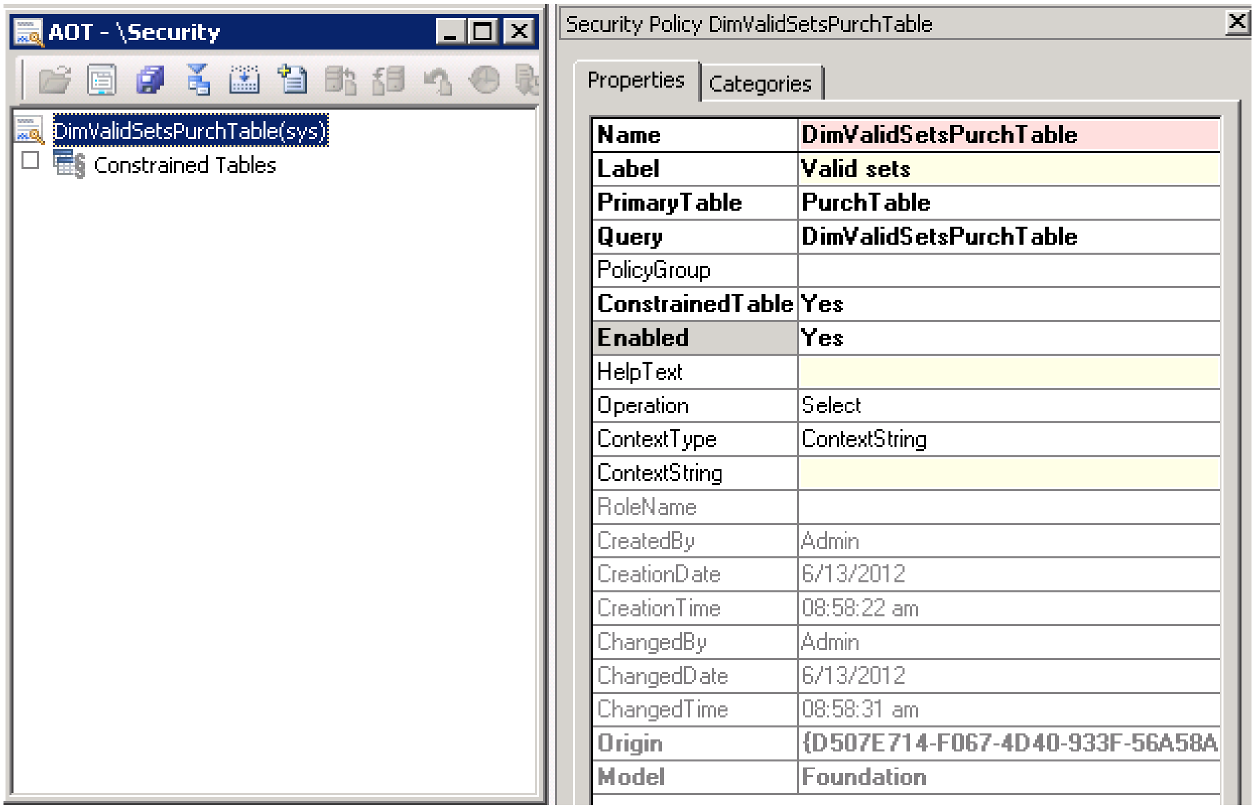
The second policy that is needed restricts the list of available ledger accounts to the valid values determined by the DimValidAcctsLedgerJournalTrans query. The policy itself is set up just like the DimensionValidDepartments policy. A similar policy needs to be set up for each LedgerDimension field that is being secured.


AOT - \Security
DimValidAccountsLedgerJournalTrans(sys)
Constrained Tables
Security Policy DimValidAccountsLedgerJournalTrans
Properties
Categories

Name	**DimValidAccountsLedgerJournalTran:**
Label	**Valid accounts**
PrimaryTable	**LedgerJournalTrans**
Query	**DimValidAcctsLedgerJournalTrans**
PolicyGroup	
ConstrainedTable	**Yes**
Enabled	**Yes**
HelpText	
Operation	Select
ContextType	ContextString
ContextString	
RoleName	
CreatedBy	Admin
CreationDate	6/13/2012
CreationTime	08:49:26 am
ChangedBy	Admin
ChangedDate	6/13/2012
ChangedTime	08:49:33 am
Origin	**{18F7FDF1-B204-4CDA-9B7F-184158**
Model	**Foundation**

Figure 14: Example of a security policy

You can apply the preceding practice to filter default dimensions such as PurchTable.DefaultDimension.


