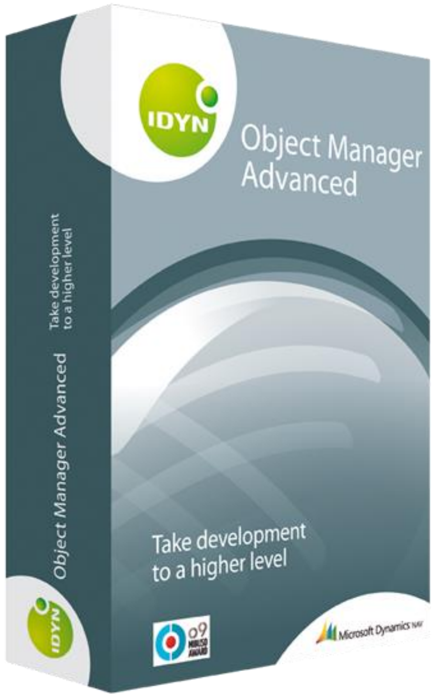
Manual
Object Manager Advanced

Award winning software

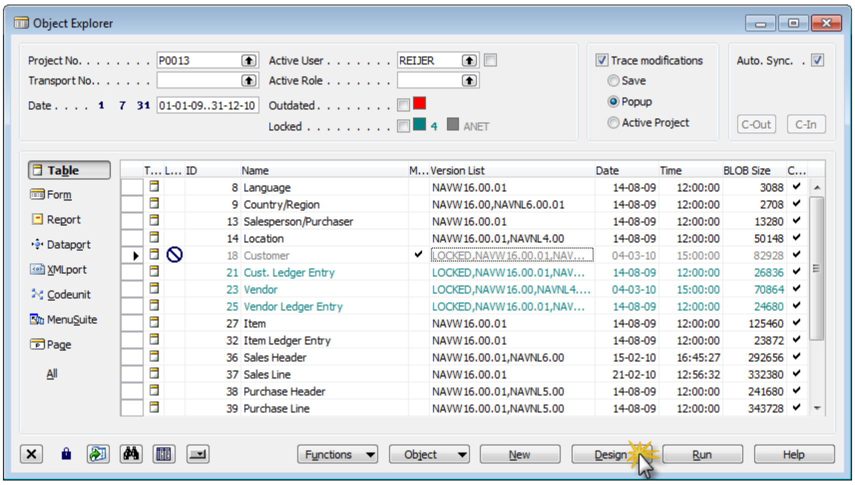
Version 8.54



1. Introduction

The Object Manager Advanced is the most comprehensive product for development and deployment for Dynamics NAV. The quality and continuity are recognized by Mibuso (“Best Download Award”) as well as Microsoft (“Microsoft Innovation Award 2010”).

The Object Manager Advanced is the best tool for supporting your developers and consultants in Microsoft Dynamics Sure Step or the other methodology tools in place. Using the Object Manager Advanced will not only help you develop in Microsoft Dynamics NAV, it will also reduce time and costs when it comes to maintaining your solution and deploying it to your customer. The newest version of our award winning tool, is packed with new features as well as enhancements on existing functionality.

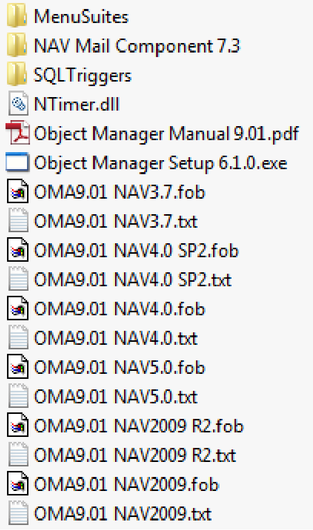


Project No. P0013
Transport No. . . . . . .
Date . . . . 1 7 31 01-01-09..31-12-10
Active User . . . . REIJER ☐
Active Role . . . .
Outdated . . . . ☐ 🟥
Locked . . . . ☐ 🟩 4 ▨ ANET
☑ Trace modifications
○ Save
● Popup
○ Active Project
Auto. Sync. . ☑

T...	L...	ID	Name	M...	Version List	Date	Time	BLOB Size	C...
		8	Language		NAVW16.00.01	14-08-09	12:00:00	3088	✓
		9	Country/Region		NAVW16.00,NAVNL6.00.01	14-08-09	12:00:00	2708	✓
		13	Salesperson/Purchaser		NAVW16.00.01	14-08-09	12:00:00	13280	✓
		14	Location		NAVW16.00.01,NAVNL4.00	14-08-09	12:00:00	50148	✓
	⃠	18	Customer	✓	LOCKED,NAVW16.00.01,NAV...	04-03-10	15:00:00	82928	✓
		21	Cust. Ledger Entry		LOCKED,NAVW16.00.01,NAV...	14-08-09	12:00:00	26836	✓
		23	Vendor		LOCKED,NAVW16.00,NAVNL4....	04-03-10	15:00:00	70864	✓
		25	Vendor Ledger Entry		LOCKED,NAVW16.00.01,NAV...	14-08-09	12:00:00	24680	✓
		27	Item		NAVW16.00.01	14-08-09	12:00:00	125460	✓
		32	Item Ledger Entry		NAVW16.00.01	14-08-09	12:00:00	23872	✓
		36	Sales Header		NAVW16.00.01,NAVNL6.00	15-02-10	16:45:27	292656	✓
		37	Sales Line		NAVW16.00.01	21-02-10	12:56:32	332380	✓
		38	Purchase Header		NAVW16.00.01,NAVNL5.00	14-08-09	12:00:00	241680	✓
		39	Purchase Line		NAVW16.00.01,NAVNL5.00	14-08-09	12:00:00	343728	✓

2. Installing and Setup

The Object Manager installation distributive contains the following files and folders:



MenuSuites
NAV Mail Component 7.3
SQLTriggers
NTimer.dll
Object Manager Manual 9.01.pdf
Object Manager Setup 6.1.0.exe
OMA9.01 NAV3.7.fob
OMA9.01 NAV3.7.txt
OMA9.01 NAV4.0 SP2.fob
OMA9.01 NAV4.0 SP2.txt
OMA9.01 NAV4.0.fob
OMA9.01 NAV4.0.txt
OMA9.01 NAV5.0.fob
OMA9.01 NAV5.0.txt
OMA9.01 NAV2009 R2.fob
OMA9.01 NAV2009 R2.txt
OMA9.01 NAV2009.fob
OMA9.01 NAV2009.txt

- **MenuSuites**
  Folder with MenuSuites with different object numbers. If MenuSuite 51 or 1051 is already in use in your database you can choose here a file with another object number.

- **SQLTriggers**
  Folder with the SQL statements that are used. For more information see sections 4.1 - Add SQL Trigger to the Object Table and 6.11 - Import Transport with SQL Trigger.

- **NAV Mail Component 7.3**
  Folder with objects of Mail Component 7.3.

- **NTimer.dll**
  A DLL of NAV which in most cases is installed when you have installed NAV. When you do not have installed NAV you will have to register this file manually with e.g. regsvr32. For more information see section 2.2 - Installing NTimer.dll.

- **Object Manager Manual 9.01.pdf**
  This document

- **Object Manager Setup 6.1.0.exe**
  A setup which makes it possible to export and import the C/AL code of an object. This is used in e.g. version control.

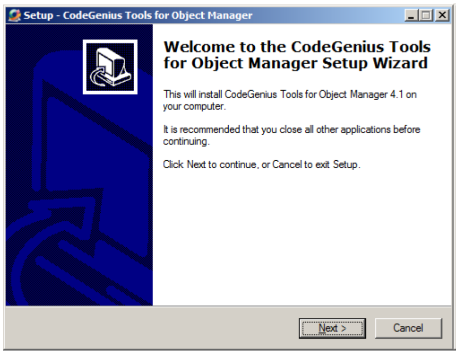
- **6 times OMA9.01 NAVxxx.fob**
  The objects in .fob format for the corresponding Microsoft Dynamics NAV version.

- **6 times OMA9.01 NAVxxx.txt**
  The objects in .txt format for the corresponding Microsoft Dynamics NAV version.

2.1. *Installing*

Start the “Object Manager Setup 6.1.0.exe”. This executable installs a DLL file which makes it possible for the Object Manager to export and import the C/AL code of an object. This is used in e.g. Version and Source Control, Where Used Functionality, Renumber Objects, Renumber Fields.

It opens a wizard, just click Next, Next, Install and Finish.



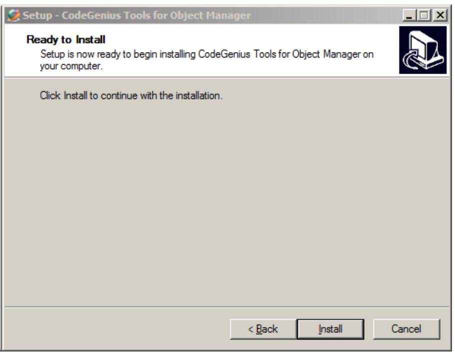
Setup - CodeGenius Tools for Object Manager

Welcome to the CodeGenius Tools for Object Manager Setup Wizard

This will install CodeGenius Tools for Object Manager 4.1 on your computer.

It is recommended that you close all other applications before continuing.

Click Next to continue, or Cancel to exit Setup.

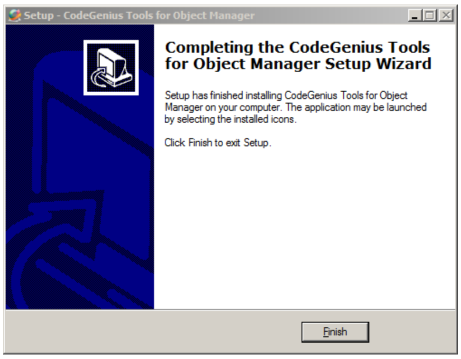


Setup - CodeGenius Tools for Object Manager

Ready to Install

Setup is now ready to begin installing CodeGenius Tools for Object Manager on your computer.

Click Install to continue with the installation.



Setup - CodeGenius Tools for Object Manager

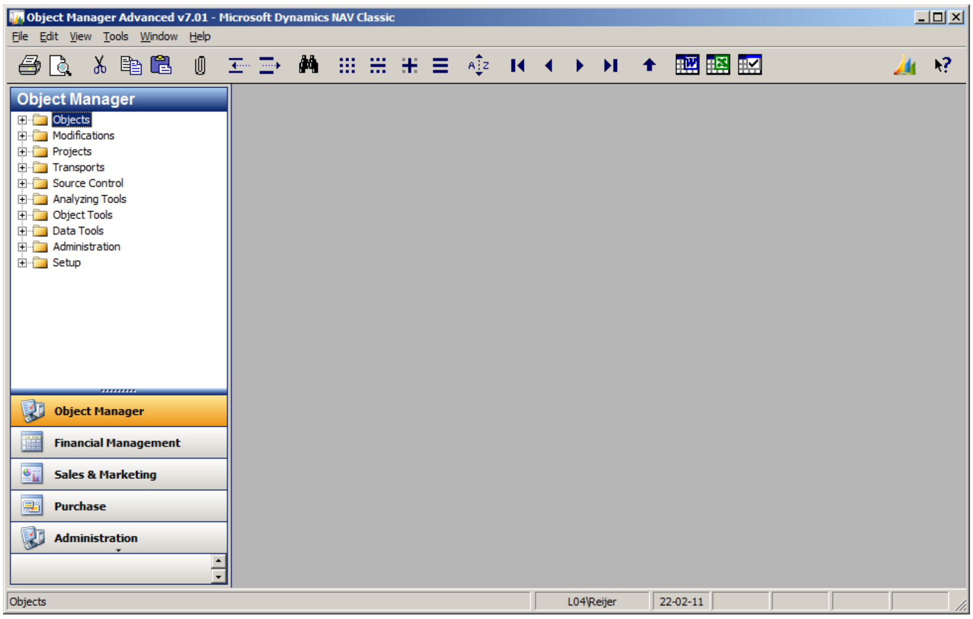
Completing the CodeGenius Tools for Object Manager Setup Wizard

Setup has finished installing CodeGenius Tools for Object Manager on your computer. The application may be launched by selecting the installed icons.

Click Finish to exit Setup.

Import the Object Manager Advanced .fob file for your version of Microsoft Dynamics NAV in the Object Designer.

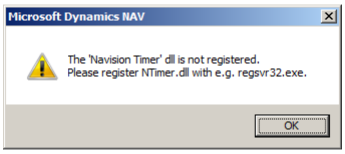
Toggle the Navigation Pane off and on, or restart Microsoft Dynamics NAV to enable the Object Manager menu.



Object Manager Advanced v7.01 - Microsoft Dynamics NAV Classic

2.2. Installing NTimer.dll

The NTimer DLL makes it possible for the Object Manager to poll. This is used in tracing object modifications. The DLL is in most cases already installed when you have installed NAV. If not you will get an error if you start tracing object modifications. For more information see 3.2 - Trace Modifications.



