MULTISITE

Microsoft Dynamics® AX 2009

Multisite Activation

White Paper

Date: June 2008

Microsoft Dynamics

Introduction 3

Multisite activation 4

Prerequisites 4

Multisite activation update jobs 5

Multisite activation consistency check 6

Inventory dimensions update 7

Update patterns and macros 13

Warehouse items update 14

Inventory transactions update 16

Purchase lines update 19

Updating historical data 23

Introduction

Microsoft Dynamics 2009 includes new multisite activation functionality that differs from previous versions. The multisite activation process is divided into two parts. This ensures that the activation runs as quickly as possible and with as little down time for a company as possible. The first part is the activation itself, where data is updated so that the company can continue its daily work once multisite is activated. The second part is the update of the historical data, which does not impact the daily work, and can be performed after the initial multisite activation is finished. Customers may want to update the historical data because inconsistencies may arise in data when the inventory dimensions are updated.

For customers who have a new installation of Microsoft Dynamics AX 2009, the multisite functionality is activated by default.

When upgrading from a previous release of Microsoft Dynamics AX (formerly known as Microsoft Business Solutions–Axapta), the new multisite functionality is initially inactive. This allows customers to set up the master data required for the multisite functionality and correct any data inconsistencies before generating any site-specific transactions. The process of upgrading to Microsoft Dynamics AX 2009 and activating the multisite functionality is illustrated in **Figure 1** below.

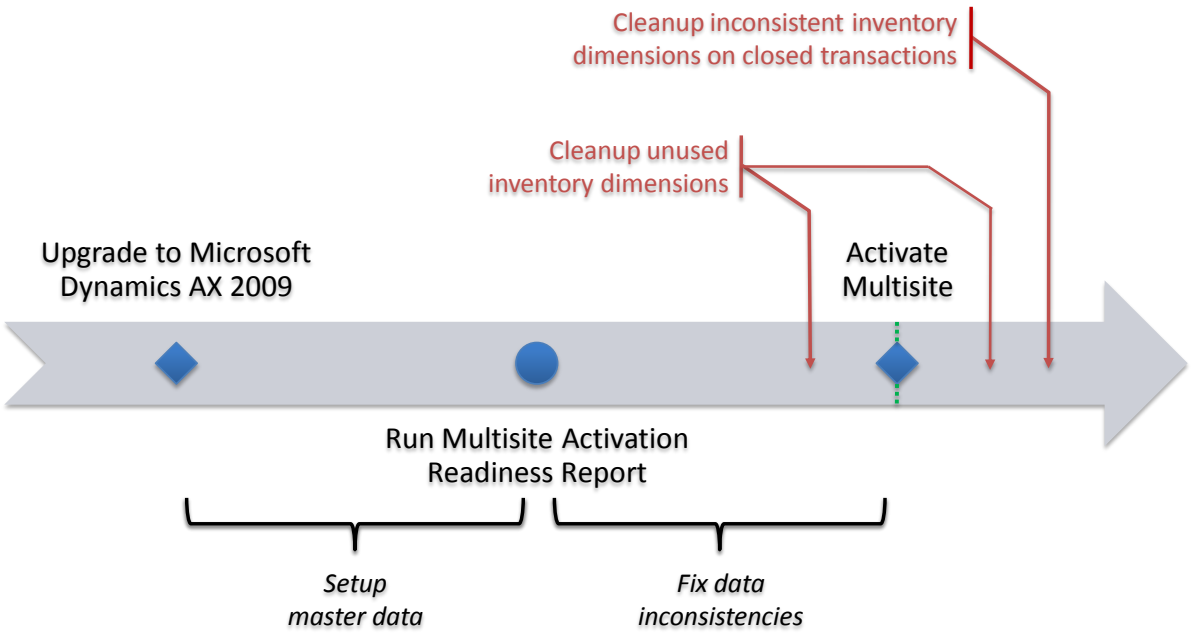


Figure 1 Multisite activation timeline

To start using the new multisite functionality, customers upgrading from a previous release must run the multisite activation wizard for each company account to make the necessary changes to the inventory dimension settings and the inventory related transactions. These changes include:

- Activating the site and warehouse inventory dimensions in all dimension groups
- Making the site dimension mandatory on all inventory transactions
- Updating all transactions with an association to a site

After the multisite functionality has been activated, the site dimension becomes mandatory, which ensures that a site value is entered on all inventory transactions.

The multisite activation process has been optimized for performance, and therefore, some historical data will not be updated. The resulting inconsistencies in the historical data can be corrected using the **Multisite activation – dimension inconsistency clean up** job (**Administration** > **Periodic** > **System** > **Multisite activation – dimension inconsistency clean up**).

Multisite activation

The multisite activation wizard is responsible for updating records in a Microsoft Dynamics AX 2009 company, so the records are activated for multisite. This is required before those records can be used in a company that is activated for multisite.

In most cases, the activation ensures that each record that uses inventory dimensions is associated with a site. If a warehouse is already specified in the record, then the site that is associated with that warehouse is assigned to the record. If a warehouse is not specified in the record, then either the default site is applied, or both the default site and warehouse are applied. In some cases a site is not applied, and then both the site and warehouse remain blank.

Prerequisites

The multisite activation process is designed to run as quickly as possible. A prerequisite for running the activation is that no inventory dimension record can reference a value for both the site and warehouse. All other dimension combinations are allowed.

The four possible combinations of warehouse and site are shown in **Table 1** below. The table shows the combinations that are allowed and the one combination that is not allowed.

Site / Warehouse	Blank	Value
Blank	✓	✓
Value	✓	✗

Table 1 Allowed combinations of warehouse and site

Technically this means that the inventory dimensions (InventDim) table may not contain records where both the site and warehouse are specified as shown for the highlighted record with Dim. No. 4 in **Table 2** below.

Inventory Dimensions				
Dim. No.	Configuration	Site	Warehouse	…
1				
2			Mw	
3		Hq		
4		Hq	Mw	

Table 2 Example inventory dimensions

When multisite is not activated, a record will be created for either site or warehouse, but not both inventory dimensions. If a record has both site and warehouse inventory dimensions, then the multisite activation cannot be processed.

If an inventory dimension record references both a site and warehouse value for a company that does not have multisite activated, the company must have been customized. You must customize the multisite activation wizard to handle the scenario.

If an inventory dimension references both a site and warehouse value for an unknown reason and the inventory dimension is not referenced by any other table in the application, you can use the **Clean up unused inventory dimensions** form (**Inventory management** > **Periodic** > **Clean up** > **Inventory dimensions clean up**) to delete the unused inventory dimensions. You must complete this process before you can activate the multisite functionality.

Multisite activation update jobs

The multisite activation process is divided into a number of update jobs that deal with all or some of the records in a table. Because the updates of records in some tables depend on

already updated records in other tables, the activation implements dependencies between the individual update jobs.

Figure 2 below shows an example of these dependencies. For example, the records in the Warehouse items table and Price agreements table are dependent on the update of the inventory dimensions.

Each update job is implemented as an instance method on the *InventSiteActivateUpd* class. The dependencies between the update jobs are implemented in the *updateBatch* and *updateSequential* instance methods on the *InventSiteActivateUpdMgr* class. The *updateBatch* method is used when the multisite activation is processed in batch, and the individual update jobs are then processed in parallel by the batch server. The *updateSequential* method is used when the activation is not processed in batch, so each update job is processed sequentially.