AOT - \Security
DimValidSetsPurchTable(sys)
Constrained Tables
Security Policy DimValidSetsPurchTable
Properties
Categories

Name	DimValidSetsPurchTable
Label	Valid sets
PrimaryTable	PurchTable
Query	DimValidSetsPurchTable
PolicyGroup	
ConstrainedTable	Yes
Enabled	Yes
HelpText	
Operation	Select
ContextType	ContextString
ContextString	
RoleName	
CreatedBy	Admin
CreationDate	6/13/2012
CreationTime	08:58:22 am
ChangedBy	Admin
ChangedDate	6/13/2012
ChangedTime	08:58:31 am
Origin	{D507E714-F067-4D40-933F-56A58A
Model	Foundation

Figure 18: Example of filtered default dimensions by the DimensionAttributeValueSet table

Impact

While the general impact of applying these security policies is what would be expected, there are a few side effects that should be considered.

First, journal records are *not* filtered by the lines that they contain. A more advanced XDS policy could be constructed to perform this additional filtering, but the additional joins would be fairly costly from a performance perspective. Determining which journals existed would require joining LedgerJournalTable to LedgerJournalTrans, and then also joining all of the security information. Assuming the journal records are not filtered, that means a user may see fewer lines in a given journal than actually exist. In this case, the journal totals will not match the total amount for all lines shown on the form, because some lines are hidden. The filtering of journal records is configurable based on how the XDS policy is constructed.

Second, there is a performance cost to applying additional security. Instead of simply querying a table such as LedgerJournalTrans directly, the system needs to join through a few other tables to determine which records should be visible. The performance cost, measured in SQL Server processor and memory load as well as in AOS processor and memory load, is minimized by pre-compiling the list of valid departments using the MyDepartments construct, but there is still overhead to the join. The added cost of this join should be weighed against the required performance characteristics of any given scenario. This decision can be made on a table-by-table basis as well because constrained tables are added individually to security policies.

Dimension tables and fields

This section provides a list of tables and fields that may be filtered using the practices described in this white paper.

Individual dimension values

Individual values are queried in only the dimension forms and dimension lookups. Setting up the DimensionValidDepartments security policy and setting ConstrainedTable to true will automatically filter all these locations. No additional constrained tables are necessary.

Ledger accounts

The following table fields hold a foreign key to the DimensionAttributeValueCombination table and therefore can be secured by using the pattern shown in the DimValidAcctsLedgerJournalTrans policy. Tables that contain more than one field will require the use of additional policies.

Note: The set of ledger account fields also includes budget ledger dimensions that should be filtered by using the same filtering practice that you used for ledger accounts.