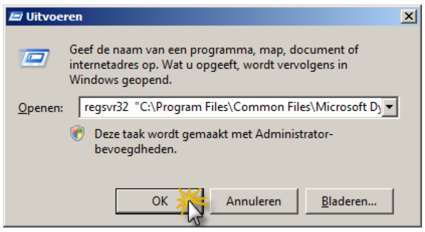
The 'Navision Timer' dll is not registered.
Please register NTimer.dll with e.g. regsvr32.exe.

If you get this error you will have to register the DLL manually. Copy the DLL to the folder:

```
C:\Program Files\Common Files\Microsoft Dynamics
NAV\Timer
```

Then execute the following command line with regsvr32.

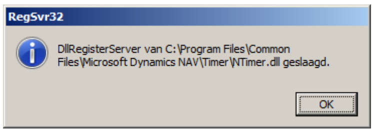
```
"C:\Program Files\Common Files\Microsoft Dynamics
NAV\Timer\NTimer.dll"
```



Geef de naam van een programma, map, document of internetadres op. Wat u opgeeft, wordt vervolgens in Windows geopend.

Openen: regsvr32 "C:\Program Files\Common Files\Microsoft Dy

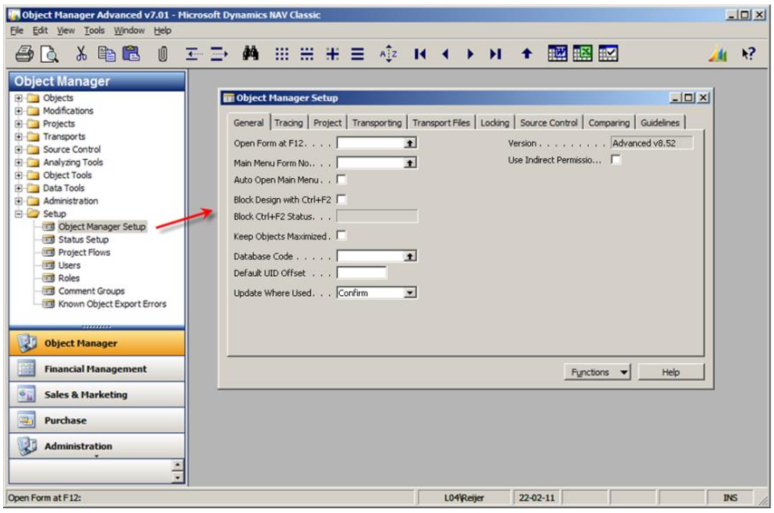
Deze taak wordt gemaakt met Administrator-bevoegdheden.



DllRegisterServer van C:\Program Files\Common Files\Microsoft Dynamics NAV\Timer\NTimer.dll geslaagd.

2.3. Initialize Setup

Default settings (statuses, project flows, etc.) are initialized the first time you open the “Object Manager Setup” card in a development database. If you installed the Object Manager in a test database or in a production database you can initialize the setup manual with another initialize option under the button Functions.

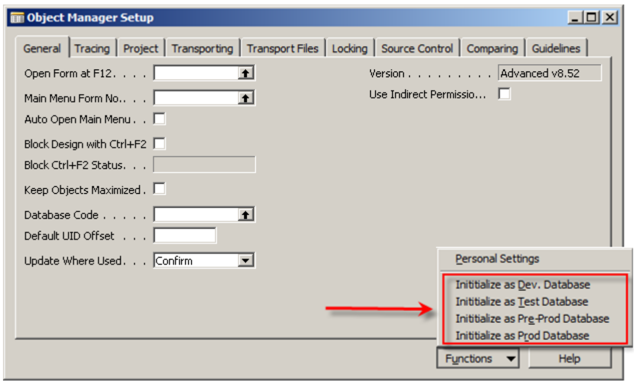


Version . . . . Advanced v8.52

Update Where Used. . . Confirm

The various setup possibilities will be explained later when used, but one we want to address here already: Database Code.
Use this field to select a database defined on the “Database Card” window in case the Object Manager:

- Cannot determine automatically what server/database it should use
- Needs more permissions to the database server then provided to the user working with the Object Manager



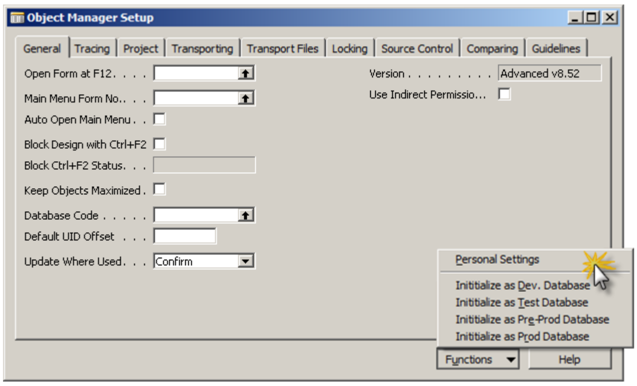
Version . . . . Advanced v8.52

Update Where Used. . . Confirm

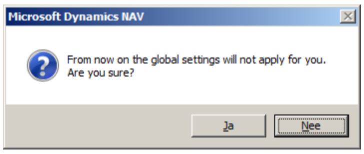
The differences between these options are listed in appendix D - Setup Initialization Methods.

2.4. Personal Setup

When developing with multiple developers in one database it can be necessary to have your personal settings. E.g. if somebody is using UltraCompare for analyzing modifications and somebody else is using Beyond Compare. You can activate your personal setup by selecting the menuoption “Personal Settings” under button Functions in the Setup.

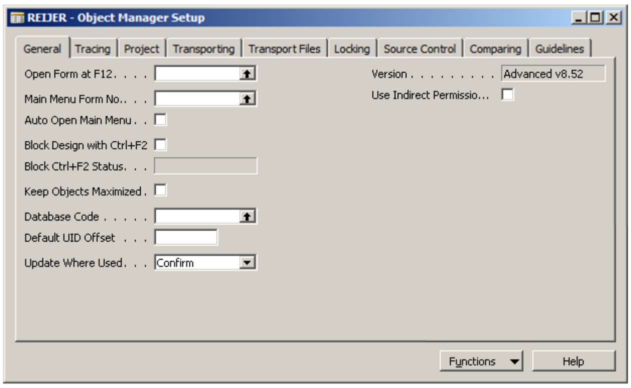


Open Form at F12. . . .
Main Menu Form No.. . .
Auto Open Main Menu . . ☐
Block Design with Ctrl+F2 ☐
Block Ctrl+F2 Status. . .
Keep Objects Maximized . ☐
Database Code . . . . .
Default UID Offset . . .
Update Where Used. . . Confirm
Version . . . . Advanced v8.52
Use Indirect Permissio... ☐



From now on the global settings will not apply for you.
Are you sure?

By choosing Yes in the dialog, the Object Manager saves your settings to the database. The title bar has now changed to "USERID – Object Manager Setup", indicating that you use personal settings for Object Manager.

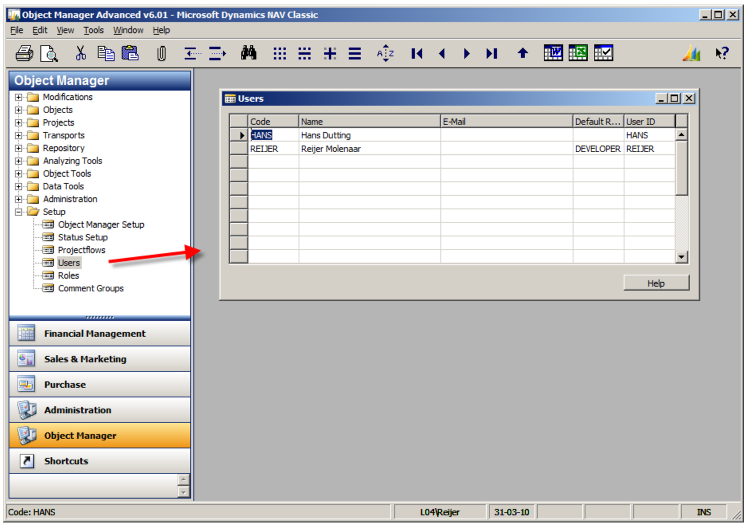


Open Form at F12. . . .
Main Menu Form No.. . .
Auto Open Main Menu . . ☐
Block Design with Ctrl+F2 ☐
Block Ctrl+F2 Status. . .
Keep Objects Maximized . ☐
Database Code . . . . .
Default UID Offset . . .
Update Where Used. . . Confirm
Version . . . . Advanced v8.52
Use Indirect Permissio... ☐

You can disable your personal settings by deleting the record.

2.5. Users and Roles

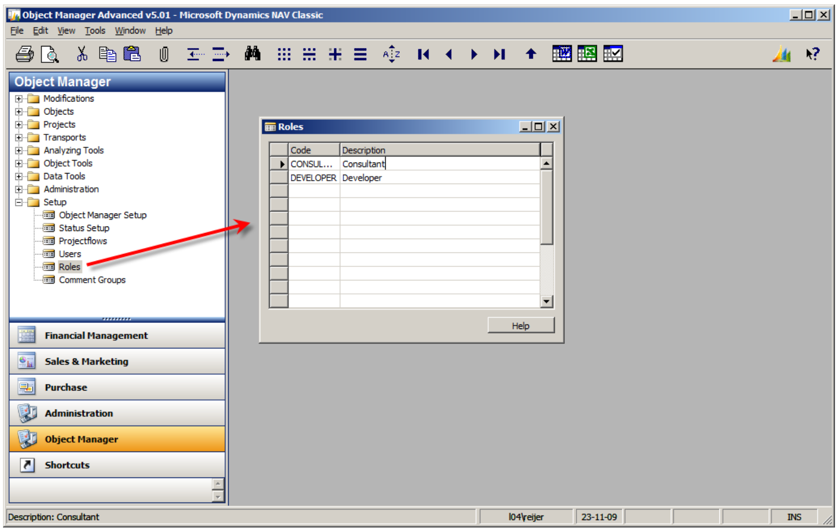
Add Object Manager Users
Open Setup > Users and add the user information.



Code	Name	E-Mail	Default R…	User ID
HANS	Hans Dutting			HANS
REIJER	Reijer Molenaar		DEVELOPER	REIJER

A user can have a default role. This is used when you insert a new project. The code of the active user is placed in the corresponding role on the project card.

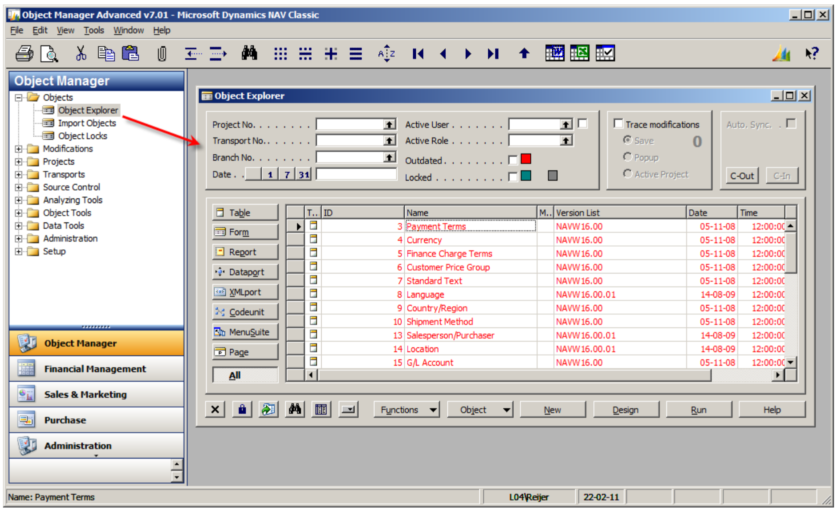
Add Object Manager Roles
Open Setup > Roles and add the necessary roles.



Code	Description
CONSUL…	Consultant
DEVELOPER	Developer

2.6. Update C/AL History

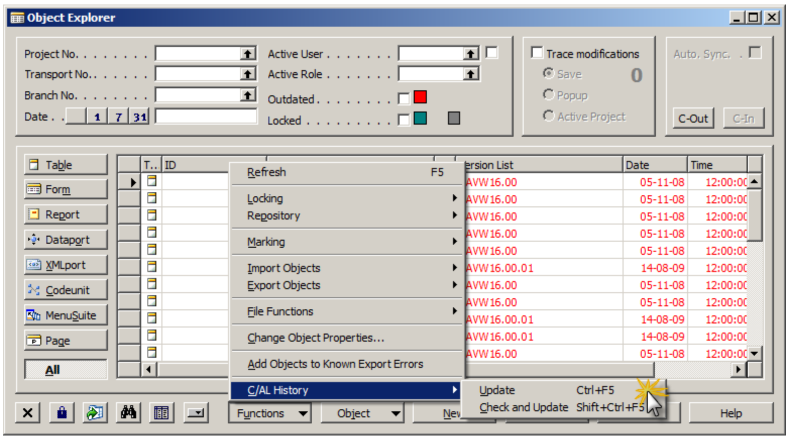
Open menu Object Manager > Objects > Object Explorer.