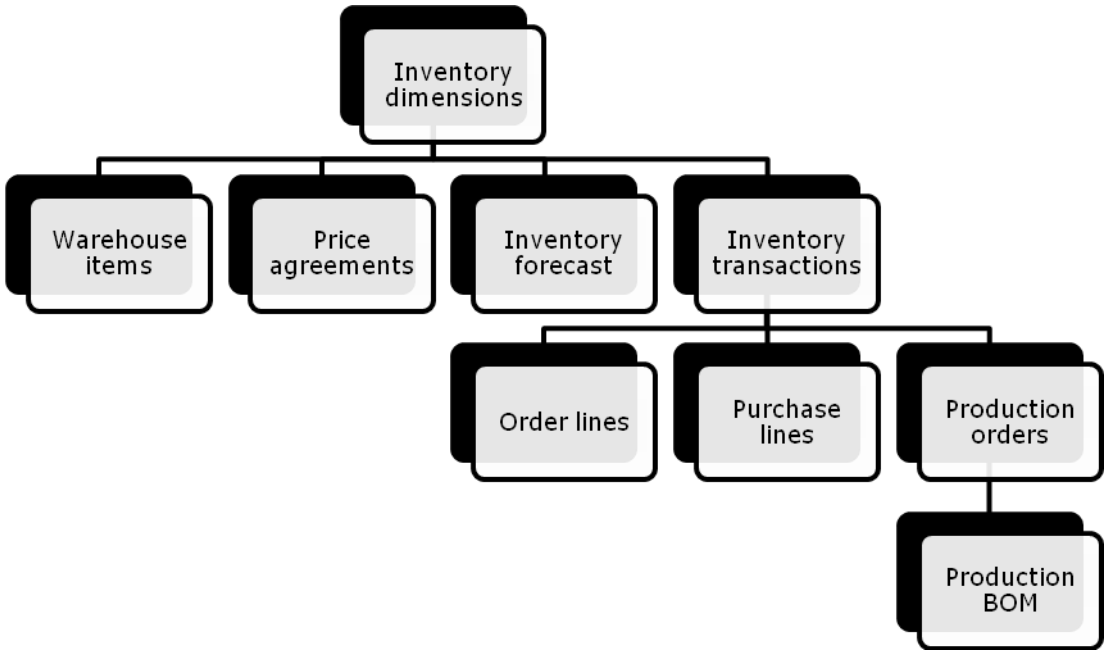


Figure 2 Update job dependencies

Multisite activation consistency check

Records in the table will be checked for consistency and updated, if necessary. If the consistency check fails, the multisite activation cannot be started.

The consistency check searches all tables for fields of the type *InventDimId*, or a type that is derived from *InventDimId*, to determine whether the table needs to be updated by the multisite activation. The consistency check uses two methods on the *InventSiteActivateDimFieldsCheck* class to execute the search. The two methods are:

- The *updateableFields* method creates a list of tables that need to be updated.
- The *notUpdatableFields* method creates a list of tables that do not need to be updated.
- If a table does not appear on either of these lists, the consistency check will fail.

When a new field type *InventDimId* is added to the Microsoft Dynamics AX 2009 application – either as part of a new table or as an addition to an existing table – then the table that contains the new field must be added to either of the two lists in the *InventSiteActivateDimFieldsCheck* class. If the records in the table also need to be updated by the multisite activation, an update job that contains the update logic must also be added to the activation logic.

For security reasons, the multisite activation does not report the unhandled table(s) when the consistency check fails. The table(s) that cause the failure can be found by executing a Best practice check (**Specific Checks** > **Tables** > **Inventory dimension fields are handled by multisite activation** check enabled) for all tables in the system.

Note: Remember to run the Best practice check against all tables in the application. *InventDimId* fields are widely used, and experience shows that unhandled *InventDimId* fields are often found in unexpected places

After the multisite activation has been customized to handle the tables that fail, the multisite activation can be started again.

Inventory dimensions update

One of the first update jobs that are processed by the multisite activation is the inventory dimensions update job shown in **Figure 2**.

To understand how the multisite activation process works, it is essential to understand how the inventory dimensions are updated and how this update impacts other tables that reference inventory dimensions.

Inventory dimensions are considered to be mandatory in Microsoft Dynamics AX 2009. This means that if a table references a set of inventory dimension values and the referencing table is updated to reference a different set of inventory dimension values, the values in the referenced inventory dimension record are not updated. Instead, the application attempts to find an already existing record that matches the new values completely. If such a record does not exist, then a new record is created, and the referencing table is updated to reference the new inventory dimension record.

However, using this logic when activating multisite would affect performance and the activation would not finish in a timely manner. Therefore, the logic that is used in the multisite activation updates the inventory dimensions table directly by assigning a site value to all records where a warehouse is already specified. Those records where both site and warehouse are blank are assigned a default site and default warehouse. This update logic ensures that the content of the majority of the tables that reference inventory dimensions are updated correctly.

The updated records in other tables that reference inventory dimensions with a blank site and/or warehouse will be corrected when the update job for the specific table is processed at a later stage during the multisite activation. At that time, the update job will create the necessary new inventory dimensions and redirect the records in the specific table to the newly created inventory dimensions.

The logic that is described above and the state of the inventory dimensions after multisite has been activated is illustrated by the following example:

Assume that the site and warehouse definitions that are shown in **Figure 3** below, where the Mw and Aw warehouses are assigned to site Hq, and site Sub only contains a single warehouse named Lw.

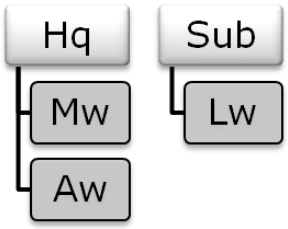


Figure 3 Multisite inactive site and warehouse definitions

As part of the multisite activation setup, the user entered Bw as the default warehouse. This default warehouse will be created as a new warehouse that is assigned to the specified default site (Hq). After the activation, the structure of the sites and warehouses will be as shown in **Figure 4** below.

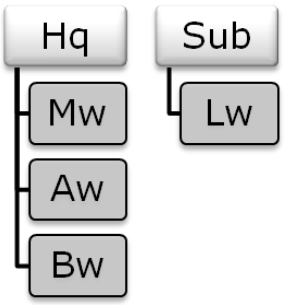


Figure 4 Multisite active site and warehouse definitions

The site and warehouse structure will be used when updating the inventory dimensions.

Inventory Dimensions				
Dim. No.	Configuration	Site	Warehouse	…
1000				
1010			Mw	
1020		Hq		
1030	Blue		Mw	
1040	Red	Hq		
1050	Blue			
1060	Red		Mw	
1070			Lw	
1080			Aw	

Figure 5 Multisite inactive inventory dimensions

Assuming that the inventory dimensions table contained the records shown in **Figure 5** above when multisite was inactive.

The inventory dimensions update job will update the content of the table as shown in **Figure 6** below.

Inventory Dimensions				
Dim. No.	Configuration	Site	Warehouse	…
1000		Hq	Bw	
1010		Hq	Mw	
1020		Hq		
1030	Blue	Hq	Mw	
1040	Red	Hq		
1050	Blue	Hq	Bw	
1060	Red	Hq	Mw	
1070		Sub	Lw	
1080		Hq	Aw	

Figure 6 Multisite updated inventory dimensions

The inventory dimensions update job is implemented in the *updateInventDim* instance method on the *InventSiteAcitvateUpdate* class. This method calls the *updateInventDim*

instance method on *InventSiteActivateUpdRuleInvLoc*, where the method again calls other protected methods which utilize the *UpdateSiteFromLocation* macro. There is more on the *InventSiteActivateUpdRuleInvLoc* class and macros later in this document.