AccountingDistribution.LedgerDimension

AssetBudget.LedgerDimension

AssetBudget.OffsetLedgerDimension

BankDocumentPosting.BankLGLiquidationLedgerDimension

BankDocumentPosting.MarginLedgerDimension

BankDocumentPosting.MiscChargesLedgerDimension

BankDocumentPosting.SettleLedgerDimension

BudgetCheckResultErrorWarningDetail.BudgetControlLedgerDimension

BudgetGroupLedgerDimension.BudgetControlLedgerDimension

BudgetSourceTrackingDetail.BudgetControlLedgerDimension

BudgetSourceTrackingSummary.BudgetControlLedgerDimension

BudgetTransactionLine.LedgerDimension

CustInvoiceDistributionTemplate.LedgerDimension

CustVendPaymJournalFee.LedgerDimension

CustVendPaymProposalLine.LedgerDimension

DimensionFocusBalance.FocusLedgerDimension

DimensionFocusBalanceCalculationView.FocusLedgerDimension

GeneralJournalAccountEntry.LedgerDimension

GeneralJournalAccountEntryHash.LedgerDimension

InventCostTransVariance.LedgerDimension

InventCostTransVariance.OffsetLedgerDimension

InventSettlement.BalanceSheetLedgerDimension

InventSettlement.OperationsLedgerDimension

InventTransPosting.LedgerDimension

InventTransPosting.OffsetLedgerDimension

InventValueReportFinancialAdjustment.LedgerDimension

InventValueReportFinancialAdjustment.LedgerDimensionOffset

InventValueReportFinancialBalance.LedgerDimension

InventValueReportFinancialBalance.LedgerDimensionOffset

InventValueReportFinancialTransaction.LedgerDimension

InventValueReportFinancialTransaction.LedgerDimensionOffset

InventValueReportPhysicalAdjustment.LedgerDimension

InventValueReportPhysicalAdjustment.LedgerDimensionOffset

InventValueReportPhysicalBalance.LedgerDimension

InventValueReportPhysicalBalance.LedgerDimensionOffset

InventValueReportPhysicalReversed.LedgerDimension

InventValueReportPhysicalReversed.LedgerDimensionOffset

InventValueReportPhysicalSettlement.LedgerDimension

InventValueReportPhysicalSettlement.LedgerDimensionOffset

InventValueReportPhysicalTransaction.LedgerDimension

InventValueReportPhysicalTransaction.LedgerDimensionOffset

InventValueReportUnionAll.LedgerDimension

InventValueReportUnionAll.LedgerDimensionOffset

JmgIpcLedgerTrans.LedgerDimension

JmgIpcLedgerTrans.OffsetLedgerDimension

LedgerAllocationRuleDestination.ToLedgerDimension

LedgerClosingTrans.LedgerDimension

LedgerClosingTrans.OffsetLedgerDimension

LedgerDimTransactionMap.LedgerDimension

LedgerJournalTrans.LedgerDimension

LedgerJournalTrans.OffsetLedgerDimension

LedgerJournalTransAccrualTrans.LedgerDimension

LedgerJournalTransLedgerDimensionView.LedgerDimension

LedgerJournalTransLedgerDimensionView.OffsetLedgerDimension

LedgerOpeningTrans_ES.LedgerDimension

LedgerOpeningTrans_ES.OffsetLedgerDimension

ProdIndirectTrans.EstimatedIndirectAbsorpLedgerDimension

ProdIndirectTrans.EstimIndirectAbsorpOffsetLedgerDimension

ProdIndirectTrans.IndirectAbsorptionLedgerDimension

ProdIndirectTrans.IndirectAbsorptionOffsetLedgerDimension

ProdRouteTrans.IssueLedgerDimension

ProdRouteTrans.IssueOffsetLedgerDimension

ProdRouteTrans.WIPIssueLedgerDimension

ProdRouteTrans.WIPValuationLedgerDimension

SMAServiceOrderLine.LedgerDimension

SubledgerJournalAccountEntry.LedgerDimension

SubLedgerJournalAccountEntryView.LedgerDimension

TaxJournalTrans.OperationLedgerDimension

TaxTransGeneralJournalAccountEntry.LedgerDimension

TaxTransTransactionLineLedgerDimension.LedgerDimId

TaxTransTransactionLineLedgerDimension.LedgerDimTaxId

TaxUncommitted.LedgerDimension

TaxUncommitted.OperationLedgerDimension

TaxUncommitted.TaxOffsetUseTaxLedgerDimension

VendReportApproveCollection.LedgerDimension

Default dimensions

The following table fields hold a foreign key to the DimensionAttributeValueSet table and therefore can be secured by using the pattern shown in the DimValidSetsPurchTable policy.