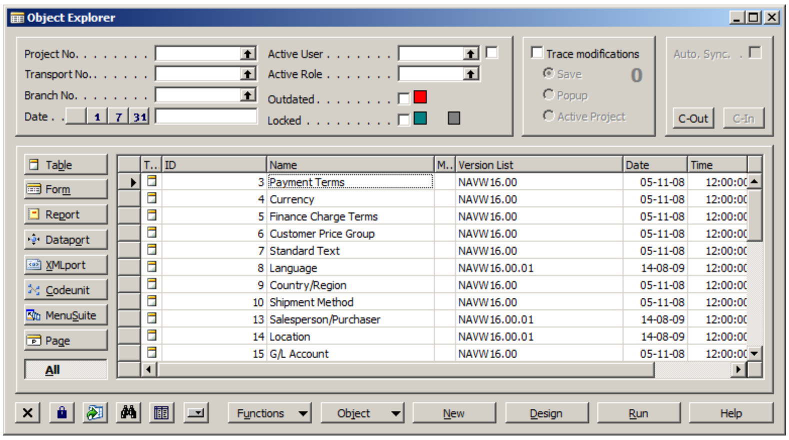
T..	ID	Name	M..	Version List	Date	Time
	3	Payment Terms		NAVW16.00	05-11-08	12:00:0
	4	Currency		NAVW16.00	05-11-08	12:00:0
	5	Finance Charge Terms		NAVW16.00	05-11-08	12:00:0
	6	Customer Price Group		NAVW16.00	05-11-08	12:00:0
	7	Standard Text		NAVW16.00	05-11-08	12:00:0
	8	Language		NAVW16.00.01	14-08-09	12:00:0
	9	Country/Region		NAVW16.00	05-11-08	12:00:0
	10	Shipment Method		NAVW16.00	05-11-08	12:00:0
	13	Salesperson/Purchaser		NAVW16.00.01	14-08-09	12:00:0
	14	Location		NAVW16.00.01	14-08-09	12:00:0
	15	G/L Account		NAVW16.00	05-11-08	12:00:0

Name: Payment Terms | L04\Reijer | 22-02-11

All objects are displayed as red lines, which means that the “C/AL History” is out of date. This is always the case when you setup the Object Manager for the first time in a database. To update the “C/AL History” you click the option “Update C/AL History”.



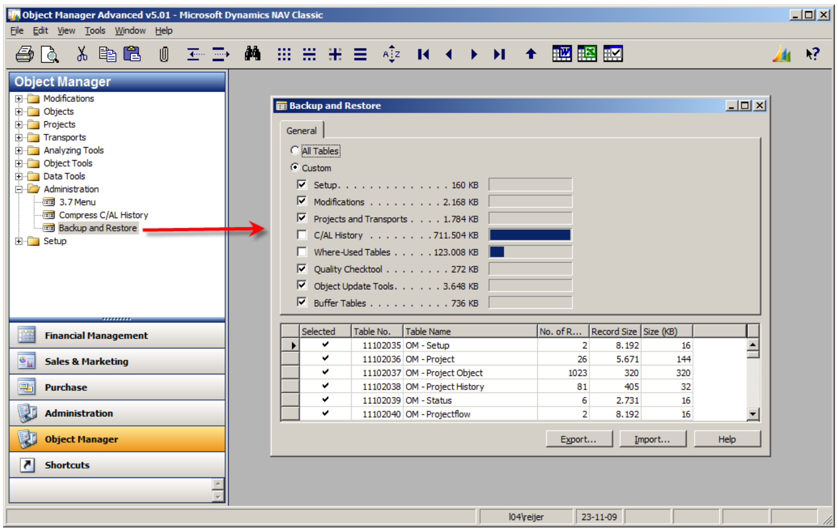
These function stores the C/AL of all objects to the “C/AL History” (for more information see chapter 9 - Version and Source Control).



T..	ID	Name	M..	Version List	Date	Time
	3	Payment Terms		NAVW16.00	05-11-08	12:00:0
	4	Currency		NAVW16.00	05-11-08	12:00:0
	5	Finance Charge Terms		NAVW16.00	05-11-08	12:00:0
	6	Customer Price Group		NAVW16.00	05-11-08	12:00:0
	7	Standard Text		NAVW16.00	05-11-08	12:00:0
	8	Language		NAVW16.00.01	14-08-09	12:00:0
	9	Country/Region		NAVW16.00	05-11-08	12:00:0
	10	Shipment Method		NAVW16.00	05-11-08	12:00:0
	13	Salesperson/Purchaser		NAVW16.00.01	14-08-09	12:00:0
	14	Location		NAVW16.00.01	14-08-09	12:00:0
	15	G/L Account		NAVW16.00	05-11-08	12:00:0

2.7. Backup and Restore

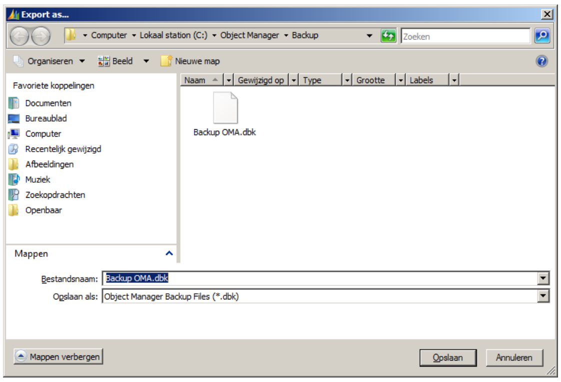
When you create a new development database (e.g. a new copy from the live database) you will have to transfer all your Object Manager data from the old development database to the new database.
This can be done with the backup and restore functionality.



○ All Tables
● Custom
☑ Setup. . . . . 160 KB
☑ Modifications . . . . 2.168 KB
☑ Projects and Transports . . . . 1.784 KB
☐ C/AL History . . . . 711.504 KB
☐ Where-Used Tables . . . . . 123.008 KB
☑ Quality Checktool . . . . 272 KB
☑ Object Update Tools. . . . . . . 3.648 KB
☑ Buffer Tables . . . . 736 KB

Selected	Table No.	Table Name	No. of R...	Record Size	Size (KB)
✓	11102035	OM - Setup	2	8.192	16
✓	11102036	OM - Project	26	5.671	144
✓	11102037	OM - Project Object	1023	320	320
✓	11102038	OM - Project History	81	405	32
✓	11102039	OM - Status	6	2.731	16
✓	11102040	OM - Projectflow	2	8.192	16

Check the modules you want to transfer and press Export. All data will be exported in dbk format.



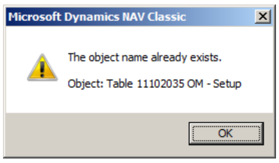
Bestandsnaam: Backup OMA.dbk
Opslaan als: Object Manager Backup Files (*.dbk)

You can import this file in your new development database with the Import button.

2.8. Upgrading Trial to Full Version

When you were using the trial version of the Object Manager and you purchased the full version you will get a fob with other object numbers. The trial objects have numbers starting at 91800. The full version objects are starting at number 11102035.

Importing the full version fob in a database with trial objects will result in the following error.



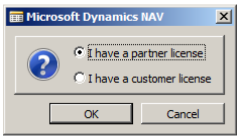
The object name already exists.

Object: Table 11102035 OM - Setup

This is because the objects have the same name. There are two methods to upgrade from the trial to the full version. It depends if you license allows you to create objects in the full version object range.

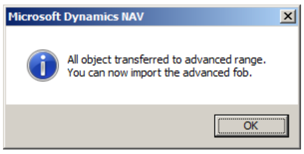
2.8.1. Upgrading with a Partner License

Start codeunit 91829 - OM - Upgrade to Full Version.
Choose “I have a partner license”.



● I have a partner license
○ I have a customer license

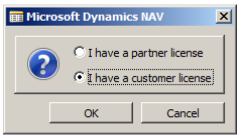
All objects are renumbered to the full version object range. You now can import the fob with the full version objects without any problems.



All object transferred to advanced range.
You can now import the advanced fob.

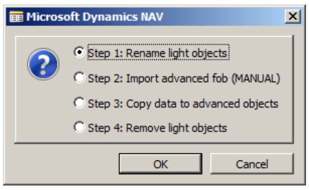
2.8.2. Upgrading with Customer License

If you do not have the permission to create objects in the full version object range you will have to choose “I have a customer license”.



○ I have a partner license
● I have a customer license

The following options have to be executed one by one.



● Step 1: Rename light objects
○ Step 2: Import advanced fob (MANUAL)
○ Step 3: Copy data to advanced objects
○ Step 4: Remove light objects

Steps 1, 3 and 4 are done automatically. Step 2 is a manual task.

NOTE: Step 3 and 4 have to be executed with codeunit 11102064 instead of codeunit 91829.

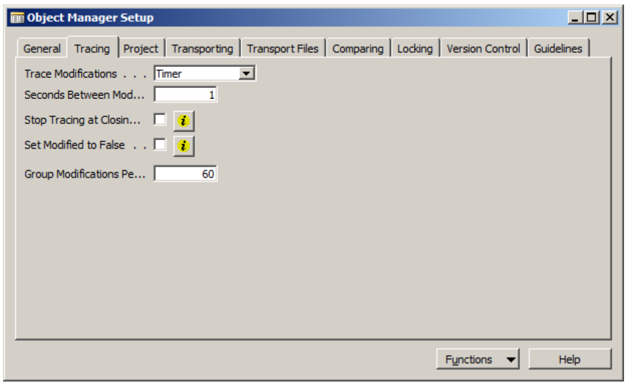
2.9. Cleaning Up

To remove Object Manager from your database select Administration > Administration Tasks.
On the “Administration Tasks” window select “Remove Object Manager from database” and press “Execute” to have all Object Manager objects and data removed. This action will take some time to complete.

3. Modifications

This chapter discusses the “Trace Modification” function and assigning modifications to projects.

3.1. Setup



Trace Modifications . . . Timer

Seconds Between Mod... 1

Stop Tracing at Closin... ☐

Set Modified to False . . ☐

Group Modifications Pe... 60

- **Trace Modifications**
  There are three methods to trace the modifications.
  - Timer: If you are using a C/SIDE database prior to NAV2009 R2 you must use the Timer option. With this option the Object Manager will poll every x seconds to see if any objects are changed.
  - SQL Trigger: If you are using an SQL database it is best to use the SQL Trigger option. (See section 4.1 - Add SQL Trigger to the Object Table). Every time an object changes, SQL will add a record to the modifications table. This method is also preferred above the Timer mode because the Object Manager can see who changed the objects. This is not possible when using the Timer method.
  - Integration Management: When using NAV2009 R2 or above it is best to use the option Integration Management. This method uses triggers in codeunit 1 to trace when an object changes. To use these triggers you will have to add some code to the Integration Management triggers. In the following code the red lines are added.

```
    PROCEDURE GetDatabaseTableTriggerSetup@25(TableId@1000 : Integer;VAR Insert@1001 : Boolean;VAR Modify@1002 :
Boolean;VAR Delete@1003 : Boolean;VAR Rename@1004 : Boolean);
    VAR
      IntegrationManagement@1005 : Codeunit 5150;
      OMIntegrationManagement@1006 : Codeunit 11102080;
    BEGIN
      IntegrationManagement.GetDatabaseTableTriggerSetup(TableId,Insert,Modify,Delete,Rename);
      OMIntegrationManagement.GetDatabaseTableTriggerSetup(TableId,Insert,Modify,Delete,Rename);
    END;

    PROCEDURE OnDatabaseInsert@26(RecRef@1000 : RecordRef);
    VAR
      IntegrationManagement@1001 : Codeunit 5150;
```

```
  OMIntegrationManagement@1002 : Codeunit 11102080;
BEGIN
  IntegrationManagement.OnDatabaseInsert(RecRef);
  OMIntegrationManagement.OnDatabaseInsert(RecRef);
END;

PROCEDURE OnDatabaseModify@27(RecRef@1000 : RecordRef);
VAR
  IntegrationManagement@1001 : Codeunit 5150;
  OMIntegrationManagement@1002 : Codeunit 11102080;
BEGIN
  IntegrationManagement.OnDatabaseModify(RecRef);
  OMIntegrationManagement.OnDatabaseModify(RecRef);
END;

PROCEDURE OnDatabaseDelete@28(RecRef@1000 : RecordRef);
VAR
  IntegrationManagement@1001 : Codeunit 5150;
  OMIntegrationManagement@1002 : Codeunit 11102080;
BEGIN
  IntegrationManagement.OnDatabaseDelete(RecRef);
  OMIntegrationManagement.OnDatabaseDelete(RecRef);
END;

PROCEDURE OnDatabaseRename@29(RecRef@1000 : RecordRef;xRecRef@1001 : RecordRef);
VAR
  IntegrationManagement@1002 : Codeunit 5150;
  OMIntegrationManagement@1003 : Codeunit 11102080;
BEGIN
  IntegrationManagement.OnDatabaseRename(RecRef,xRecRef);
  OMIntegrationManagement.OnDatabaseRename(RecRef,xRecRef);
END;
```

- **Seconds between Mod. Trace**
  Number of seconds between each time that the Object Manager looks for new changes.

