Microsoft Dynamics[®] AX 2012

Migrating Extended Data Type Relations in Microsoft Dynamics AX 2012

White Paper

[Abstract]

Date: May 11, 2011



Table of Contents

Overview	Error!	Bookmark n	ot defined.
Migration Process	Error!	Bookmark n	ot defined.
Migration Steps EDT Relations Migration Tool			

Using the EDT Relations Migration Tool..... Error! Bookmark not defined.

Migration Scenarios E	rror! Bookmark not defined.
Scenario 1	
Scenario 2	
Scenario 3	
Scenario 4	
Case 1	
Case 2	
R	

Overview

Relations between extended data types (EDTs) are used in conjunction with relations defined on tables in:

- Join condition
- Query/Form data source
- Form dyna-link
- Delete action
- Lookup

The order of relations matters when multiple relationships present between two tables: a problem by itself

- EDT relation first or table relation first
- Order by name or by AOT order

EDT relations are also used stand alone for unbound control lookup

The issues with EDT Relations are:

- There are approximately 14000 total relations to be added to tables for complete system layer migration
 - CustTableCube with the largest number of relations (68)
- 176 EDT relations need to be investigated
- Possible change of behavior when multiple relationships present between two tables
 - Relation used for joins
 - Query: can be corrected by explicit join/relation
 - Form join/dyna-link
 - Relation used for delete action.

Migration Process

Migration Steps:

- 1. Fix uniqueness issue if applicable
- 2. Copy the relation from the extended data type to all hosting tables.
 - a. Foreign key relation if possible.
 - b. Normal relation/fixed relation.
- 3. Add relational properties.
- 4. Move the EDT relation to a table reference node.

MIGRATING EXTENDED DATA TYPE RELATIONS IN MICROSOFT DYNAMICS AX 2012

- 5. Test
 - a. Test after the relation has been marked as "migrated".
 - b. Limited test if no multiple relations with other tables present
 - i. How to identify
 - c. Targeted test otherwise
 - i. Determine what to test.
 - ii. Explicitly set relationship if needed.
- 6. Check-in

EDT Relation Migration Tool

The EDT relation migration tool is used to automate the migration process.

This tool can be used to perform the following actions:

- Copy an EDT relation to all hosting tables.
- Automatically populate relation properties.
 - Derive cardinality from the index on the foreign key.
 - \circ $\;$ Derive the relationship type from the delete action/key composition.
 - Assign role names.
- Copy an EDT relation a to table reference node.
- Report artifacts impacted with before/after behavior change depending on which relation is used.
 - o Query
 - Forms

4

- Delete actions on tables
- Explicitly set the relationship to maintain existing behavior.

Using the EDT Relation Migration Tool

Open the form for the migration tool by navigating to it from **Tools > Code upgrade > EDT relation migration tool**.

Denise Juppe 12/15/2010 2:05 PM Deleted: R

Denise Juppe 12/15/2010 2:06 PM Comment [1]: What relationship?



The first time you open the tool, it will ask if you want to refresh the EDT relations data. Select **Yes** to perform this action, which takes about 5-10 minutes to complete.

EDT relation migration tool (1 - dat)												- 🗆 ×
File Refresh relation data Sca	in test ar	tifact	s Migrate multip	e tables	Migrate s	ngle table						L 😧
Table name		Log	file: <u>C:\Users\ducluor</u>	a\Microsoft\	Dynami							
AccountingDistribution			EDT relations									
AccountingDistributionTemplate												
AccountingDistributionTemplateDetail		0	Extended data	Field name	Relate	Related fiel	Has relation con	Index type	Multi rela	Migration status	Migration act	Migrate all
AccountingDistributionTmp			CurrencyCode	Transac	Currency	CurrencyCo		Primary key		Migrated to a new tabl	Skip	
AccountingDistributionTmpAmounts												
AccountingDistributionTmpJournalize												
AccountingDistributionTmpOrderLineSumm												
AccountingDistributionTmpPurchSummary												
AccountingEvent												
AccountingTmpEvent												
AddressCountryRegionBLWI												
AddressCountryRegionGroupBLWI												
AddressZipCodeImportLog_NL	_											
AgreementClassification												
AgreementClassificationTranslation			1			1	1	1				
AgreementFollowUpTmp		Þ	Relation properties									
AgreementHeader		Þ	Affected delete actio	n								
AgreementHeaderDefault		Þ	Affected forms									
AgreementHeaderDefaultHistory			Affected queries									
AgreementHeaderHistory		- P	Arrected queries									
AgreementHeaderTmp		P	Affected data sets									
AgreementLine		Þ	Affected X++ report	5								
AgreementLineDefault												
AgreementLineDefaultHistory												
AgreementLineHistory												
AgreementLineQuantityCommitment												
AgreementLineQuantityCommitmentHist												
AgreementLineReference	-											
Name.												Close