AccountingDistributionTemplateDetail.DefaultDimension

AssetBook.DefaultDimension

AssetBookMerge.DefaultDimension

AssetDepBookTrans.DefaultDimension

AssetLending.DefaultDimension

AssetTrans.DefaultDimension

AssetTransferHistory.FromDimension

AssetTransferHistory.ToDimension

AssetTransMerge.DefaultDimension

BankAccountTable.DefaultDimension

BankAccountTrans.DefaultDimension

BankCustPaymModeBankAccounts.DefaultDimension

BankDeposit.DefaultDimension

BankVendPaymModeBankAccounts.DefaultDimension

BOMParmReportFinish.DefaultDimension

BudgetAllocationTermDetail.DefaultDimension

BudgetMap.defaultDimension

CommissionTrans.DefaultDimension

CostControlTransCommittedCost.DefaultDimension

CostControlTransCommittedCostCube.DefaultDimension

CostSheetNodeTable.DefaultDimension

CustCollectionLetterJour.DefaultDimension

CustConfirmJour.DefaultDimension

CustConfirmTrans.DefaultDimension

CustInterestJour.DefaultDimension

CustInvoiceJour.DefaultDimension

CustInvoiceLine.DefaultDimension

CustInvoiceStandardLineTemplate.DefaultDimension

CustInvoiceTable.DefaultDimension

CustInvoiceTemplate.DefaultDimension

CustInvoiceTrans.DefaultDimension

CustPackingSlipJour.DefaultDimension

CustPackingSlipTrans.DefaultDimension

CustQuotationConfirmJour.DefaultDimension

CustQuotationConfirmTrans.DefaultDimension

CustQuotationJour.DefaultDimension

CustQuotationTrans.DefaultDimension

CustSettlement.DefaultDimension

CustTable.DefaultDimension

CustTrans.DefaultDimension

CustVendInvoiceJour.DefaultDimension

CustVendPaymJournalFee.DefaultDimension

CustVendPaymProposalLine.DefaultDimension

CustVendSettlement.DefaultDimension

CustVendTable.DefaultDimension

CustVendTrans.DefaultDimension

DimensionDefaultMap.DefaultDimension

ForecastPurch.DefaultDimension

ForecastSales.DefaultDimension

FormletterJournal.DefaultDimension

HcmCourseType.DefaultDimension

HcmCourseTypeDefaultDimension.DefaultDimension

HcmPositionDefaultDimension.DefaultDimension

HRMAbsenceCode.DefaultDimension

HRMCompFixedEmpl.DefaultDimension

HRMCompVarAwardEmpl.DefaultDimension

HRMCompVarEnrollEmpl.DefaultDimension

HRMCourseAttendee.DefaultDimension

HRMCourseTable.DefaultDimension

InventCostTrans.DefaultDimension

InventCostTransMap.DefaultDimension

InventCostTransSum.DefaultDimension

InventCostTransVariance.DefaultDimension

InventCostTransView.DefaultDimension

InventJournalTrans.DefaultDimension

InventNonConformanceTable.DefaultDimension

InventQualityOrderTable.DefaultDimension

InventQuarantineOrder.DefaultDimension

InventSettlement.DefaultDimension

InventSite.DefaultDimension

InventTable.DefaultDimension

InventTransPosting.DefaultDimension

InventValueReportIndirectBalance.DefaultDimension

InventValueReportIndirectFinTransaction.DefaultDimension

InventValueReportIndirectPhysTransaction.DefaultDimension

InventValueReportIndirectUnionAll.DefaultDimension

InventValueReportSubContBalance.DefaultDimension

InventValueReportSubContFinTransaction.DefaultDimension

InventValueReportSubContPhysTransaction.DefaultDimension

InventValueReportSubContUnionAll.DefaultDimension

InventValueReportWrkCtrBalance.DefaultDimension

InventValueReportWrkCtrFinTransaction.DefaultDimension

InventValueReportWrkCtrPhysTransaction.DefaultDimension

InventValueReportWrkCtrUnionAll.DefaultDimension

JmgIpcActivity.DefaultDimension

JmgIpcJournalTrans.DefaultDimension

JmgIpcTrans.DefaultDimension

JmgPayAddTable.DefaultDimension

JmgPayEvents.DefaultDimension

JmgScheduledLoan.DefaultDimension

JmgStampJournalTable.DefaultDimension

JmgStampJournalTrans.DefaultDimension

JmgStampTrans.DefaultDimension

JmgStampTransMap.DefaultDimension

Kanban.DefaultDimension

KanbanJob.DefaultDimension

KanbanJobPickingList.DefaultDimension

LedgerAllocateTrans.DefaultDimension

LedgerAllocation.FromDefaultDimension

LedgerAllocation.ToDefaultDimension

LedgerAllocationRule.OffsetDefaultDimension

LedgerAllocationRuleDestination.ToDefaultDimension