- **Stop Tracing at Closing Menu**
  The Object Manager stops monitoring changes when the “Object Manager 3.7 Menu” or the “Object Explorer” is closed.

- **Set Modified to False**
  - True: The Object Manager sees an object with the modify flag as a change. When the modification is assigned to a project the modify flag of the object is set to false.

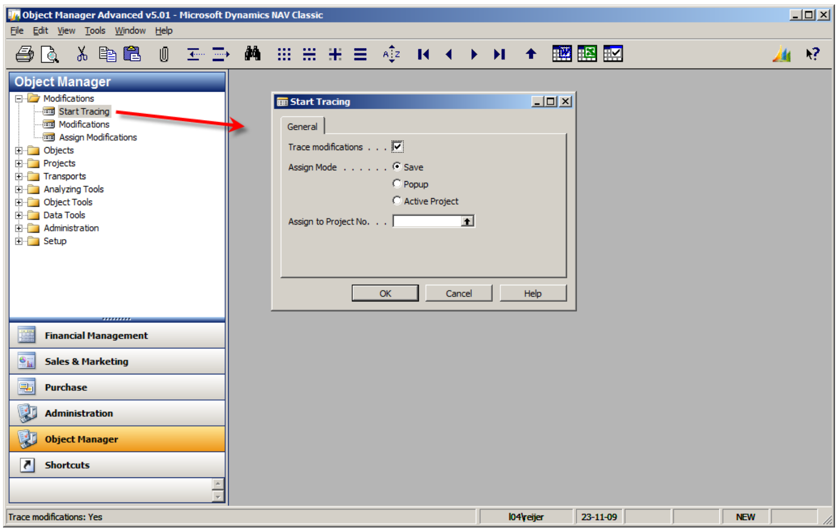
  - False: The Object Manager looks at all objects with the modify flag on and compares them to the last saved modification. If the datetime stamp of the object differs, the Object Manager sees it as a change which has to be assigned to a project. All objects keep the modify flag until the project is transported to your customer database. (See chapter 6 - Transport)

- **Group Modifications Period (sec.)**
  The number of seconds that several modifications will be grouped as one. If this setting if set to 60 you will have maximum one modification record per object per minute in the modification table.

NOTE: It is possible to disable Integration Management for the active session with the menu option “Disable Integration Management”. This can be needed for fob imports that will give a conflict in Integration Management.

3.2. Trace Modifications

Open the menu Modifications > Start Tracing.



Trace modifications . . . ☑

Assign Mode . . . . . . ● Save ○ Popup ○ Active Project

Assign to Project No. . .

Trace modifications: Yes

NOTE: If you get an error that the Navision Timer DLL is not registered you will have to register the NTimer.dll. For more information see section 2.2 - Installing NTimer.dll.

Start Tracing by selecting the “Trace modifications” option. Choose Save or Popup or Active Project.

- **Save**
  Save means that every time the Object Manager traces a modification on an object it will be saved to the Modifications table. You have to assign these modifications later to the project using the “Assign Modifications” form.

- **Popup**
  Popup means that every time the Object Manager traces a modification it will show a popup where you can choose the project the modification has to be assigned to.

- **Active Project**
  Active Project means that every time the Object Manager traces a modification it will be assigned to the project you have specified in the “Active Project No.” field.

  - Timer mode: If you use the trace modifications Timer method the object has to be locked by you. This is because the Object Manager does not know who changed the object.

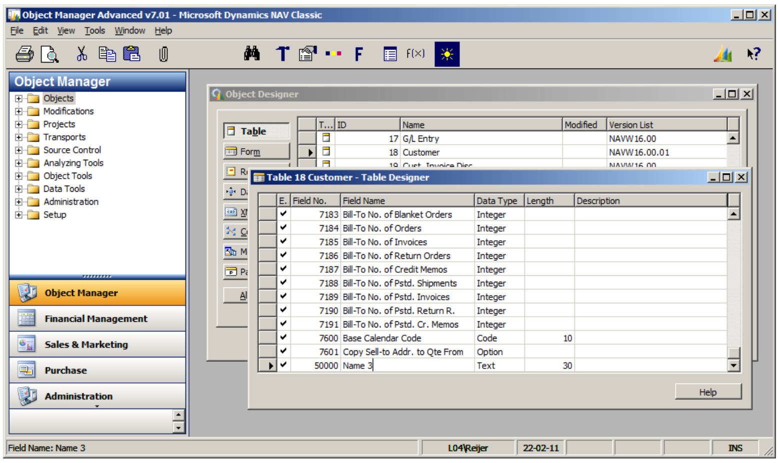
  - SQL Trigger and Integration Management: If you use the trace modification SQL Trigger or Integration Management method the modification will be assigned to your project even if you do not have locked the object. This is because the Object Manager knows who changed the object.

NOTE if working with Timer mode: If there are to many objects with the modified flag on the system can react slow. This is because the Object Manager compares all these objects with the last saved modification.

IMPORTANT NOTE if working with Timer mode: When you lock an object your colleague developers will not get the assign popup when you modify the object. So if you are working in an environment with several developers locking is recommended. For more information see section 4.2 - Lock Objects.

Example Save

Add an extra field to the Customer table and form and save the object.

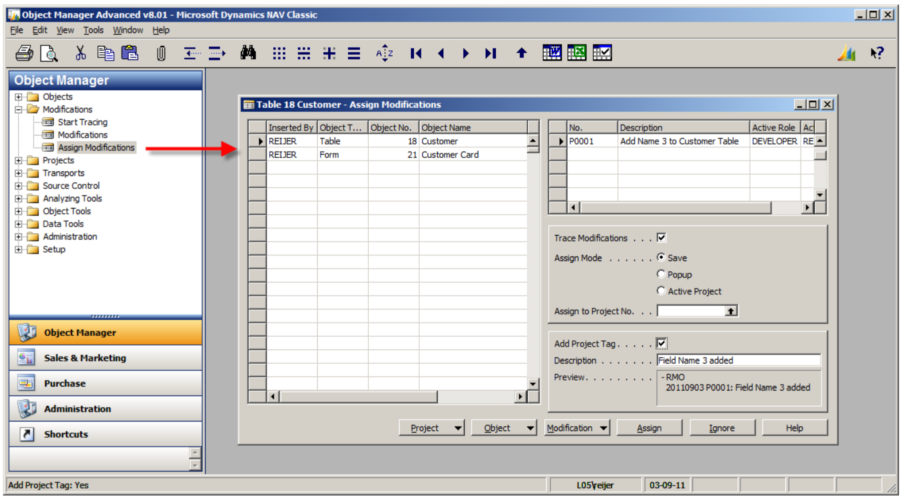


T…	ID	Name	Modified	Version List
	17	G/L Entry		NAVW16.00
	18	Customer		NAVW16.00.01

E.	Field No.	Field Name	Data Type	Length	Description
✓	7183	Bill-To No. of Blanket Orders	Integer		
✓	7184	Bill-To No. of Orders	Integer		
✓	7185	Bill-To No. of Invoices	Integer		
✓	7186	Bill-To No. of Return Orders	Integer		
✓	7187	Bill-To No. of Credit Memos	Integer		
✓	7188	Bill-To No. of Pstd. Shipments	Integer		
✓	7189	Bill-To No. of Pstd. Invoices	Integer		
✓	7190	Bill-To No. of Pstd. Return R.	Integer		
✓	7191	Bill-To No. of Pstd. Cr. Memos	Integer		
✓	7600	Base Calendar Code	Code	10	
✓	7601	Copy Sell-to Addr. to Qte From	Option		
✓	50000	Name 3	Text	30	

Field Name: Name 3 | L04\Reijer | 22-02-11 | INS

Go to Modifications > “Assign Modifications” and assign the modifications to a project. (for more information see chapter 5 - Project).



Inserted By	Object T…	Object No.	Object Name
REIJER	Table	18	Customer
REIJER	Form	21	Customer Card

No.	Description	Active Role	Ac
P0001	Add Name 3 to Customer Table	DEVELOPER	RE

Trace Modifications . . . ☑

Assign Mode . . . . . . ● Save ○ Popup ○ Active Project

Assign to Project No. . .

Add Project Tag . . . . ☑

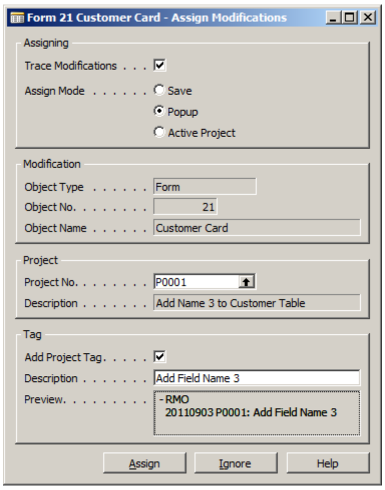
Description . . . . Field Name 3 added

Preview. . . . . - RMO
20110903 P0001: Field Name 3 added

Add Project Tag: Yes | L05\reijer | 03-09-11

Example Popup

Modify table Customer and save the object. The Assign Modification form will popup. You can assign the object direct to a project.



Form 21 Customer Card - Assign Modifications

Assigning

Trace Modifications . . . ☑

Assign Mode . . . . . . ○ Save

● Popup

○ Active Project

Modification

Object Type . . . . . . Form

Object No. 21

Object Name . . . . . . Customer Card

Project

Project No. P0001

Description . . . . Add Name 3 to Customer Table

Tag

Add Project Tag. . . . . ☑

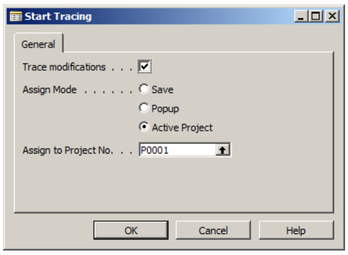
Description . . . . Add Field Name 3

Preview. . . . . -RMO
20110903 P0001: Add Field Name 3

This option works best with the trace modifications SQL Trigger or Integration Management method. Otherwise your colleagues will get the same popup.

Example Active Project

Modify table Customer and save the object. The modification will be assigned to your project straight away.



Start Tracing

General

Trace modifications . . . ☑

Assign Mode . . . . . . ○ Save

○ Popup

● Active Project

Assign to Project No. . . P0001

4. Locking

Locking is used to prevent that two developers will work on the same objects.

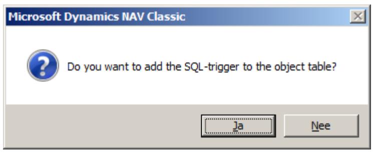
If you are working with a NAV version prior to NAV2009 R2 it is best to use the SQL possibility to execute T-SQL code whenever something changes in the object table. This is done by adding an SQL trigger to the object table.

If working with NAV2009 R2 or higher you can use the locking mechanism of NAV itself, thus there is no need to add the SQL trigger to the object table.

4.1. *Add SQL Trigger to the Object Table*

4.1.1. Automatic

When you change the option “Trace Modifications” to “SQL Trigger” in the Setup you will be asked if you want to add the SQL trigger to the object table.

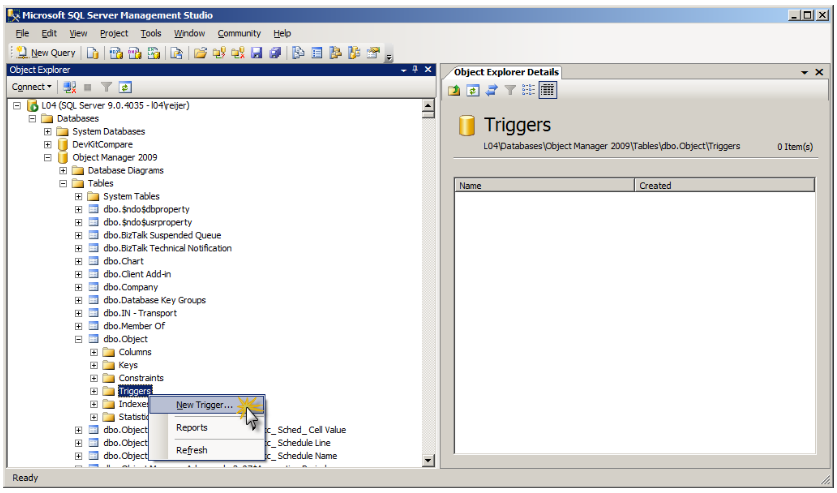


Do you want to add the SQL-trigger to the object table?