The update job results in the following:

- Records that reference a warehouse value are updated with the associated site.
- Records that reference neither a warehouse nor a site are updated with the default site and default warehouse values.
- Records that only reference a site value but not a warehouse value are not updated.

As mentioned earlier, some records must still reference inventory dimension records with no site and/or warehouse value. During the update of these tables, new inventory dimensions will be created. After multisite has been activated the content of the inventory dimensions table may look like the content shown in **Figure 7** below.

Inventory Dimensions				
Dim. No.	Configuration	Site	Warehouse	…
1000		Hq	Bw	
1010		Hq	Mw	
1020		Hq		
1030	Blue	Hq	Mw	
1040	Red	Hq		
1050	Blue	Hq	Bw	
1060	Red	Hq	Mw	
1070		Sub	Lw	
1080		Hq	Aw	
Ax#1				
Ax#2	Blue			
Ax#3	Blue		Mw	
Ax#4			Mw	
Ax#5	Red		Mw	
Ax#6			Lw	
Ax#7			Aw	
Ax#8	Blue	Hq		
Ax#9		Sub		

Figure 7 Multisite activated inventory dimensions

Notice that the Dimension No. for the new records is prefixed Ax#. The Ids for these records are not pulled from the inventory dimensions number sequences for performance reasons.

The Ax# prefix ensures that assigned Dimensions No. values will not conflict with any previously assigned values that are pulled from the number sequence. The unique index on the Dimensions No. field should not be changed.

The following figures describe examples of a scenario where some records in a table must be updated to reference inventory dimensions without a site and warehouse.

Figure 8 below shows records in a table that reference inventory dimensions when multisite is inactive. In this state, the referencing table may reference inventory dimensions where item dimensions are entered or blank. It may also reference inventory dimensions where warehouse is specified or blank. Site and the remaining dimension must be blank in this example. When multisite is active, then both site and warehouse must be either specified or left blank. The remaining dimensions should still be blank.

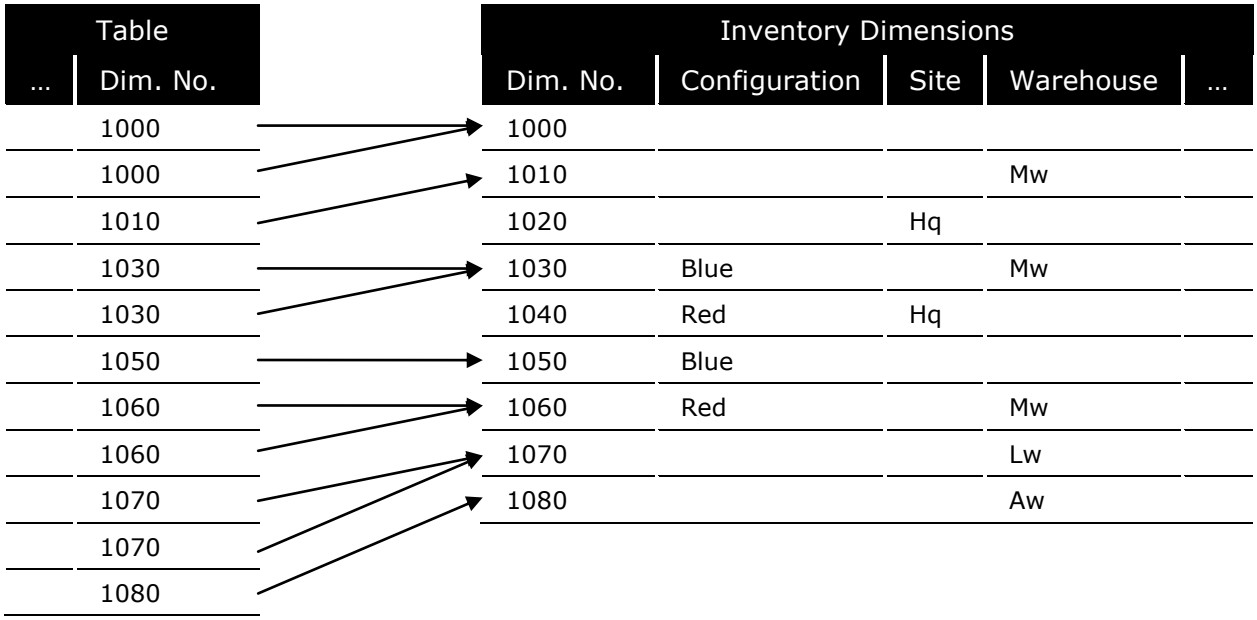


Figure 8 Multisite inactive inventory dimensions

After the inventory dimensions table has been updated by the inventory dimensions update job, then the content of the inventory dimensions table will be as shown in Figure 9 below. The records that previously referenced the inventory dimensions with Dim. No. 1000 and 1050 now reference inventory dimensions with the default site Hq and the default warehouse Bw.

When multisite is activated, these records should still reference inventory dimensions with blank site and blank warehouse.

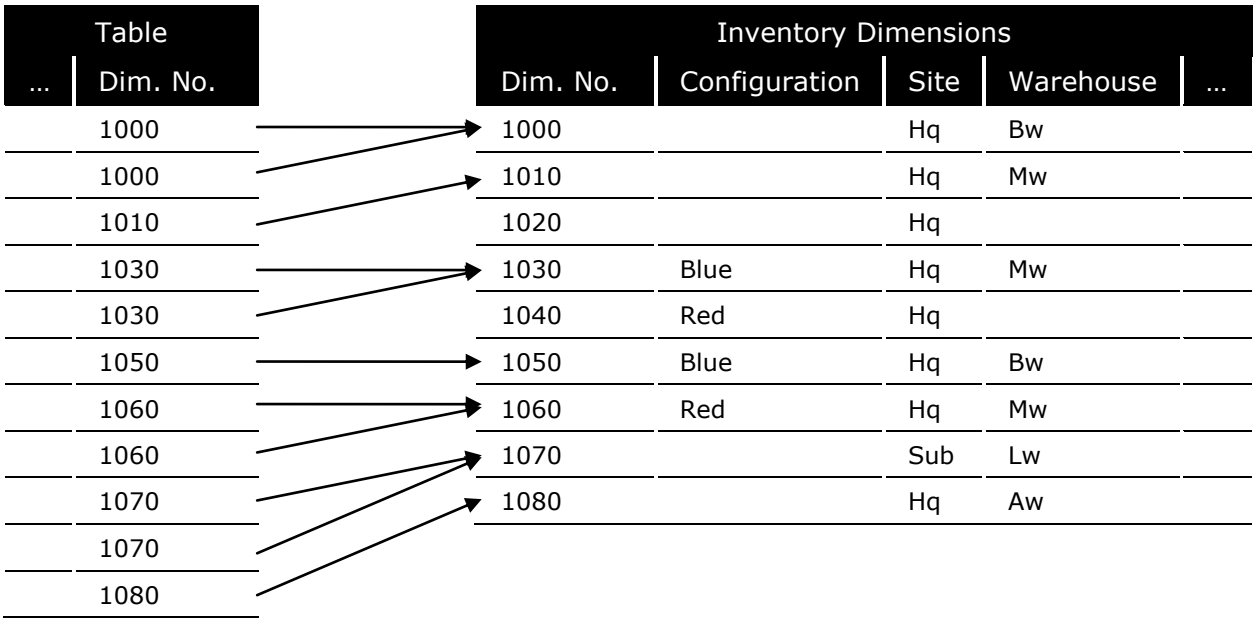


Table	
…	Dim. No.
	1000
	1000
	1010
	1030
	1030
	1050
	1060
	1060
	1070
	1070
	1080