The UI of the tool is as follows:

Key to terms on the form:

The **Table name** pane on the left side of the screen displays all of the table names in the Application Object Tree (AOT).

The **EDT relations** table displays all of the EDT relations for the table selected in the left pane.

This table contains the following columns:

MIGRATING EXTENDED DATA TYPE RELATIONS IN MICROSOFT DYNAMICS AX 2012

Extended data type name: Name of the extended data type on which the field is based. **Field name**: Name of the field which is based on an EDT.

Related table name: Name of the table to which the EDT relation on the field points. **Related field name**: Name of the table to which the EDT relation on the field points. **Has relation constraints**: Indicates whether the EDT specified in the **Extended data type name** column has any 'FieldFixed' relations. When an EDT relation is migrated, the

newly created relation should also have 'FieldFixed' relations. **Index type:** indicates whether the EDT field on the primary table is part of any primary key (PK), Unique, alternate key (AK), or composite index.

Multi relations: Determines this EDT relation is one of two or more EDT relations between the table holding the field and the related table.

Migration status = Indicates the migration state. This is a read-only column and can have one of four states:

- Migrated
- NotMigrated
- MigratedToIgnoreEDTRelation
- MigratedToExistingTableRelation

Refer to a case study <u>here</u> that explains a data model in which the EDTRelation on the field is appropriate to ignore.

Migration action: A drop-down menu provides a set of available actions from which to choose. The actions are "Skip", "Migrate", "Mark as ignore EDT relation".

- Skip: To skip the migration.
- Migrate: To migrate the relation to the table relation.
- **Mark as ignore EDT relation:** To set the value of the new **IgnoreEDTRelation** property on the EDT field to **Yes**. The default value for this property is **No**.

To migrate EDT relations on a single table

- Select a table from the **Table name** pane on the left of the screen.
- Select a relation in the **EDT relations** table and choose an action from the **Migration action** drop-down menu.
- After you have set an action on all the relations for that table, click the **Migrate single table** button.

Note The migration tool tries to find a match for the EDT relation in the existing table relations. If the relation is found, the matching relation lines in the relation are marked (SourceEDT property on the relation is set to the EDT name).

If no match is found on the existing table relations, the tool will create a new relation only if the Index on the Primary table (shown by -'IndexType') is "PK or AK". If the matching Index for EDT field on Primary is NoIndex/Unique/NonUnique, the tool will not create a new relation.

6

EDT relation migration tool (1 - dat)												_ 🗆 🗵
File - Refresh relation data Sc	an test	arti	facts	Migrate multip	le tables	Migrate s	ngle table						
Table name		•	Log f	file: osoft\Dynamics	Ax\Log\EDTI	ligratior							
				EDT relations									
AccountingDistribution													
AccountingDistributionTemplate				Extended data	Field name	Relate	Related fiel	Has relation con	Index type	Multi rela	Migration status	Migration act	<u>M</u> igrate all
AccountingDistributionTemplateDetail	_			CurrencyCode	Reportin	Currency	CurrencyCo	_	Primary key		Not migrated	Skip	
AccountingDistributionTmp				CurrencyCode	Iransac	Currency	CurrencyCo		Primary key	M	Not migrated	Skp	
AccountingDistributionTmpAmounts				CurrencyCode	Account	Currency	CurrencyCo		Primary key	M	Not migrated	Mark as ignore FI	DT relation
AccountingDistributionTmpJournalize				FiscalCalendarRe	FiscalCal	HiscalCa	Recid		Primary key		Not migrated		
AccountingDistributionTmpOrderLineSumm	1												
AccountingDistributionTmpPurchSummary													
AccountingEvent													
AccountingTmpEvent													
AddressCountryRegionBLWI													
AddressCountryRegionGroupBLWI	1 1												
AddressZipCodeImportLog_NL													
AgreementClassification													
AgreementClassificationTranslation			P 1	kelation properties									
AgreementFollowUpTmp			Þ /	Affected delete actio	m								
AgreementHeader			Þ	Affected forms									
AgreementHeaderDefault			Þ	Affected queries									
AgreementHeaderDefaultHistory			Þ	Affected data sets									
AgreementHeaderTmn			Þ	Affected X++ report	s								
AgreementLine													
Agreementl ineDefault													
Agreement ineDefaultHistory													
AgreementLineHistory													
AgreementLineOuantityCommitment													
AgreementLineQuantityCommitmentHist	1	-											
The option to take when migrating this EDT re	The option to take when migrating this EDT relation												