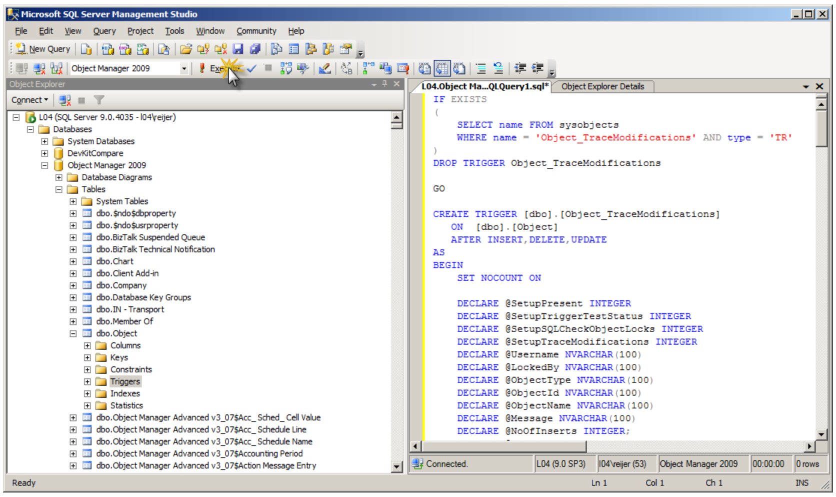
If this fails because e.g. you do not have the right permissions you have to add the trigger manually in SQL Server.

4.1.2. Manual in SQL Server

To add the SQL trigger, you open SQL Server Management Studio, go to the object table and choose the option to create a new trigger.



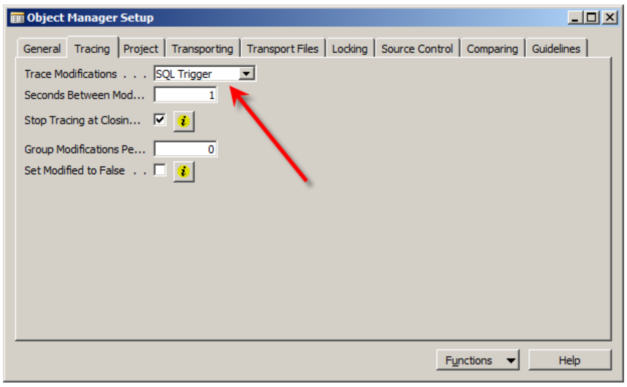
Then paste the SQL statement (see appendix E - Object Table SQL Trigger) in the Query window and press execute.


Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Community Help
New Query
Object Manager 2009
Execute
Object Explorer
Connect
L04.Object Ma...QLQuery1.sql*
Object Explorer Details
Ready

```
IF EXISTS
(
    SELECT name FROM sysobjects
    WHERE name = 'Object_TraceModifications' AND type = 'TR'
)
DROP TRIGGER Object_TraceModifications

GO

CREATE TRIGGER [dbo].[Object_TraceModifications]
   ON  [dbo].[Object]
   AFTER INSERT,DELETE,UPDATE
AS
BEGIN
    SET NOCOUNT ON

    DECLARE @SetupPresent INTEGER
    DECLARE @SetupTriggerTestStatus INTEGER
    DECLARE @SetupSQLCheckObjectLocks INTEGER
    DECLARE @SetupTraceModifications INTEGER
    DECLARE @Username NVARCHAR(100)
    DECLARE @LockedBy NVARCHAR(100)
    DECLARE @ObjectType NVARCHAR(100)
    DECLARE @ObjectId NVARCHAR(100)
    DECLARE @ObjectName NVARCHAR(100)
    DECLARE @Message NVARCHAR(100)
    DECLARE @NoOfInserts INTEGER;
```

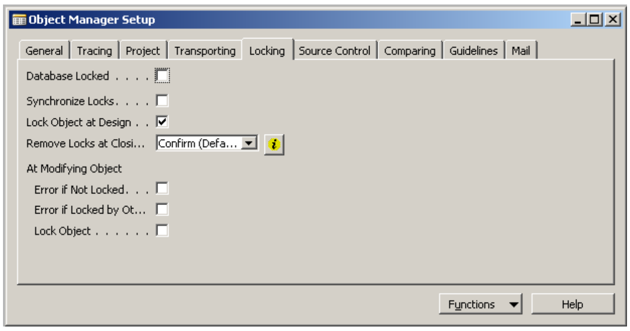
Connected. | L04 (9.0 SP3) | l04\veijer (53) | Object Manager 2009 | 00:00:00 | 0 rows

Now it is possible to set the Trace Modifications method to SQL Trigger.



Trace Modifications . . . SQL Trigger
Seconds Between Mod... 1
Stop Tracing at Closin... ☑
Group Modifications Pe... 0
Set Modified to False . . ☐

4.2. Lock Objects



Database Locked . . . . ☐
Synchronize Locks. . . . ☐
Lock Object at Design . . ☑
Remove Locks at Closi... Confirm (Defa...
At Modifying Object
Error if Not Locked. . . ☐
Error if Locked by Ot... ☐
Lock Object . . . . . . ☐

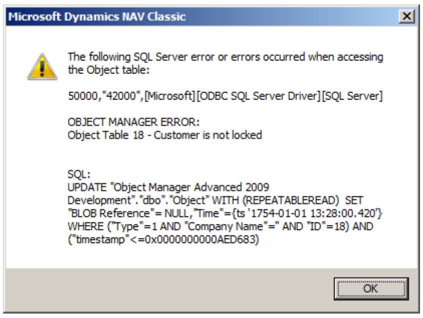
- **Database Locked**
  This option is only available in NAV2009 R2 and above. When enabling this setting locks placed in NAV are set in the Object Manager and vice versa.

- **Synchronize Locks**
  This option is only available in NAV2009 R2 and above. When enabling this setting locks placed in NAV are set in the Object Manager and vice versa.

- **Lock Objects at Design**
  When enabling this setting you will lock each object that you design with the Object Designer. This is recommended in an environment with more developers. Objects will also be locked if they are changed by one of the object tools like Renumber Objects.

- **Remove Locks at Closing 3.7 Menu and Object Explorer**
  When enabling this setting all your locks will be removed if you close the “Object Explorer”.

- **At Modifying Object**
  - Error if Not locked
    This is the safest locking mechanism. You will get an error if the object is not locked or is locked by somebody else:



```
The following SQL Server error or errors occurred when accessing
the Object table:

50000,"42000",[Microsoft][ODBC SQL Server Driver][SQL Server]

OBJECT MANAGER ERROR:
Object Table 18 - Customer is not locked


SQL:
UPDATE "Object Manager Advanced 2009
Development"."dbo"."Object" WITH (REPEATABLEREAD)  SET
"BLOB Reference"= NULL,"Time"={ts '1754-01-01 13:28:00.420'}
WHERE ("Type"=1 AND "Company Name"='' AND "ID"=18) AND
("timestamp"<=0x000000000AED683)
```

SQL Trigger

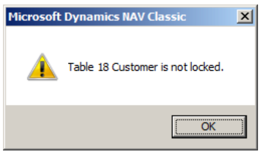
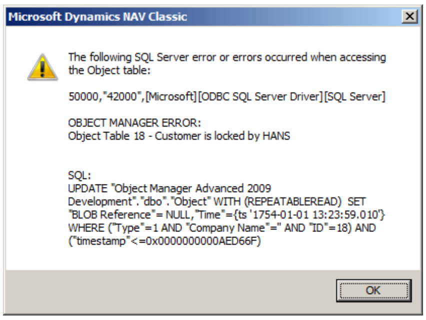


Table 18 Customer is not locked.

Integration Management

Only available if the SQL trigger is added to the object table. For more information see [section 4.1 - Add SQL Trigger to the Object Table.]()

  - Error if locked by other
    SQL will check if the object you change is locked by somebody else. If the object is locked by somebody else you will get the following error:



```
The following SQL Server error or errors occurred when accessing
the Object table:

50000,"42000",[Microsoft][ODBC SQL Server Driver][SQL Server]

OBJECT MANAGER ERROR:
Object Table 18 - Customer is locked by HANS


SQL:
UPDATE "Object Manager Advanced 2009
Development"."dbo"."Object" WITH (REPEATABLEREAD)  SET
"BLOB Reference"= NULL,"Time"={ts '1754-01-01 13:23:59.010'}
WHERE ("Type"=1 AND "Company Name"='' AND "ID"=18) AND
("timestamp"<=0x000000000AED66F)
```

SQL Trigger

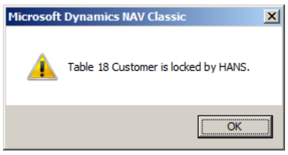


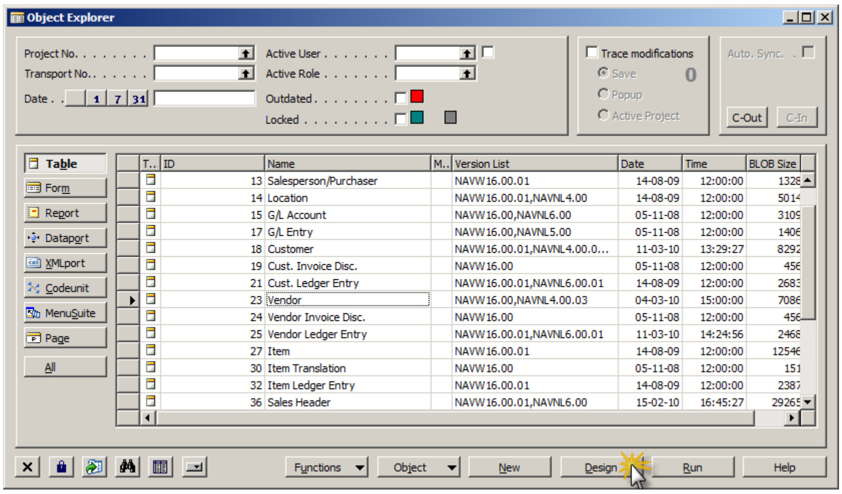
Table 18 Customer is locked by HANS.

Integration Management

- Lock Object
  If you save an object you will automatically lock the object. Only available if the SQL trigger is added to the object table. For more information see [section 4.1 - Add SQL Trigger to the Object Table](#).

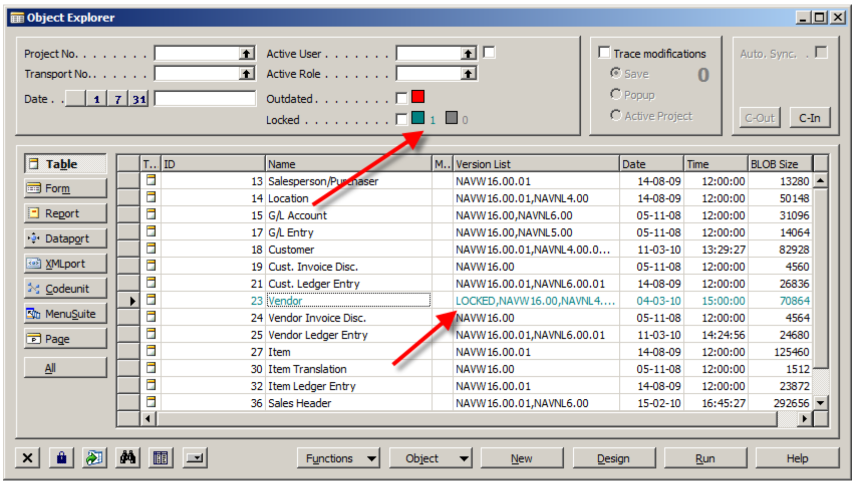
4.2.1. Lock Objects at Design

When designing with the Object Explorer.



T..	ID	Name	M..	Version List	Date	Time	BLOB Size
	13	Salesperson/Purchaser		NAVW16.00.01	14-08-09	12:00:00	1328
	14	Location		NAVW16.00.01,NAVNL4.00	14-08-09	12:00:00	5014
	15	G/L Account		NAVW16.00,NAVNL6.00	05-11-08	12:00:00	3109
	17	G/L Entry		NAVW16.00,NAVNL5.00	05-11-08	12:00:00	1406
	18	Customer		NAVW16.00.01,NAVNL4.00.0...	11-03-10	13:29:27	8292
	19	Cust. Invoice Disc.		NAVW16.00	05-11-08	12:00:00	456
	21	Cust. Ledger Entry		NAVW16.00.01,NAVNL6.00.01	14-08-09	12:00:00	2683
	23	Vendor		NAVW16.00,NAVNL4.00.03	04-03-10	15:00:00	7086
	24	Vendor Invoice Disc.		NAVW16.00	05-11-08	12:00:00	456
	25	Vendor Ledger Entry		NAVW16.00.01,NAVNL6.00.01	11-03-10	14:24:56	2468
	27	Item		NAVW16.00.01	14-08-09	12:00:00	12546
	30	Item Translation		NAVW16.00	05-11-08	12:00:00	151
	32	Item Ledger Entry		NAVW16.00.01	14-08-09	12:00:00	2387
	36	Sales Header		NAVW16.00.01,NAVNL6.00	15-02-10	16:45:27	29265

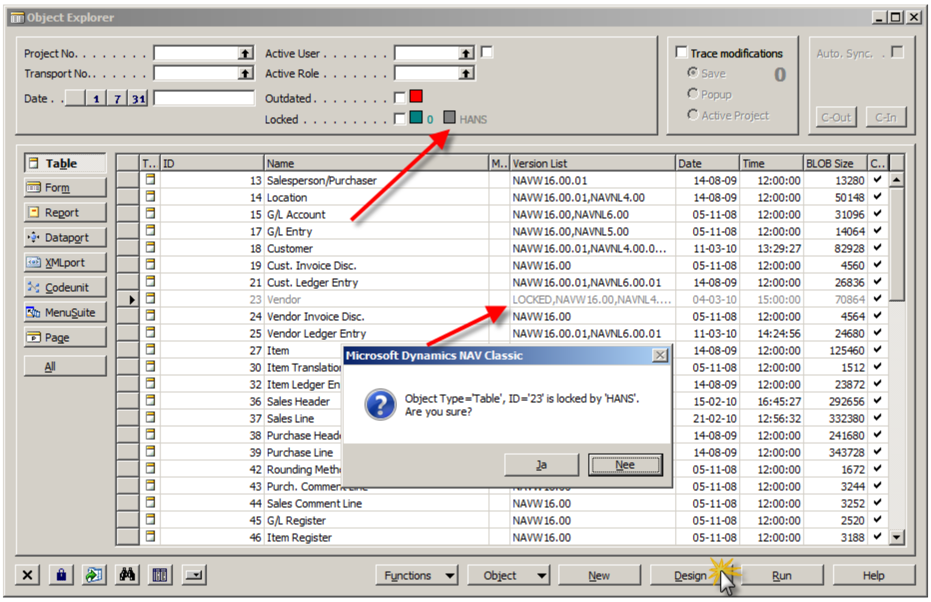
The selected object is locked automatically.