Inventory Dimensions				
Dim. No.	Configuration	Site	Warehouse	…
1000		Hq	Bw	
1010		Hq	Mw	
1020		Hq		
1030	Blue	Hq	Mw	
1040	Red	Hq		
1050	Blue	Hq	Bw	
1060	Red	Hq	Mw	
1070		Sub	Lw	
1080		Hq	Aw	

Figure 9 Multisite updated inventory dimensions

The redirection of the records that now reference the default site and warehouse is handled by the update job for the specific table. If inventory dimensions, which match the current record and also contain a blank site and warehouse, do not exist, then the update job creates the missing inventory dimensions.

When multisite has been activated, the content of the table and the inventory dimensions table may look like the data shown in **Figure 10** below.

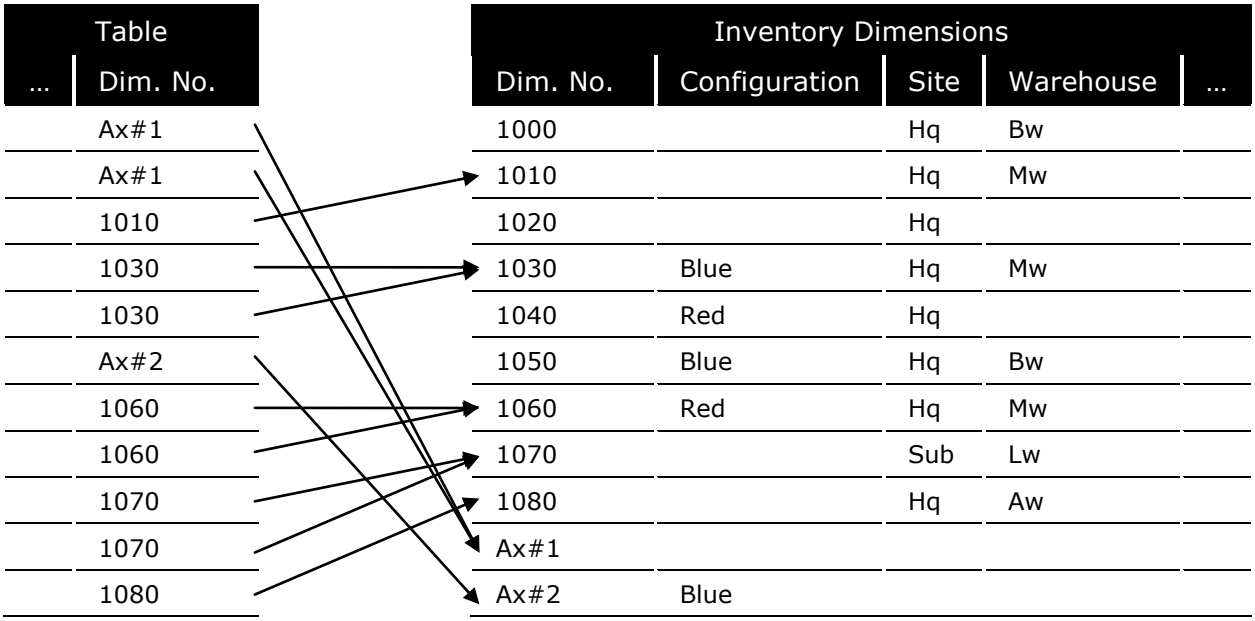


Table	
…	Dim. No.
	Ax#1
	Ax#1
	1010
	1030
	1030
	Ax#2
	1060
	1060
	1070
	1070
	1080

Inventory Dimensions				
Dim. No.	Configuration	Site	Warehouse	…
1000		Hq	Bw	
1010		Hq	Mw	
1020		Hq		
1030	Blue	Hq	Mw	
1040	Red	Hq		
1050	Blue	Hq	Bw	
1060	Red	Hq	Mw	
1070		Sub	Lw	
1080		Hq	Aw	
Ax#1				
Ax#2	Blue			

Figure 10 Multisite activated inventory dimensions

Update patterns and macros

As discussed previously in this document, the *InventSiteActivateUpdate* class contains a method for each table and the method updates the contents of the table as part of the multisite activation.

The methods do not, however, contain the actual update logic. This logic is implemented in an *InventSiteActivateUpdRule* class hierarchy with a base class and three subclasses.

The majority of the update jobs are written using SQL. None of the X++ inserts and update operations are used. The base class *InventSiteActivateUpdRule* contains logic to help implement and format the SQL statements correctly, so that the statements can be processed on all Microsoft Dynamics AX 2009 supported databases, without the developer having to worry about conversion of application table names and field names into the database table names and column names.

The *InventSiteActivateUpdRuleInvLoc* subclass contains update logic to:

- Update site fields based on the content of the warehouse field.
- Redirect records to inventory dimensions where default site and/or default warehouse is blank.

The *InventSiteActivateUpdRuleInvTrans* subclass contains update logic to:

- Redirect inventory movement to inventory dimensions where default warehouse is blank based on the inventory transactions related to the inventory movement.

The *InventSiteActivateUpdRuleMisc* subclass contains update logic that could not be generalized and implemented in the previous two subclasses.

Much of the update logic follows the same patterns and is implemented using X++ macros to provide as much compile checking as possible on the code that generates the SQL statements.

The following information provide examples of how different tables in Microsoft Dynamics AX 2009 are updated with descriptions of data in the inactive, intermediate, and active multisite statuses.

Warehouse items update

Warehouse items must only reference inventory dimensions when item dimensions are potentially specified and warehouse is potentially specified. Site and the remaining dimension must always be blank. This is true when multisite is active and when multisite is inactive. An example of the content of the Warehouse items and inventory dimensions tables when multisite is inactive is shown below in **Figure 11**.