To migrate EDT relations on multiple tables

In the left Grid, select all the tables on which you want to migrate the relation and click the "Migrate multiple tables" button.

Note In multi select mode, you cannot pick a migration action per table relation. It is "Migrate" by default and need not be set for each relation on each table. A dialog shows up saying that default action is "Migrate" and will migrate EDT relations on all the selected tables. If "Yes" is chosen on this Dialog box, then the relations are migrated, if "No" is chosen nothing will happen.

MIGRATING EXTENDED DATA TYPE RELATIONS IN MICROSOFT DYNAMICS AX 2012

EDT relation migration tool (1 - dat)	= 0 ×
File Refresh relation data Scan test artifacts Migrate multiple tables	figrate single table
Table name Log file: osoft\Dvnamics Ax\Loo\EDTMi	ration
A EDT relations	
AccountingDistribution	
AccountingDistributionTemplate	Relate Related fiel Has relation con Index type Multi rela Migration status Migration act Migrate all
AccountingDistribution TemplateDetail	skp _
AccountingDistribution TmpAmounts	
AccountingDistribution TmpJournalize	
AccountingDistribution TmpOrderLineSumm	
AccountingDistributionTmpPurchSummary	
AccountingEvent	
AddressCountryRegionBLWI	
AddressCountryRegionGroupBLWI	
AddressZipCodeImportLog_NL	
AgreementClassification	
AgreementClassification Translation	
AgreementHolder	
AgreementHeaderDefault	
AgreementHeaderDefaultHistory	
AgreementHeaderHistory	
AgreementHeaderTmp Affected X++ reports	
AgreementLine	
AgreementLineDefaultHistory	
AgreementLineHistory	
AgreementLineQuantityCommitment	
AgreementLineQuantityCommitmentHist	1
Mark all migration actions to 'Migrate'	Close

Migration Scenarios

This document explains what happens when EDT relation migration happens in different scenarios. The scenario number is as follows (the list is not complete yet).

Scenario	EDT normal relation points to Unique Key?	EDT fixed relation points to unique key	Existing table relation exactly matching EDT relation?	Existing table relation a super set of EDT relation?	Existing table relation a subset of EDT relation?
1	Yes	No	No	No	No
2	Yes	No	Yes	No	No
3	Yes	No	No	Yes	No
4	No. EDT relation does not have fixed field links but referenced field is not a unique key by itself.	Νο	Νο	No	No
5	No. EDT relation has fixed field links.	Yes	No	No	No
6	Yes. EDT relation has fixed field links	EDT fixed relation points to a super set of a unique key	No	No	No

9

In the example, we will use PKTable and FKTable with PKTable holding the referenced key and FKTable holding referencing key pointing to PKTable.

This document does not cover filling in the table relation property part, which is discussed in a separate document

(http://dynamics/functionalteams/Architecture/AX6/planvteam/relational101/Shared%20Do cuments/Dynamics%20AX%20AOT%20Table%20Relation%20Properties%20Guideline.docx).

The section under runtime behavior is for understanding how the markers are used by kernel to maintain backward compatibility and not something app developer should concern themselves with, as long as the markers are set correctly.

Scenario 1

In the example below, the EDT PKTableField1 defines a relation to 'PKTable.Field1' which by itself is an alternate key AK1. So this is a conforming relation.



<code>`FKTable.Field1'</code> uses EDT PKTableField1 which makes it a FK into PKTable but there is no table relation defined on the table.