T..	ID	Name	M..	Version List	Date	Time	BLOB Size
	13	Salesperson/Purchaser		NAVW16.00.01	14-08-09	12:00:00	13280
	14	Location		NAVW16.00.01,NAVNL4.00	14-08-09	12:00:00	50148
	15	G/L Account		NAVW16.00,NAVNL6.00	05-11-08	12:00:00	31096
	17	G/L Entry		NAVW16.00,NAVNL5.00	05-11-08	12:00:00	14064
	18	Customer		NAVW16.00.01,NAVNL4.00.0...	11-03-10	13:29:27	82928
	19	Cust. Invoice Disc.		NAVW16.00	05-11-08	12:00:00	4560
	21	Cust. Ledger Entry		NAVW16.00.01,NAVNL6.00.01	14-08-09	12:00:00	26836
	23	Vendor		LOCKED,NAVW16.00,NAVNL4....	04-03-10	15:00:00	70864
	24	Vendor Invoice Disc.		NAVW16.00	05-11-08	12:00:00	4564
	25	Vendor Ledger Entry		NAVW16.00.01,NAVNL6.00.01	11-03-10	14:24:56	24680
	27	Item		NAVW16.00.01	14-08-09	12:00:00	125460
	30	Item Translation		NAVW16.00	05-11-08	12:00:00	1512
	32	Item Ledger Entry		NAVW16.00.01	14-08-09	12:00:00	23872
	36	Sales Header		NAVW16.00.01,NAVNL6.00	15-02-10	16:45:27	292656

IMPORTANT NOTE: When you lock an object your colleague developers will not get the assign popup when you modify the object. So if you are working in an environment with several developers locking is recommended.

Other users will see the object in a grey color and sees the user id next to the grey square in the header of the form. If another user wants to design that locked object, the user will get a warning.



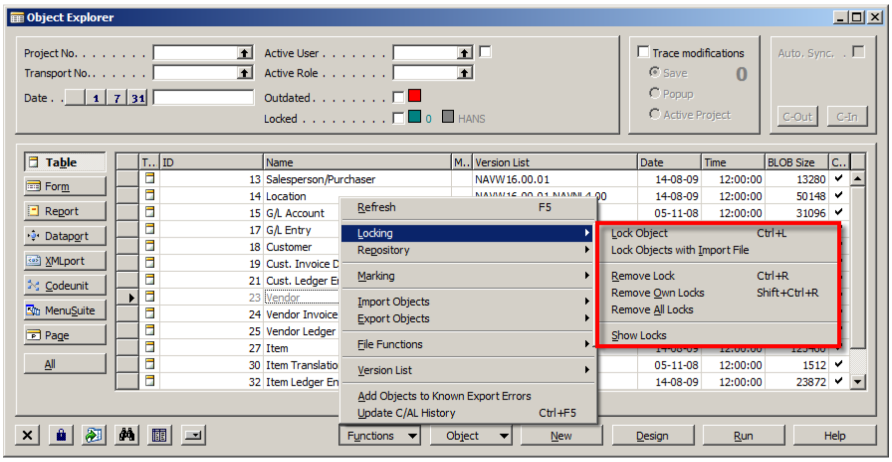
T..	ID	Name	M..	Version List	Date	Time	BLOB Size	C..
	13	Salesperson/Purchaser		NAVW16.00.01	14-08-09	12:00:00	13280	✓
	14	Location		NAVW16.00.01,NAVNL4.00	14-08-09	12:00:00	50148	✓
	15	G/L Account		NAVW16.00,NAVNL6.00	05-11-08	12:00:00	31096	✓
	17	G/L Entry		NAVW16.00,NAVNL5.00	05-11-08	12:00:00	14064	✓
	18	Customer		NAVW16.00.01,NAVNL4.00.0…	11-03-10	13:29:27	82928	✓
	19	Cust. Invoice Disc.		NAVW16.00	05-11-08	12:00:00	4560	✓
	21	Cust. Ledger Entry		NAVW16.00.01,NAVNL6.00.01	14-08-09	12:00:00	26836	✓
	23	Vendor		LOCKED,NAVW16.00,NAVNL4.…	04-03-10	15:00:00	70864	✓
	24	Vendor Invoice Disc.		NAVW16.00	05-11-08	12:00:00	4564	✓
	25	Vendor Ledger Entry		NAVW16.00.01,NAVNL6.00.01	11-03-10	14:24:56	24680	✓
	27	Item			14-08-09	12:00:00	125460	✓
	30	Item Translatio			05-11-08	12:00:00	1512	✓
	32	Item Ledger En			14-08-09	12:00:00	23872	✓
	36	Sales Header			15-02-10	16:45:27	292656	✓
	37	Sales Line			21-02-10	12:56:32	332380	✓
	38	Purchase Head			14-08-09	12:00:00	241680	✓
	39	Purchase Line			14-08-09	12:00:00	343728	✓
	42	Rounding Meth			05-11-08	12:00:00	1672	✓
	43	Purch. Commen			05-11-08	12:00:00	3244	✓
	44	Sales Comment Line		NAVW16.00	05-11-08	12:00:00	3252	✓
	45	G/L Register		NAVW16.00	05-11-08	12:00:00	2520	✓
	46	Item Register		NAVW16.00	05-11-08	12:00:00	3188	✓

Microsoft Dynamics NAV Classic

Object Type='Table', ID='23' is locked by 'HANS'.
Are you sure?

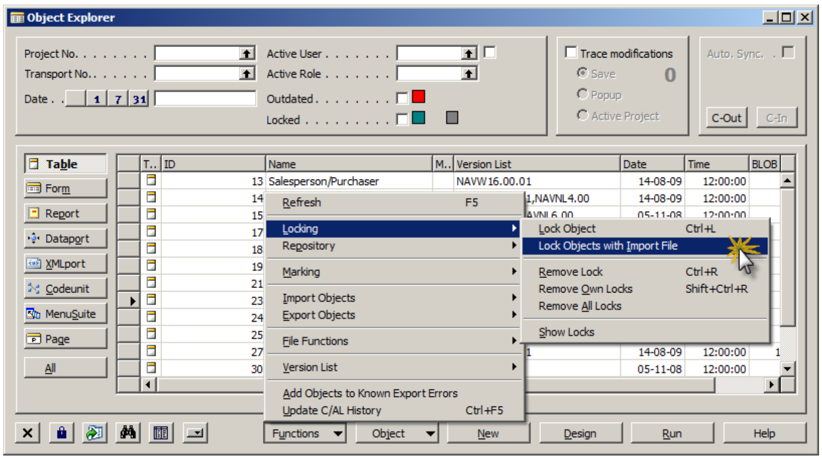
If the user chooses “Yes” the object will remain locked by the original user.

The locks can be remove in the Locking menu.



4.2.2. Lock Objects with Import File

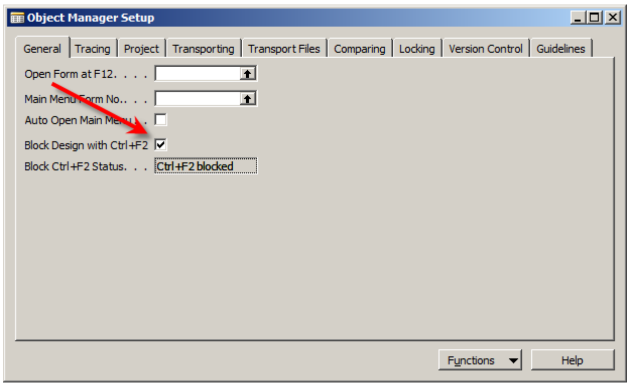
If you want to import an object file and you have the option “Error if Not Locked” enabled you will get an error. Therefore it is necessary to lock these objects before you import the object file. This can be done with the option Lock Objects with Import File in the Object Explorer.



Files with extension FOB, TXT, FIB and OBJ are supported.

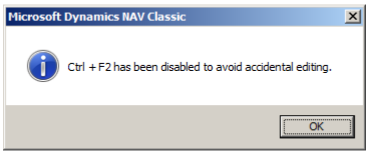
4.3. Block Design with Ctrl+F2

The warning that somebody has locked an object will not be shown when you run an object and then design it with the shortcut Ctrl+F2. This can be prevented with the “Block Design with Ctrl+F2” option in the setup.



Block Design with Ctrl+F2 ☑
Block Ctrl+F2 Status. . . Ctrl+F2 blocked

If you try to design an object with Ctrl+F2 you will get the following error.



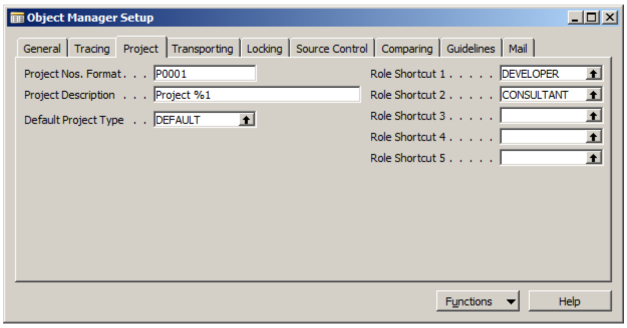
Ctrl + F2 has been disabled to avoid accidental editing.

You will need to run the Object Manager setup before you can use this functionality. For more information see section 2.1 - Installing.

5. Project

When you are developing solutions you can assign a number of objects to a project. To a project you can attach files and other information which is necessary for the consultant or developer involved in the project. You can also monitor the duration between modifications and status changes. Next to that you can assign actions to a project like removing data from tables.

5.1. Setup



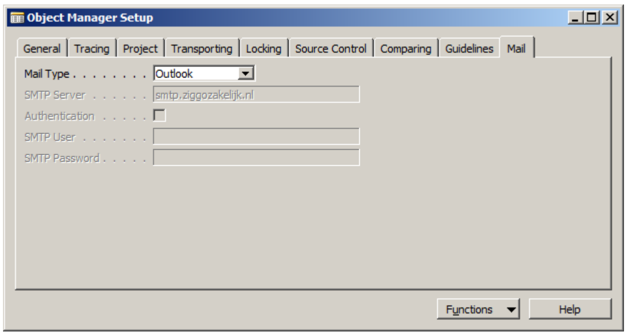
Project Nos. Format . . . P0001
Project Description . . . Project %1
Default Project Type . . DEFAULT
Role Shortcut 1 . . . . . DEVELOPER
Role Shortcut 2 . . . . . CONSULTANT
Role Shortcut 3 . . . . .
Role Shortcut 4 . . . . .
Role Shortcut 5 . . . . .

- **Project Nos. Format**

- **Project Description**
  %1 will be replaced by the "Project No.".
  You can use date expressions like: <Day> <Month Text> <Year4>.

- **Default Project Type**
  For more information about Project Types see chapter *5.2 - Project Type*.

- **Role Shortcuts**
  Here you select the roles that are visible in your project card.

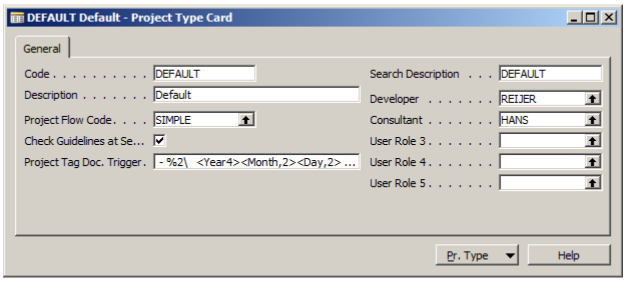


Mail Type . . . . Outlook
SMTP Server . . . . . . smtp.ziggozakelijk.nl
Authentication . . . . . ☐
SMTP User . . . .
SMTP Password . . . . .