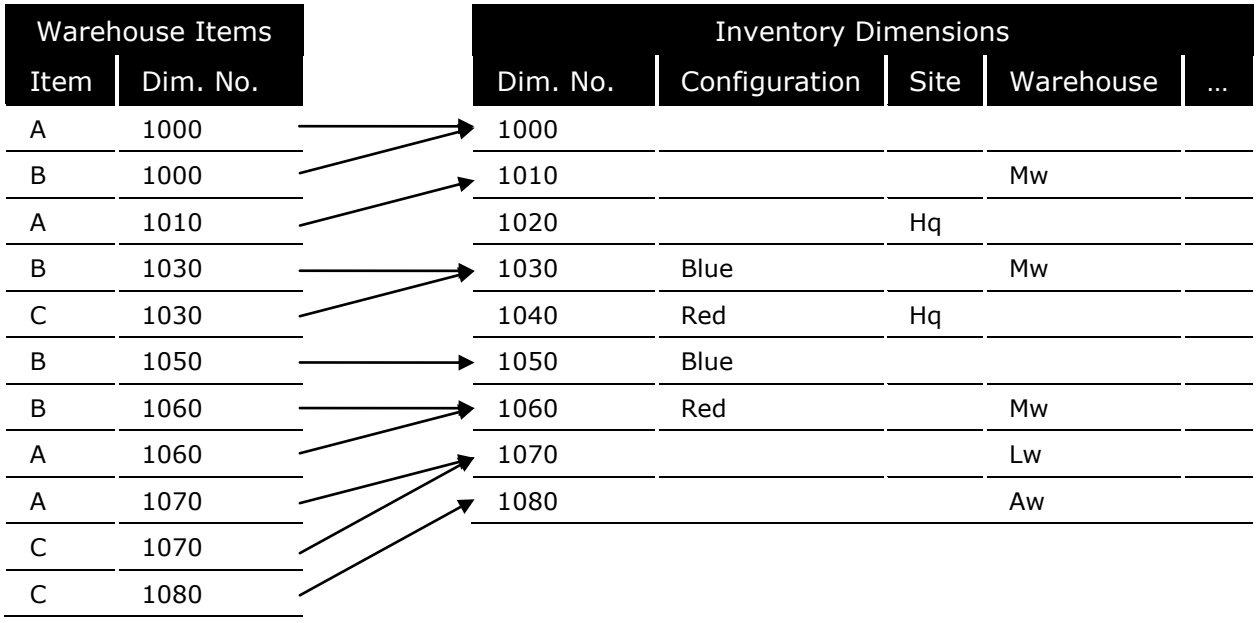
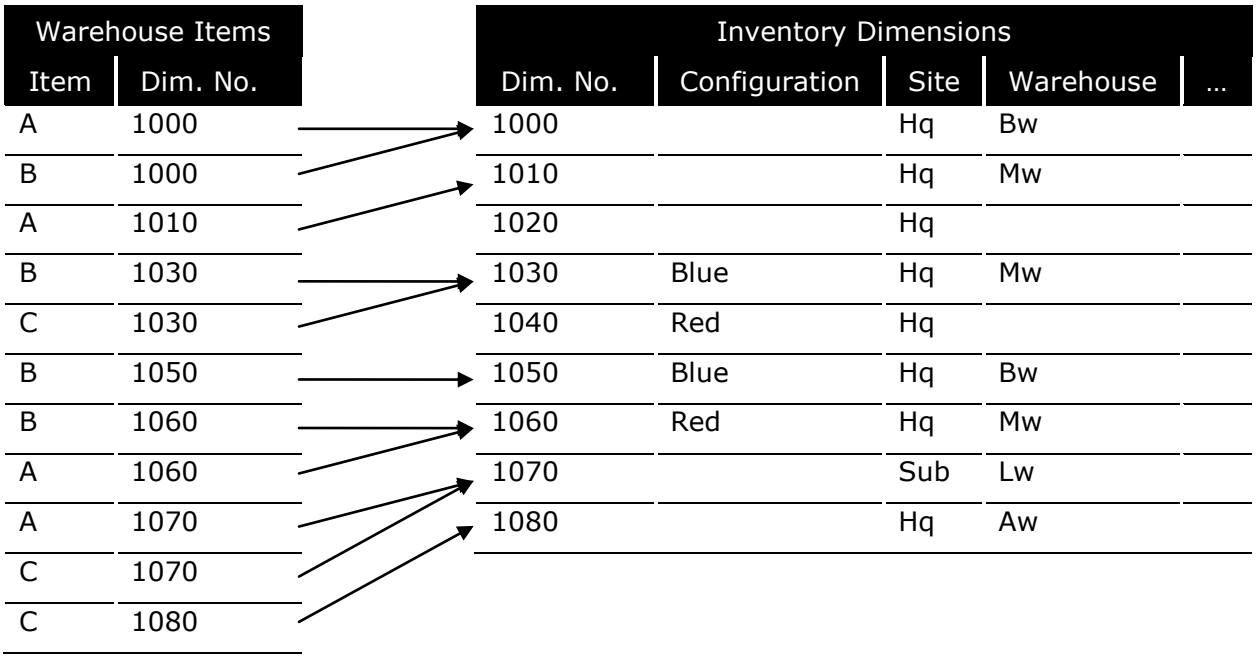


Figure 11 Multisite inactive warehouse items

When the inventory dimensions have been updated by the multisite activation, all the inventory dimensions that reference warehouse items will contain a site value and a warehouse value. The content is shown below in **Figure 12**.



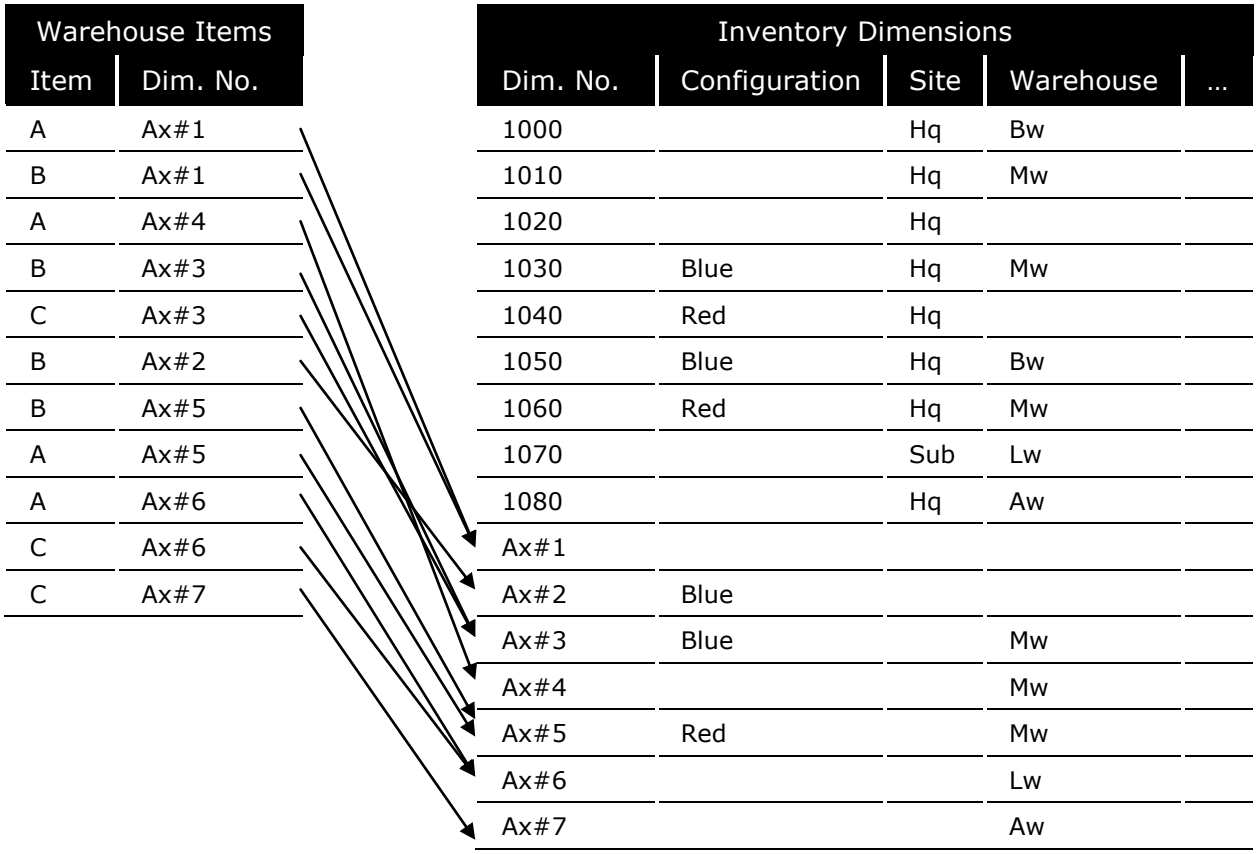
Warehouse Items	
Item	Dim. No.
A	1000
B	1000
A	1010
B	1030
C	1030
B	1050
B	1060
A	1060
A	1070
C	1070
C	1080