MIGRATING EXTENDED DATA TYPE RELATIONS IN MICROSOFT DYNAMICS AX 2012



When you open the EDTMigration tool, this is what shows

EDTMigration (2 - dat)					_ _ X
File Edit View Tools Comma	nd Help				
			•		
Table List					MigrateTableRelations
Table Name	Table C	lwner		-	
FKTable	*	*			1
EKTable					
					-
	1				-
EDTRelationInfo					
FieldName EDT Name	Primary Table Name	Index Characteristic on Primary	Filters?	Pick a Migration Action	
Field1 PKTableField1	PKTable			Skip 💌	
				Skip	
				Migrate	
			լ	MarkAsMigrate	
Test Artifacts					
Delete Actions Queries Forms	Data Sets XPP Report	ts			
					1
DeleteActions:]					
Unique name for the ID					SYS Model dat sys 🛢 🛛 🏒

In the 'Pick a Migration Action' column, pick Migrate and click the 'Migrate Table Relations' button on the upper right corner. The migration will

- (1) Create a new relation PKTable of type 'Normal' (as the Key is not PK).
- (2) The EDTRelation property it set to 'true' as this relation is added due to EDT relation migration.
- (3) The relation will have one field link.
- (4) The field link will have SourceEDT set to EDT PKTableField1.

Runtime behavior

- (1) APIs that uses EDT relation first on 'FKTable.Field1' now will be able to find the same relation with the same field link under PKTable relation by examining the SourceEDT flag.
- (2) APIs that uses table relation first will not pick up the PKTable relation first as it is flagged as EDTRelation.

MIGRATING EXTENDED DATA TYPE RELATIONS IN MICROSOFT DYNAMICS AX 2012



MIGRATING EXTENDED DATA TYPE RELATIONS IN MICROSOFT DYNAMICS AX 2012



Scenario 2

This scenario is similar to scenario 1. The 'FKTable.Field2' holds the extended data type PKTableField1 as a foreign key into PKTable. However, there is already an existing table relation PKTable2 defined on the table with exactly the same field link.

MIGRATING EXTENDED DATA TYPE RELATIONS IN MICROSOFT DYNAMICS AX 2012



The migration will

- (1) Not create a new relation as there is already an existing table relation that matches the EDT relation.
- (2) The EDTRelation property remains 'No' as this relation is not added due to EDT relation migration.
- (3) The field link will have SourceEDT set to EDT PKTableField1 to indicate that this matches the EDT relation defined on EDT PKTableField1.

Runtime behavior

- (1) APIs that uses EDT relation first on 'FKTable.Field2' now will be able to find the same relation with the same field link under PKTable2 relation by examining the SourceEDT flag.
- (2) APIs that uses table relation first will still pick up the PKTable2 relation first as it is not flagged as EDTRelation.

16

Scenario 3

Scenario 3 is similar to scenario 2, the 'FKTable.Field3' uses EDT PKTableField1 which makes it a FK into PKTable but there is an existing table relation PKTable3 defined already on the table but the relation has more field links defined on it than the EDT relation. In the example shown below, Field4 in FKTable is duplicated (denormalized) from PKTable Field2. But it could also be a related fixed field link.

Project EDTMigration	_ _ 	R	elation PKTable3	×
🖻 🖆 🥔 🍝 🐺 🔛			Properties Categories	
EDTMigration(sys)			Name	PKT able3
🗉 🧾 PKTable(sys)			Table	PKTable
😑 🚍 Fields			Validate	Yes
Field2(sys)			EntityRelationshipRole	
🔄 🤤 Field1(sys)			RelatedTableCardinality	NotSpecified
🛨 🔝 Field Groups			Cardinality	NotSpecified
🖃 🖵 Indexes			RelationshipType	NotSpecified
🗆 🖳 AK1(sys)			RelatedTableRole	
🔤 Field1			Role	
🗉 🚞 Full Text Indexes			EDTRelation	No
🗖 🗖 🖙 Relations			ModelName	SYS Model
🛨 🌆 DeleteActions				
🔳 🗉 💐 Methods				
🗆 🖽 FKTable(sys)				
🖻 🚍 Fields				
Field4(sys)				
Field3(sys)				
Field2(sys)				
Field1(sys)				
🖃 🔄 Field Groups				
🗉 🖽 AutoReport				
🖿 💷 AutoLookup				
🖽 🖾 AutoSummary(sys)				
🗉 🖭 AutoIdentification(sys)				
🔲 🗆 🕎 Indexes				
🗖 🗖 🔁 Full Text Indexes				
🗖 🖃 🏹 Relations				
PKTable3(sys)				
📕 🔤 FKTable.Field4 == PKTable.Field2				
FKTable.Field3 == PKTable.Field1	-			
			1	

The migration will

- (1) Not create a new relation as there is already an existing table relation that covers the EDT relation field link.
- (2) The EDTRelation property remains 'No' as this relation is not added due to EDT relation migration.