- The emails that are sent in the project and transport module can be send by outlook or with a SMTP server.

5.2. Project Type

Open Setup > Project Types



DEFAULT Default - Project Type Card

Code	DEFAULT	Search Description	DEFAULT
Description	Default	Developer	REIJER
Project Flow Code	SIMPLE	Consultant	HANS
Check Guidelines at Se...	☑	User Role 3	
Project Tag Doc. Trigger	- %2\ <Year4><Month,2><Day,2> ...	User Role 4	
		User Role 5	

Here you can define different types of projects to differentiate for example between hotfixes and longterm projects.

- **Project Flow Code**
  For more information about project flows see section 5.5 Project Flow.

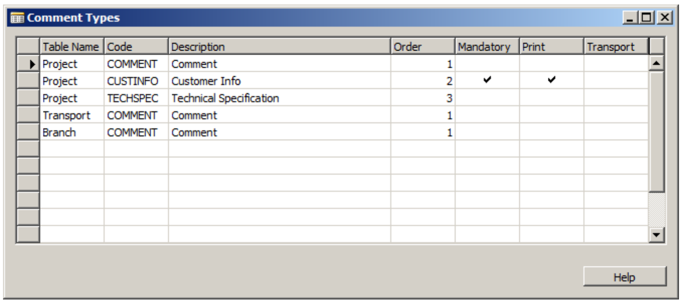
- **Check Guidelines at Set Ready Project**
  This setting will run a guidelines check on the objects in the project when the status of the project is set to ready.
  See more information see chapter 18 - Check Guidelines.

- **Project Tag Doc. Trigger**
  Here you can define the text for the automatic documentation trigger insert. For more information see section 5.7 - Add Project Tag to Documentation Trigger.

- **Default User Roles**
  Here you can define the default users if this project type.

5.3. Comment Types

Open Setup > Comment Types.



Comment Types

Table Name	Code	Description	Order	Mandatory	Print	Transport
Project	COMMENT	Comment	1			
Project	CUSTINFO	Customer Info	2	✓	✓	
Project	TECHSPEC	Technical Specification	3			
Transport	COMMENT	Comment	1			
Branch	COMMENT	Comment	1			

Here you can define Comment Groups for the comments on a project.

- **Order**

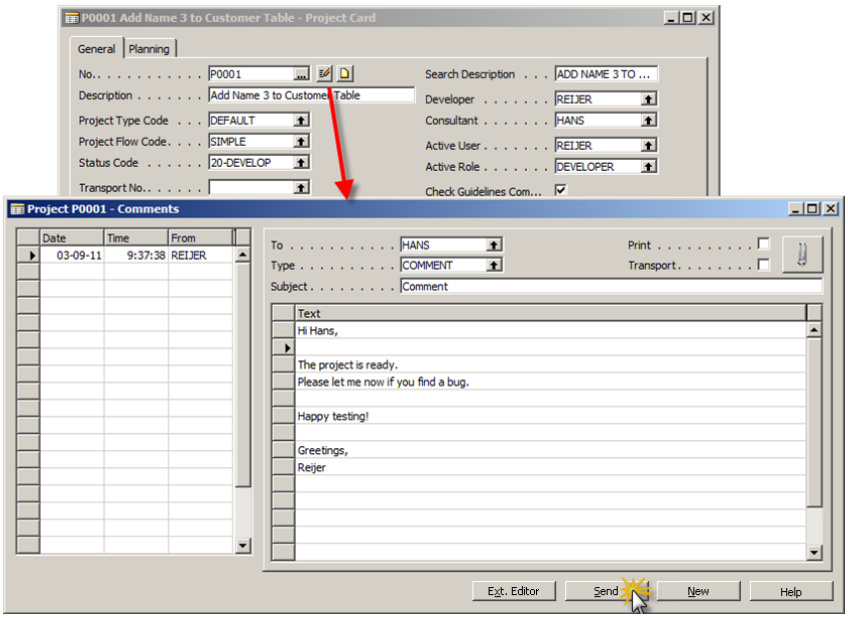
This is the order in which the comments will be printed on the project and transport reports.

- **Mandatory**
  Transport will be only possible if a comment is present.

- **Print**
  Comment(s) will be printed on the project and transport report.

- **Transport**
  Comment(s) will be transported to your customer database.

Use the “Comments” window to write down your comments using either the Text lines or an external editor by pressing ”Ext. Editor”. You can also send a comment pressing “Send” provided you entered a user with an associated e-mail address in the “To” field.



No.. . . . . P0001
Description . . . . Add Name 3 to Customer Table
Project Type Code . . . DEFAULT
Project Flow Code. . . . SIMPLE
Status Code 20-DEVELOP
Transport No.. . . . . .
Search Description . . . ADD NAME 3 TO ...
Developer . . . . REIJER
Consultant . . . . HANS
Active User . . . . REIJER
Active Role . . . . DEVELOPER
Check Guidelines Com... ☑

Date	Time	From
03-09-11	9:37:38	REIJER

To . . . . HANS
Type . . . . COMMENT
Subject . . . . Comment
Print . . . . ☐
Transport. . . . . ☐

Text

Hi Hans,

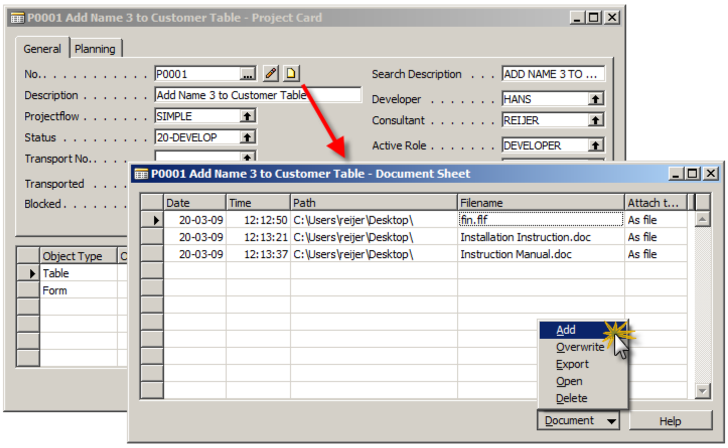
The project is ready.
Please let me now if you find a bug.

Happy testing!

Greetings,
Reijer

5.4. Documents

You can add files to a project. For example a customer license or an installation instruction. With the button Document you can add, overwrite and delete files which are of importance to the project.



P0001 Add Name 3 to Customer Table - Project Card

No.: P0001
Description: Add Name 3 to Customer Table
Projectflow: SIMPLE
Status: 20-DEVELOP
Transport No.:
Transported
Blocked
Search Description: ADD NAME 3 TO ...
Developer: HANS
Consultant: REIJER
Active Role: DEVELOPER

Object Type
Table
Form

P0001 Add Name 3 to Customer Table - Document Sheet

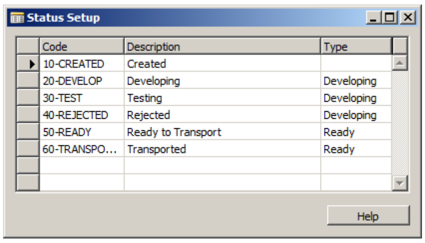
Date	Time	Path	Filename	Attach t...
20-03-09	12:12:50	C:\Users\reijer\Desktop\	fin.flf	As file
20-03-09	12:13:21	C:\Users\reijer\Desktop\	Installation Instruction.doc	As file
20-03-09	12:13:37	C:\Users\reijer\Desktop\	Instruction Manual.doc	As file

- **Attach to Transport**
  - As File: File will be saved in the transport folder as a separate file.
  - In FIB: The file is merged into the FIB file (see chapter 6 - Transport).

5.5. Project Flow

A project flow is a collection of project statuses. You can add a project flow to a project so you can keep track of the progress and which user is responsible for a specific project status. Default there are two flows in the Object Manager but you can add as many flows as you like. You can also create more statuses in the “Status Setup”.

Setup > “Status Setup”



Status Setup

Code	Description	Type
10-CREATED	Created	
20-DEVELOP	Developing	Developing
30-TEST	Testing	Developing
40-REJECTED	Rejected	Developing
50-READY	Ready to Transport	Ready
60-TRANSPO...	Transported	Ready

Here you can add an extra status with a brief description and type.

- **Type**
  This is the type of the status. It has the following options:
  - Developing: You can assign modifications to projects that have this status.
  - Ready: You can add projects that have this status to a transport.

Select Setup > Flows to open the Flow Card

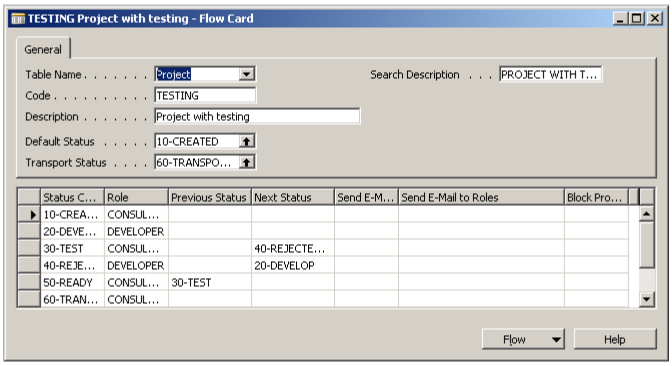


Table Name . . . . Project
Code . . . . TESTING
Description . . . . Project with testing
Default Status 10-CREATED
Transport Status . . . . 60-TRANSPO...
Search Description . . . PROJECT WITH T...

Status C...	Role	Previous Status	Next Status	Send E-M...	Send E-Mail to Roles	Block Pro...
10-CREA...	CONSUL...					
20-DEVE...	DEVELOPER					
30-TEST	CONSUL...		40-REJECTE...			
40-REJE...	DEVELOPER		20-DEVELOP			
50-READY	CONSUL...	30-TEST				
60-TRAN...	CONSUL...					

- **Default Status**
  Default status for each new project.

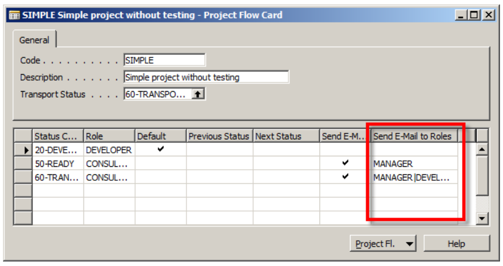
- **Transport Status**
  The project will get this status when it is transported to your customer database.

- **Previous Status**
  User can go back to this status.

- **Next Status**
  User can only choose the next statuses defined here.

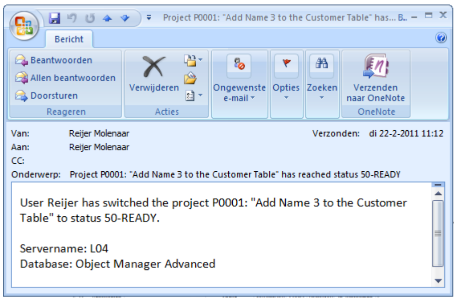
- **Send E-Mail**
  When this status is reached an e-mail will be sent to the active user. You will be prompted if you want to send that e-mail.

- **Send E-Mail to Roles**
  When one or more roles are filled in here and this status is reached the user with this role is also sent an e-mail.



Code . . . . SIMPLE
Description . . . . Simple project without testing
Transport Status . . . . 60-TRANSPO...

Status C...	Role	Default	Previous Status	Next Status	Send E-M...	Send E-Mail to Roles
20-DEVE...	DEVELOPER	✓				
50-READY	CONSUL...				✓	MANAGER
60-TRAN...	CONSUL...				✓	MANAGER|DEVEL...



Van: Reijer Molenaar
Aan: Reijer Molenaar
CC:
Onderwerp: Project P0001: "Add Name 3 to the Customer Table" has reached status 50-READY
Verzonden: di 22-2-2011 11:12

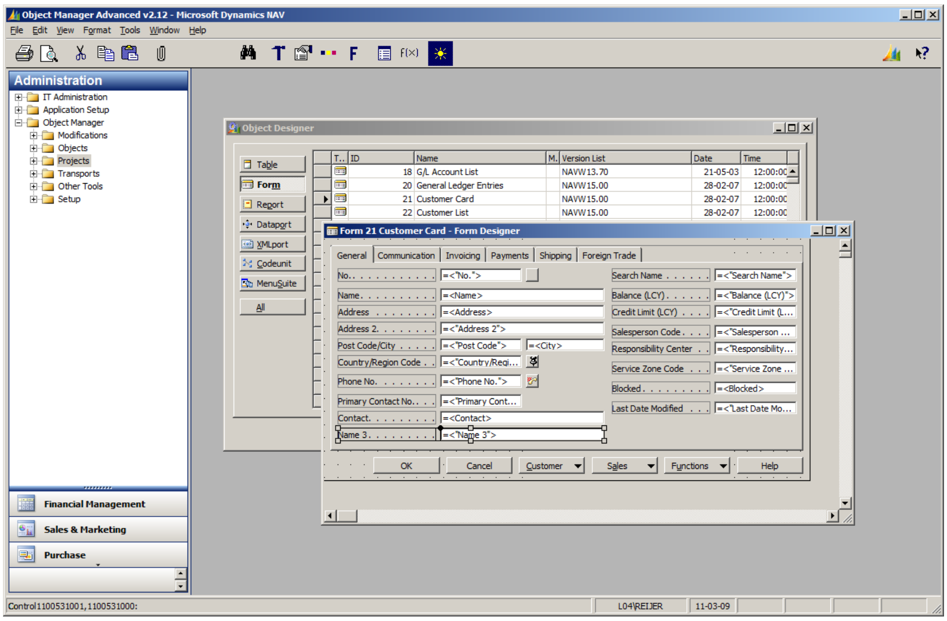
User Reijer has switched the project P0001: "Add Name 3 to the Customer Table" to status 50-READY.