Inventory Dimensions				
Dim. No.	Configuration	Site	Warehouse	…
1000		Hq	Bw	
1010		Hq	Mw	
1020		Hq		
1030	Blue	Hq	Mw	
1040	Red	Hq		
1050	Blue	Hq	Bw	
1060	Red	Hq	Mw	
1070		Sub	Lw	
1080		Hq	Aw	

Figure 12 Multisite updated Warehouse items (intermediate results)

When the job that updates the Warehouse items table is processed, it first updates the Warehouse items that reference the default site and default warehouse. These are redirected to reference inventory dimensions with a blank site and blank warehouse. Afterwards, the Warehouse items that reference a site value are redirected to inventory dimensions without a site.

The results of these two updates are shown in **Figure 13**. The update job for the Warehouse items table is implemented in the updateInventItemLocation instance method on the InventSiteActivateUpdate class. This method calls the updateInventItemLocation instance method on the InventSiteActivateupdRuleInvLoc class. This method, in turn, calls two methods that utilize two macros: BlankDefaultLocAndDefaultSiteOnInventDim and BlankSiteOnInventDim. BlankDefaultLocAndDefaultSiteOnInventDim updates the Warehouse items that refer to the default site and the default site warehouse and redirects them to records where site and warehouse are blank. BlankSiteOnInventDim updates the Warehouse items that refer to inventory dimensions that contain a site value and redirects them to records where only site is blank.



Warehouse Items	
Item	Dim. No.
A	Ax#1
B	Ax#1
A	Ax#4
B	Ax#3
C	Ax#3
B	Ax#2
B	Ax#5
A	Ax#5
A	Ax#6
C	Ax#6
C	Ax#7

Inventory Dimensions				
Dim. No.	Configuration	Site	Warehouse	…
1000		Hq	Bw	
1010		Hq	Mw	
1020		Hq		
1030	Blue	Hq	Mw	
1040	Red	Hq		
1050	Blue	Hq	Bw	
1060	Red	Hq	Mw	
1070		Sub	Lw	
1080		Hq	Aw	
Ax#1				
Ax#2	Blue			
Ax#3	Blue		Mw	
Ax#4			Mw	
Ax#5	Red		Mw	
Ax#6			Lw	
Ax#7			Aw	

Figure 13 Multisite activated Warehouse items

Inventory transactions update

When multisite is activated, inventory transactions must always refer to an inventory dimension where site is specified.

When multisite is inactive, the inventory transaction should refer to an inventory dimension with blank site and warehouse. When multisite is active, the inventory transaction should refer to the default site and default warehouse. An exception applies for transactions that are not issued or received. These transactions should only refer to the default site and a blank warehouse, because warehouse is not required if the transaction has not been issued or received.

Figure 14 below shows an example of inventory transactions and inventory dimensions when multisite is inactive.

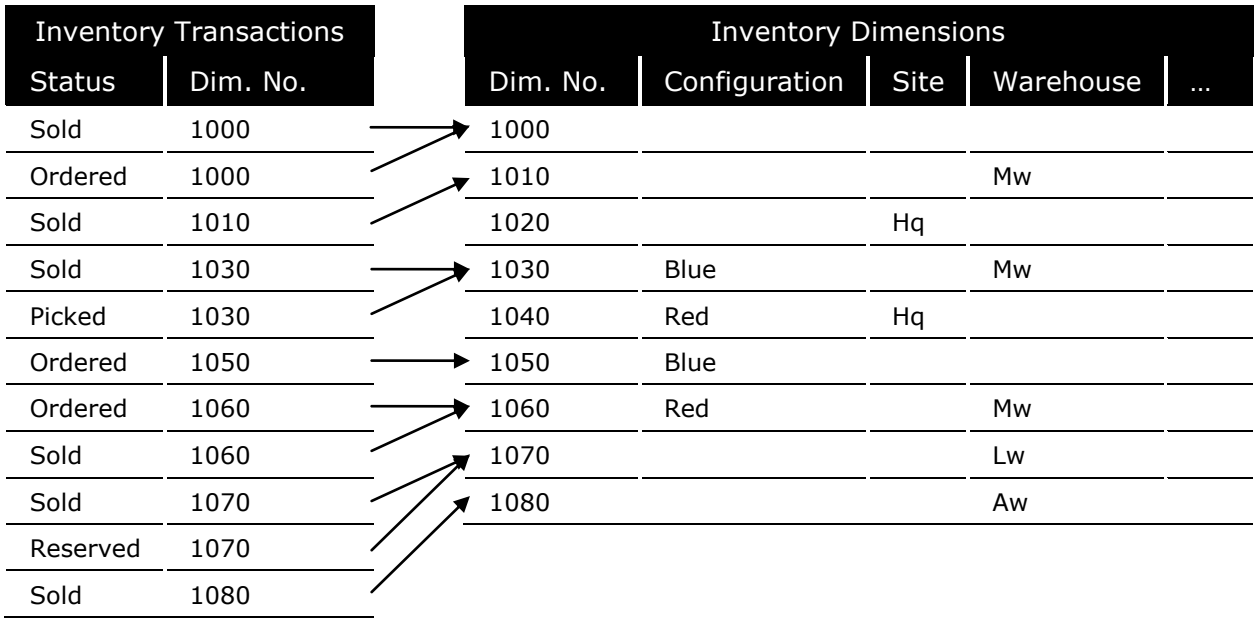


Figure 14 Multisite inactive inventory transactions

When the inventory dimensions have been updated by the multisite activation, the content of the tables will be as shown in **Figure 15**. The figure shows that several inventory transactions refer to inventory dimensions that contain the default site and warehouse. This is allowed for issued or received transactions. Transactions with an Ordered status should only refer to an inventory dimension with a site and a blank warehouse.