LedgerAllocationTmpSource.DefaultDimension

LedgerCov.DefaultDimension

LedgerEliminationRuleLine.DefaultDimension

LedgerJournalName.DefaultDimension

LedgerJournalTable.DefaultDimension

LedgerJournalTrans.DefaultDimension

LedgerJournalTrans.OffsetDefaultDimension

LedgerTransferOpeningSumTmp.DefaultDimension

MainAccountLegalEntity.DefaultDimension

PlanReference.DefaultDimension

ProdBOM.DefaultDimension

ProdBOMTransProj.DefaultDimension

ProdIndirectTrans.DefaultDimension

ProdJournalBOM.DefaultDimension

ProdJournalRoute.DefaultDimension

ProdRoute.DefaultDimension

ProdRouteTrans.DefaultDimension

ProdTable.DefaultDimension

ProjBegBalJournalTrans_CostSales.DefaultDimension

ProjBegBalJournalTrans_Fee.DefaultDimension

ProjBegBalJournalTrans_OnAcc.DefaultDimension

ProjBIEmplTrans.DefaultDimension

ProjBIForecastEmpl.DefaultDimension

ProjClosingProfile.DefaultDimension

ProjCostTrans.DefaultDimension

ProjCostTransCost.DefaultDimension

ProjCostTransSale.DefaultDimension

ProjEmplTrans.DefaultDimension

ProjEmplTransCost.DefaultDimension

ProjEmplTransSale.DefaultDimension

ProjForecastCost.DefaultDimension

ProjForecastEmpl.DefaultDimension

ProjForecastOnAcc.DefaultDimension

ProjForecastRevenue.DefaultDimension

ProjInventJournalTransMap.DefaultDimension

ProjInvoiceCost.DefaultDimension

ProjInvoiceEmpl.DefaultDimension

ProjInvoiceItem.DefaultDimension

ProjInvoiceJour.DefaultDimension

ProjInvoiceOnAcc.DefaultDimension

ProjInvoiceRevenue.DefaultDimension

ProjInvoiceTable.DefaultDimension

ProjItemTrans.DefaultDimension

ProjItemTransCost.DefaultDimension

ProjItemTransSale.DefaultDimension

ProjJournalTrans.DefaultDimension

ProjJournalTransMap.DefaultDimension

ProjOnAccTrans.DefaultDimension

ProjOnAccTransSale.DefaultDimension

ProjProposalJour.DefaultDimension

ProjRevenueTrans.DefaultDimension

ProjRevenueTransSale.DefaultDimension

ProjTable.DefaultDimension

ProjTableCube.DefaultDimension

ProjTimesheetLineMap.DefaultDimension

ProjTransBudget.DefaultDimension

ProjTransBudgetCube.DefaultDimension

ProjTransPosting.DefaultDimension

ProjTransPostingCube.DefaultDimension

PurchLine.DefaultDimension

PurchLineAllVersions.DefaultDimension

PurchLineArchivedVersions.DefaultDimension

PurchLineHistory.DefaultDimension

PurchLineMap.DefaultDimension

PurchLineNotArchivedVersions.DefaultDimension

PurchReqLine.DefaultDimension

PurchReqLineHistory.DefaultDimension

PurchRFQCaseLine.DefaultDimension

PurchRFQCaseTable.DefaultDimension

PurchRFQLine.DefaultDimension

PurchRFQLineMap.DefaultDimension

PurchRFQTable.DefaultDimension

PurchRFQTableMap.DefaultDimension

PurchTable.DefaultDimension

PurchTableAllVersions.DefaultDimension

PurchTableArchivedVersions.DefaultDimension

PurchTableHistory.DefaultDimension

PurchTableMap.DefaultDimension

PurchTableNotArchivedVersions.DefaultDimension

SalesLine.DefaultDimension

SalesLineCube.DefaultDimension

SalesPurchLine.DefaultDimension

SalesQuotationLine.DefaultDimension

SalesQuotationTable.DefaultDimension

SalesTable.DefaultDimension

SMAAgreementLine.DefaultDimension

SMAServiceLineMap.DefaultDimension

SMAServiceOrderLine.DefaultDimension

smmBusRelTable.DefaultDimension

smmImportRelationJournal.DefaultDimension

TrvCashAdvance.DefaultDimension

TrvExpTable.DefaultDimension

TrvExpTableCube.DefaultDimension

TrvExpTrans.DefaultDimension

TrvRequisitionLine.DefaultDimension

TrvRequisitionTable.DefaultDimension

TSTimesheetLine.DefaultDimension

TSTimesheetSummaryLine.DefaultDimension

VendInvoiceInfoLine.DefaultDimension

VendInvoiceInfoTable.DefaultDimension

VendInvoiceJour.DefaultDimension

VendInvoiceTrans.DefaultDimension

VendPackingSlipJour.DefaultDimension

VendPackingSlipTrans.DefaultDimension

VendRFQJour.DefaultDimension

VendRFQTrans.DefaultDimension

VendSettlement.DefaultDimension

VendTable.DefaultDimension

VendTrans.DefaultDimension

WrkCtrTable.DefaultDimension

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.