MIGRATING EXTENDED DATA TYPE RELATIONS IN MICROSOFT DYNAMICS AX 2012

- (3) The field link that matches EDT relation will have SourceEDT set to EDT PKTableField1 to indicate that this matches the EDT relation defined on EDT PKTableField1.
- (4) The field link that does not match EDT relation will have SourceEDT set o empty as it does not match the EDT relation field link.

Runtime behavior

- (1) APIs that uses EDT relation first on 'FKTable.Field2' now will be able to find the same relation with the same field link under PKTable2 relation by examining the SourceEDT flag and taking only field links that have a matching SourceEDT.
- (2) If this field link is deleted, it will then look up relation under EDT.
 - a. If EDT relation is removed, then the lookup field is used if the lookup is a result of EDT migration (we need a flag there as well?).
- (3) APIs that uses table relation first will still pick up the PKTable2 relation first as it is not flagged as EDTRelation.

Scenario 4

There are two cases for this:

Case	Is the referenced field part of an alternate key?	
1	Yes	
2	No	

Case 1

In the example below, EDT PKTableField2 defines a relation to `PKTable.Field2' which by itself is not a unique key but belongs to AK2 which includes Field2 and Field3.

📴 Project EDTMigration 📃 🗖 🕽	Table FKTable	
😂 🖀 🥔 🏅 🍞 🕮	Properties Categories	
EDTMigration(sys)		6967
🖻 📰 PKTable(sys)	Name	FKTable
🗉 🚍 Fields	Label	
표 📷 Field Groups	FormBef	
🖃 🕎 Indexes	ListPageRef	
🗆 📕 AK2(sys)	ReportRef	
(🔄 Field3)	PreviewPartRef	
📲 Field2	SearchLinkRefType	Url
🗄 🚛 AKI(sys)	SearchLinkRefName	
🗉 🚞 Full Text Indexes	TitleField1	
🗆 🖙 Relations	TitleField2	
🖽 🌠 DeleteActions	TableType	Regular
🖽 📝 Methods	TableContents	Not specified
🗄 🏢 FKTable(sys)	Systemtable	No
🗉 🎫 PKTableField1(sys)	ConfigurationKey	
🗉 🃷 PKTableField2(sys)	SecurityKey	
🗉 📳 Array Elements	Visible	Yes
🖻 🖙 Relations	AOSAuthorization	None
PKTableField2 == PKTable.Field2	CacheLookup	None
🛨 🛋 Table References	CreateRecIdIndex	Yes
	SaveDataPerCompany	Yes
	TableGroup	Miscellaneous
	PrimaryIndex	SurrogateKey
	ClusterIndex	SurrogateKey
	NaturalKey	
	AnalysisVisibility	Auto
	AnalysisSelection	Auto
	TypicalRowCount	Auto
	IsLookup	No
	AnalysisDimensionType	Auto

FKTable.Field5 uses PKTableField2 but there is no existing relation on FKTable that contains Field5.

The developer can do one of the two things

(1) If there is a relationship from FKTable to PKTable

- a. If the corresponding foreign key on FKTable that points to PKTable.Field3 already exists, create a relation with that field together with FKTable.Field5.
- b. If not, this indicates the data model is wrong. Create an additional field on FKTable and create a relation with that field together with FKTable.Field5. You may need upgrade script to populate the added fields.
- (2) If there is no relationship, ignore the EDT relation.

Case 2

For 1.2, the migration will

(1) Create a new relation which includes two field links as mentioned above.

19

- (2) The EDTRelation property set to 'Yes' as this relation is added due to EDT relation migration.
- (3) The field link that matches EDT relation will have SourceEDT set to EDT PKTableField2 to indicate that this matches the EDT relation defined on EDT PKTableField2.
- (4) The field link that does not match EDT relation will have SourceEDT set o empty as it does not match the EDT relation field link.