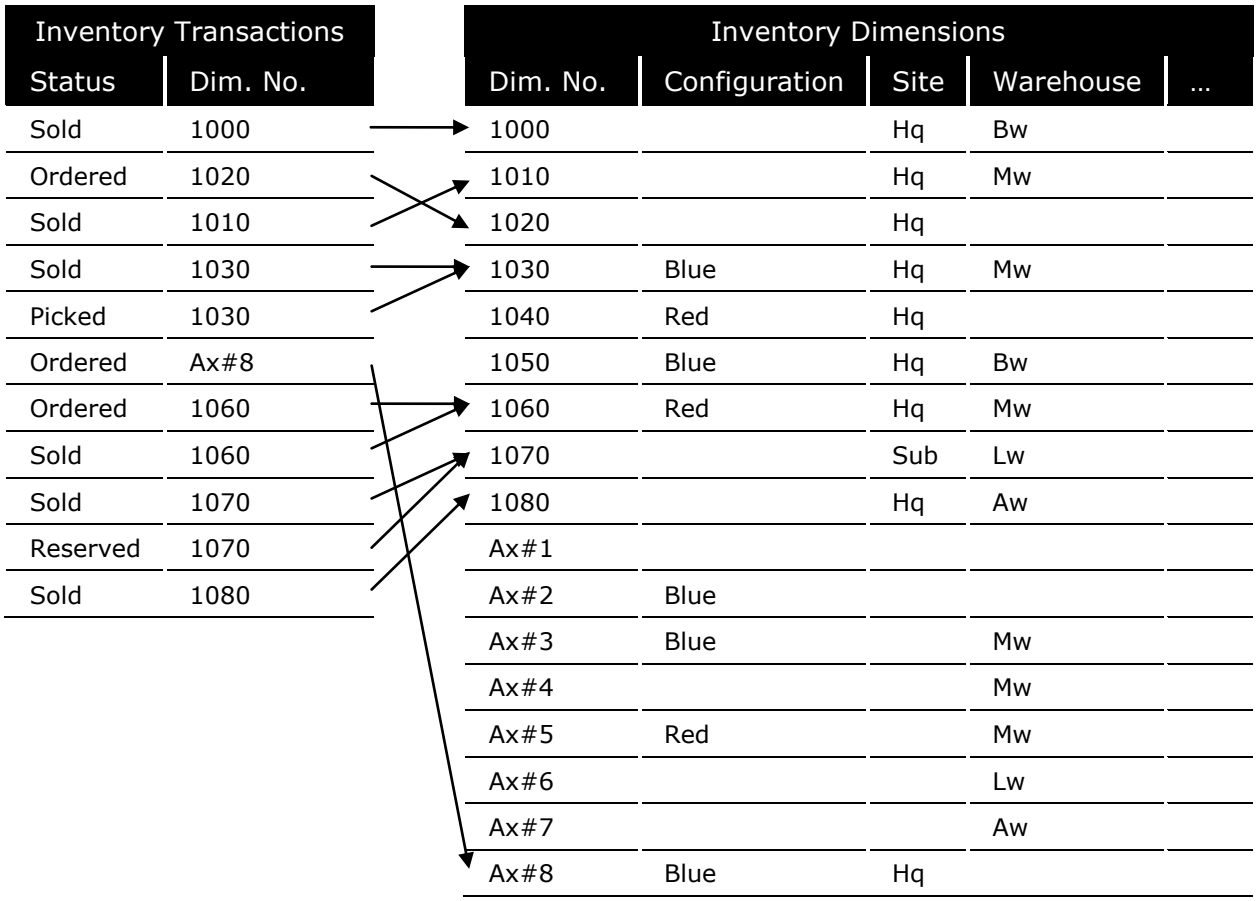
Inventory Transactions	
Status	Dim. No.
Sold	1000
Ordered	1000
Sold	1010
Sold	1030
Picked	1030
Ordered	1050
Ordered	1060
Sold	1060
Sold	1070
Reserved	1070
Sold	1080

Inventory Dimensions				
Dim. No.	Configuration	Site	Warehouse	…
1000		Hq	Bw	
1010		Hq	Mw	
1020		Hq		
1030	Blue	Hq	Mw	
1040	Red	Hq		
1050	Blue	Hq	Bw	
1060	Red	Hq	Mw	
1070		Sub	Lw	
1080		Hq	Aw	

Figure 15 Multisite updated inventory transactions (intermediate results)

The update job that is implemented in the *updateInventTrans* instance method on the *InventSiteActivateUpdate* class handles the redirection of the inventory transactions that are not issued or received and that refer to inventory dimensions that contain both default site and default warehouse. The method calls the *updateInventTrans* instance method on *InventSiteActivateupdRuleInvLoc*, which through the use of the *BlankDefaultLocOnInventDim* macro, redirects the transactions to inventory dimensions where the warehouse is blank when the current warehouse is equal to the default site. The method uses the macro by parsing in a set of field-value pairs, which ensures that only inventory transactions that match either of the field-value pairs, are updated.

After the update job has completed, the content of the tables will be as shown in **Figure 16** below.



Inventory Transactions	
Status	Dim. No.
Sold	1000
Ordered	1020
Sold	1010
Sold	1030
Picked	1030
Ordered	Ax#8
Ordered	1060
Sold	1060
Sold	1070
Reserved	1070
Sold	1080

Inventory Dimensions				
Dim. No.	Configuration	Site	Warehouse	…
1000		Hq	Bw	
1010		Hq	Mw	
1020		Hq		
1030	Blue	Hq	Mw	
1040	Red	Hq		
1050	Blue	Hq	Bw	
1060	Red	Hq	Mw	
1070		Sub	Lw	
1080		Hq	Aw	
Ax#1				
Ax#2	Blue			
Ax#3	Blue		Mw	
Ax#4			Mw	
Ax#5	Red		Mw	
Ax#6			Lw	
Ax#7			Aw	
Ax#8	Blue	Hq		

Figure 16 Multisite activated inventory transactions

Purchase lines update

When multisite is activated, purchase lines must always refer to inventory dimensions where site is specified.

When multisite is inactive, the purchase line should refer to an inventory dimension with blank site and warehouse and all related inventory transactions have the status of either Purchased or Sold. When multisite is active, the purchase line should refer to the default site and default warehouse. If an inventory transaction does not have the status of Purchased or Sold, then the purchase line should refer to the default site and a blank warehouse.

Figure 17 below shows an example of purchase lines and inventory dimensions where multisite is inactive.

Purchase Lines		Inventory Dimensions				
Status	Dim. No.	Dim. No.	Configuration	Site	Warehouse	…
Open	1000	1000				
Open	1000	1010			Mw	
Invoiced	1010	1020		Hq		
Invoiced	1030	1030	Blue		Mw	
Received	1030	1040	Red	Hq		
Open	1050	1050	Blue			
Open	1060	1060	Red		Mw	
Invoiced	1060	1070			Lw	
Invoiced	1070	1080			Aw	
Open	1070					
Received	1080					

Figure 17 Multisite inactive purchase lines

When the inventory dimensions have been updated by the multisite activation, the content of the tables will be as shown in **Figure 18**. The figure shows that several purchase lines refer to inventory dimensions that contain the default site and warehouse. This is allowed for purchase lines that have the status Purchased or Sold. Purchase lines that have the status Open should only refer to inventory dimensions with a site and a blank warehouse.

Purchase Lines	
Status	Dim. No.
Open	1000
Open	1000
Invoiced	1010
Invoiced	1030
Received	1030
Open	1050
Open	1060
Invoiced	1060
Invoiced	1070
Open	1070
Received	1080

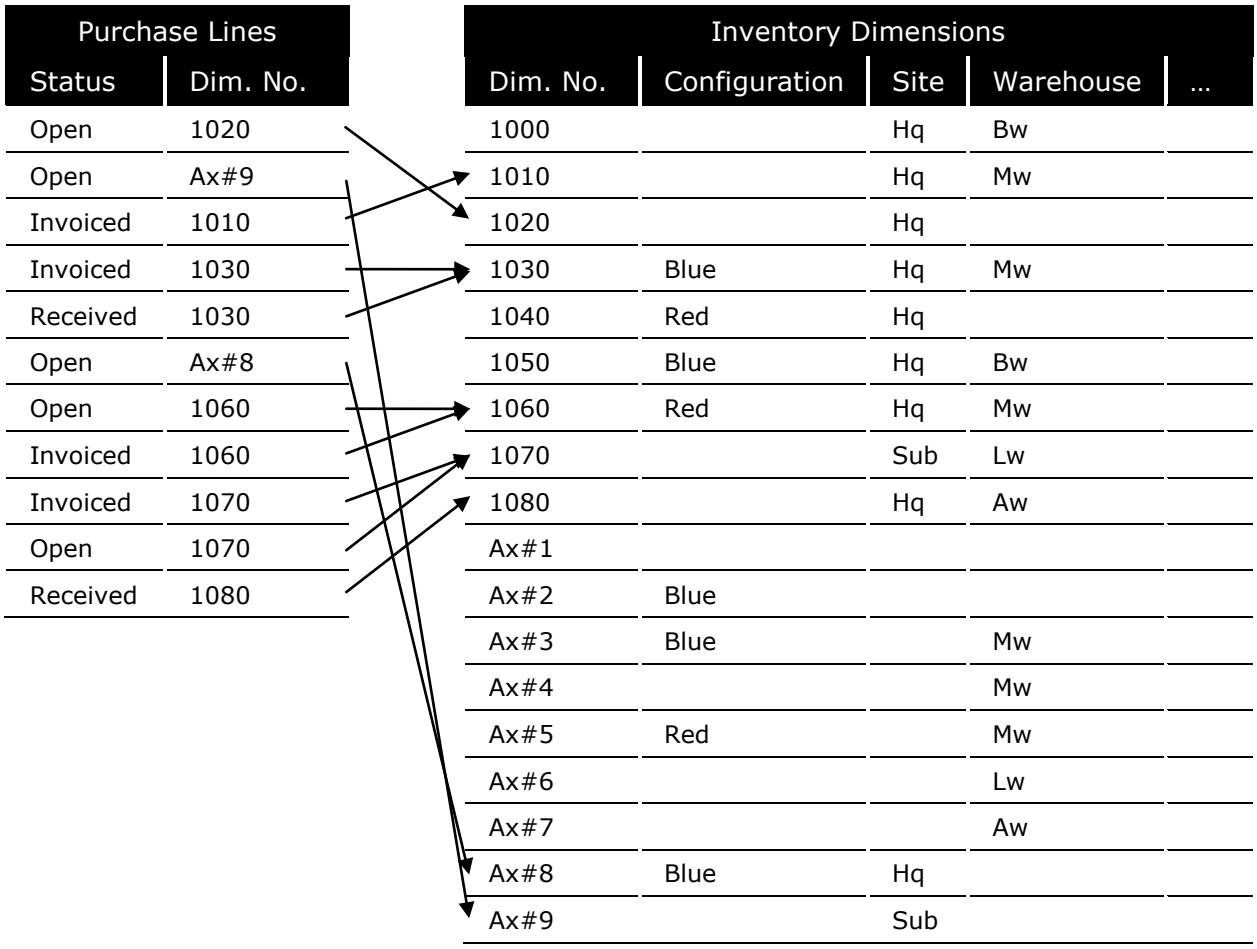
Inventory Dimensions				
Dim. No.	Configuration	Site	Warehouse	…
1000		Hq	Bw	
1010		Hq	Mw	
1020		Hq		
1030	Blue	Hq	Mw	
1040	Red	Hq		
1050	Blue	Hq	Bw	
1060	Red	Hq	Mw	
1070		Sub	Lw	
1080		Hq	Aw	

Figure 18 Multisite updated purchase lines (intermediate results)

The purchase lines update job updates the purchase lines that refer to default site and default warehouse and that relate to either inventory transactions that do not have the status Purchased or Sold or no inventory transactions at all. The update redirects the purchase lines to inventory dimensions where warehouse is blank.

This update job is implemented in the *updatePurchLine* instance method on the *InventSiteActivateUpdate* class calls the *updatePurchLine* instance method on *InventSiteActivateupdRuleInvTrans*. This method calls other methods that utilize the *BlankDefaultLocFromOpenInventTrans* and *BlankDefaultLocWhenNoInventTrans* macros. The *BlankDefaultLocFromOpenInventTrans* macro redirects Purchase lines to inventory dimension with a blank site when the purchase line is related to inventory transactions that do not have the status Purchased or Sold. The *BlankDefaultLocWhenNoInventTrans* macro redirects purchase lines to a blank warehouse as well, when the purchase lines are not related to any inventory transaction.

After the update job has completed, the content of the tables will be as shown in **Figure 19** below.



Purchase Lines	
Status	Dim. No.
Open	1020
Open	Ax#9
Invoiced	1010
Invoiced	1030
Received	1030
Open	Ax#8
Open	1060
Invoiced	1060
Invoiced	1070
Open	1070
Received	1080

Inventory Dimensions				
Dim. No.	Configuration	Site	Warehouse	…
1000		Hq	Bw	
1010		Hq	Mw	
1020		Hq		
1030	Blue	Hq	Mw	
1040	Red	Hq		
1050	Blue	Hq	Bw	
1060	Red	Hq	Mw	
1070		Sub	Lw	
1080		Hq	Aw	
Ax#1				
Ax#2	Blue			
Ax#3	Blue		Mw	
Ax#4			Mw	
Ax#5	Red		Mw	
Ax#6			Lw	
Ax#7			Aw	
Ax#8	Blue	Hq		
Ax#9		Sub		

Figure 19 Multisite activated purchase lines

Note: After multisite has been activated, the first two purchase lines shown in **Figure 19** no longer refer to the same inventory dimensions records. This can occur when the related inventory transactions refer to different sites. In this example, the first purchase line refers to inventory transactions that reference site Hq, whereas the second purchase line refers to inventory transactions that reference site Sub. Consequently, the first purchase line now refers to an inventory dimensions record that contains only site Hq, and the second purchase line refers to an inventory dimensions record that contains only site Sub.

Updating historical data

The multisite activation updates company data in Microsoft Dynamics AX 2009 where it is possible to continue the daily work in a multisite activated company.

The activation does not consider historical data like invoices, packing slips, posted inventory journals, posted production journals, invoiced sales lines, invoiced purchase lines, etc. The update of the inventory dimensions does result in these tables having assigned sites and warehouses through their references to inventory dimensions.

In cases where the table records referenced inventory dimensions with blank warehouse and blank site when multisite was inactive, these records now refer to the default site and default warehouse.

This may result in an inconsistency between the site that is referenced by the table and the site that is referenced by the related inventory transaction. This could happen in cases where the inventory transaction originally referred to a warehouse that is not related to the default site. When multisite is activated, the inventory transaction then refers to a different site than what the historical table refers to.

The inconsistency can be reconciled by executing the **Multisite activation – dimension inconsistency clean up** job (**Administration** > **Periodic** > **System** > **Multisite activation – dimension inconsistency clean up**). The job can be processed for a specific time, so that a company can work backwards in time and correct the historical data.

Caution: Do not run this batch job on transactions that are generated after you have activated the multisite functionality. If the warehouse that is specified as the default warehouse during the multisite activation process is used on any transactions that are generated after the multisite activation, the warehouse information on those transactions will be lost after running this batch job.

Note: It is important to note that these inconsistencies do not result in errors in the application if they are not corrected.

The job is implemented by the *InventSiteInconsistencyCleanUp* class, so this class should be customized when tables contain historical data that are not necessary to update during the multisite activation. The tables can then be updated afterwards as long as they do not result in errors in the application when the data is not updated.

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