Runtime behavior

- (1) APIs that uses EDT relation first on 'FKTable.Field5' now will be able to find the same relation with the same field link under PKTable5 relation by examining the SourceEDT flag.
- (2) APIs that uses table relation first will no pick up the PKTable2 relation first as it is flagged as EDTRelation.

The following shows the result of 1.a.



For scenario 2, in the action option for this EDT relation, pick 'Mark Ignore EDT Relation' (currently labeled 'Marked as migrated').

The change below is still going through review and not implemented yet.

20

The migration will

- (1) Will not create a new relation
- (2) FKTable.Field5 'IgnoreEDTRelation' is set to TRUE.

Runtime behavior

(1) APIs that uses EDT relation first on 'FKTable.Field5' now will have to be divided into two groups

- a. Those that require relational semantics, like joins, delete action etc will not use the EDT relation as it is marked as IgnoreEDTRelation.
 - b. Those that does not, like lookup, will still see EDT relation.

Case 5 and 6

EDT relation has fixed field links in both these two scenarios.

In scenario 5, the entire set of referenced field makes up a unique key. However, only one of the referencing fields is present in the referencing table. Syntactically, this is not a relationally correct FK relationship because the complete set of FK is not available. But because the rest of the FK values will all be constant (and thus redundant) they are simply omitted from the table and expressed in the fixed field relation.

In scenario 6, the fixed field links serve as a filter condition on top of the relation expressed by the normal field link.

The migration tool can handle both scenarios and create new normal table relations, with `EDT Relation' property set to TRUE on the new relation and `SourceEDT' set to the EDT on all of the field links, including the fixed field links.

SCIVE	PMC .	
HQ Database	RMSHQSample	
Accounts Inv	entory Tax Defaults	
RMS Department U	ode Description	GP Item Class ID Q Description
100	Infant	BABY Baby Loys
200	Action	ACTION Action Figures
300	Dolls	DOLLS Dolls
400	Sporting Goods	SPORT Sporting Goods
500	Remote Control	REMOTE Remote Control Toys
900	Misc	MISC Miscellaneous

The following is a sample table. **To use this format throughout the document, copy and paste the table, and add, resize, and delete rows and columns as needed.** For text in the table, use the defined Table Heading and Table Text paragraph formats. A 1½-pt. vertical white line separates columns. Add a blank Spacing paragraph after each table, as is done in the example below.

MIGRATING EXTENDED DATA TYPE RELATIONS IN MICROSOFT DYNAMICS AX 2012

Table Heading	Table Heading
Table Text	Table Text
Table Text	Table Text
Table Text	Table Text • Table Text Bullet • Table Text Bullet

Note: The Table Heading, Table Text, and Table Text Bullet formats have built-in left and right indents. The table cell properties specify no additional indent. Align tables flush with the left margin of the text immediately above it. Do not center tables, and make sure tables don't overlap the right text margin. To show the text margins, choose **Tools > Options**. On the **View** tab, mark the **Text Boundaries** box.

Formatting code and scripts [Heading 1]

The following is a sample of code. Format all code or scripts with the "Code" paragraph format, which uses the monospaced Courier New font. There is no indented version of this style, because it contains indents every .25 inch. Just use tabs to add indents if the code appears below an indented text block. To set the code off from text above and below it, just use blank paragraph returns.



MIGRATING EXTENDED DATA TYPE RELATIONS IN MICROSOFT DYNAMICS AX 2012

Microsoft

Margo Crandall 12/10/2010 10:03 AM Comment [2]: This is the pre-release version of the copyright.

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 supports a preliminary release of a software product that may be changed substantially prior to final commercial release. This document is provided for informational purposes only and Microsoft makes no warranties, either express or implied, in this document. Information in this document, including URL and other Internet Web site references, is subject to change without notice. The entire risk of the use or the results from the use of this document remains with the user. Unless otherwise noted, the companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted in examples herein are fictitious. No association with any real company, organization, product, domain name, e-mail address, logo, person, place, or event is intended or should be inferred. Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

 $\ensuremath{\mathbb{C}}$ <current year of release> Microsoft Corporation. All rights reserved.

Microsoft <list other MS trademarks used alphabetically> are trademarks of the Microsoft group of companies.

<This is where mention of specific, contractually obligated to, third party trademarks should be listed.>

All other trademarks are property of their respective owners.



24