Servername: L04
Database: Object Manager Advanced

- **Block Project**
  When this status is reached, and “Block Project” is check marked, the project will be blocked.

5.6. *Assign Modifications to a Project*

You can add multiple modifications to one project.



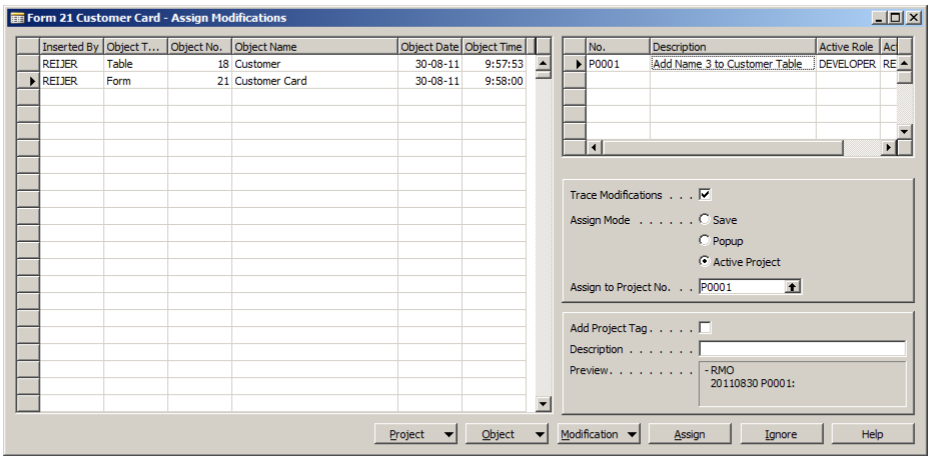
T..	ID	Name	M.	Version List	Date	Time
	18	G/L Account List		NAVW13.70	21-05-03	12:00:0C
	20	General Ledger Entries		NAVW15.00	28-02-07	12:00:0C
	21	Customer Card		NAVW15.00	28-02-07	12:00:0C
	22	Customer List		NAVW15.00	28-02-07	12:00:0C

No. . . . . =<"No.">
Name. . . . . =<Name>
Address . . . . =<Address>
Address 2. . . . . =<"Address 2">
Post Code/City . . . . . =<"Post Code"> =<City>
Country/Region Code . . =<"Country/Regi…
Phone No. . . . . =<"Phone No.">
Primary Contact No.. . . =<"Primary Cont…
Contact. . . . . =<Contact>
Name 3. . . . . =<"Name 3">
Search Name . . . . . . =<"Search Name">
Balance (LCY). . . . . . =<"Balance (LCY)">
Credit Limit (LCY) . . . . =<"Credit Limit (L…
Salesperson Code. . . . =<"Salesperson …
Responsibility Center . . =<"Responsibility…
Service Zone Code . . . =<"Service Zone …
Blocked. . . . . =<Blocked>
Last Date Modified . . . =<"Last Date Mo…

Add, for example a new field “Name 3” to the Customer table and add this field as a control to the “Customer Card”. Save these objects.

Open Modifications > Assign Modifications.

Select the objects you want to assign. Press button Assign.



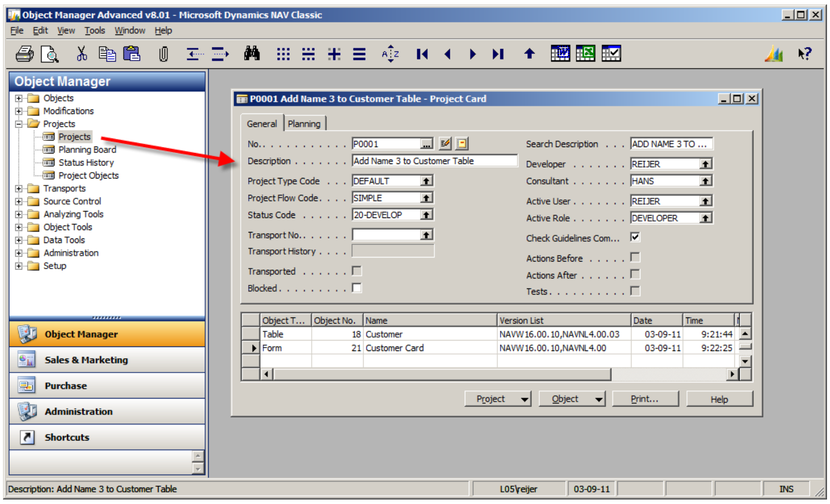
Inserted By	Object T…	Object No.	Object Name	Object Date	Object Time
REIJER	Table	18	Customer	30-08-11	9:57:53
REIJER	Form	21	Customer Card	30-08-11	9:58:00

No.	Description	Active Role	Ac
P0001	Add Name 3 to Customer Table	DEVELOPER	RE

Trace Modifications . . . ☑
Assign Mode . . . . . . ○ Save
○ Popup
● Active Project
Assign to Project No. . . P0001

Add Project Tag . . . . . ☐
Description . . . .
Preview. . . . . - RMO
20110830 P0001:

Now you can see that the objects are assigned to the project.



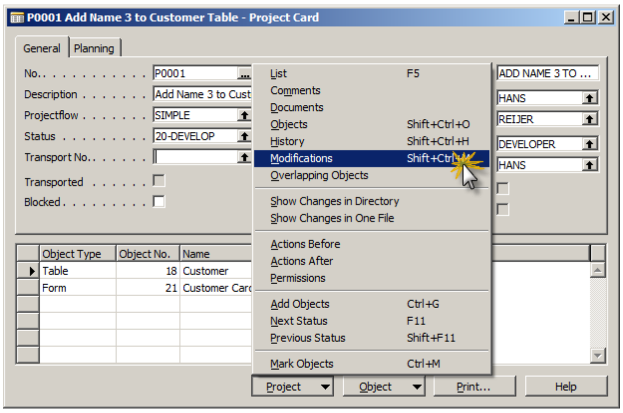
P0001 Add Name 3 to Customer Table - Project Card

No.	P0001	Search Description	ADD NAME 3 TO ...
Description	Add Name 3 to Customer Table	Developer	REIJER
Project Type Code	DEFAULT	Consultant	HANS
Project Flow Code	SIMPLE	Active User	REIJER
Status Code	20-DEVELOP	Active Role	DEVELOPER
Transport No.		Check Guidelines Com...	☑
Transport History		Actions Before	☐
Transported	☐	Actions After	☐
Blocked	☐	Tests	☐

Object T...	Object No.	Name	Version List	Date	Time
Table	18	Customer	NAVW16.00.10,NAVNL4.00.03	03-09-11	9:21:44
Form	21	Customer Card	NAVW16.00.10,NAVNL4.00	03-09-11	9:22:25

Description: Add Name 3 to Customer Table | L05\reijer | 03-09-11 | INS

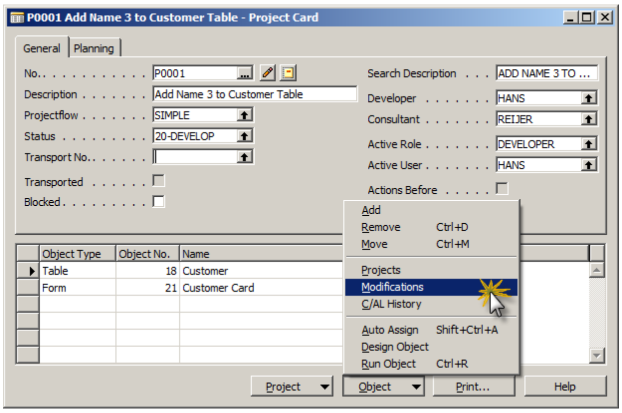
To see all modifications on a project, press Modification under button Object or Project in the “Project Card”.



No.	P0001	ADD NAME 3 TO ...
Description	Add Name 3 to Cust	HANS
Projectflow	SIMPLE	REIJER
Status	20-DEVELOP	DEVELOPER
Transport No.		HANS
Transported	☐	☐
Blocked	☐	☐

Object Type	Object No.	Name
Table	18	Customer
Form	21	Customer Card

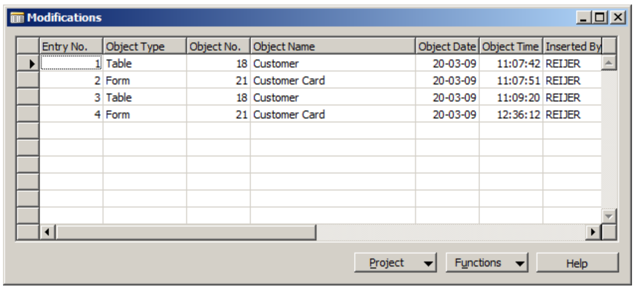
All modifications made on a project



No.	P0001	Search Description	ADD NAME 3 TO ...
Description	Add Name 3 to Customer Table	Developer	HANS
Projectflow	SIMPLE	Consultant	REIJER
Status	20-DEVELOP	Active Role	DEVELOPER
Transport No.		Active User	HANS
Transported	☐	Actions Before	☐
Blocked	☐		

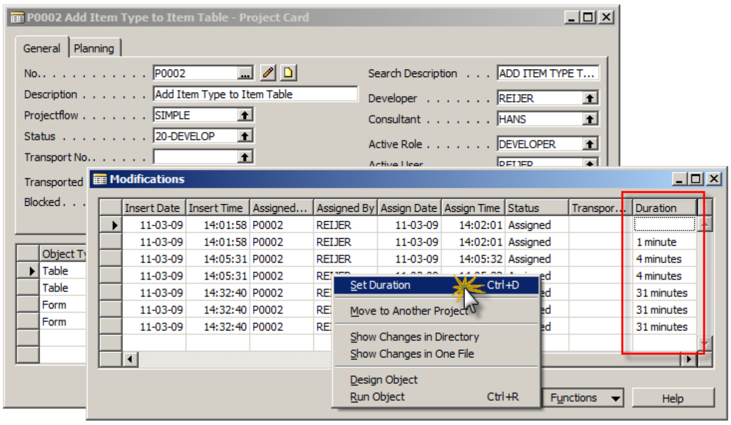
Object Type	Object No.	Name
Table	18	Customer
Form	21	Customer Card

All modifications made on the object in the project



Entry No.	Object Type	Object No.	Object Name	Object Date	Object Time	Inserted By
1	Table	18	Customer	20-03-09	11:07:42	REIJER
2	Form	21	Customer Card	20-03-09	11:07:51	REIJER
3	Table	18	Customer	20-03-09	11:09:20	REIJER
4	Form	21	Customer Card	20-03-09	12:36:12	REIJER

In the form Modifications you can set duration from a chosen point. You can use this when you want to know how long you have worked on a particular project.



No.	P0002	Search Description	ADD ITEM TYPE T…
Description	Add Item Type to Item Table	Developer	REIJER
Projectflow	SIMPLE	Consultant	HANS
Status	20-DEVELOP	Active Role	DEVELOPER
Transport No.		Active User	REIJER
Transported			
Blocked			

Object Ty
Table
Table
Form
Form

Insert Date	Insert Time	Assigned…	Assigned By	Assign Date	Assign Time	Status	Transpor…	Duration
11-03-09	14:01:58	P0002	REIJER	11-03-09	14:02:01	Assigned		
11-03-09	14:01:58	P0002	REIJER	11-03-09	14:02:01	Assigned		1 minute
11-03-09	14:05:31	P0002	REIJER	11-03-09	14:05:32	Assigned		4 minutes
11-03-09	14:05:31	P0002	REI			ed		4 minutes
11-03-09	14:32:40	P0002	REI			ed		31 minutes
11-03-09	14:32:40	P0002	REI			ed		31 minutes
11-03-09	14:32:40	P0002	REI			ed		31 minutes

5.7. Add Project Tag to Documentation Trigger

It is possible to add a predefined string to the documentation trigger of an object. This can be done when you assign an object to a project or with a chosen set of objects in the project card.

On the Project Type card you can setup the format of the string that has to be added to the objects. Default the Object Manager uses the following format:

```
<Year4><Month,2><Day,2> %1 %2: %3
```

You can see that it is possible to add date expressions and 3 percentages that will be replaced with the following.

%1: Project No.
%2: Initials of the active user
%3: Description of the modification

Press Add Project Tag and type a description.

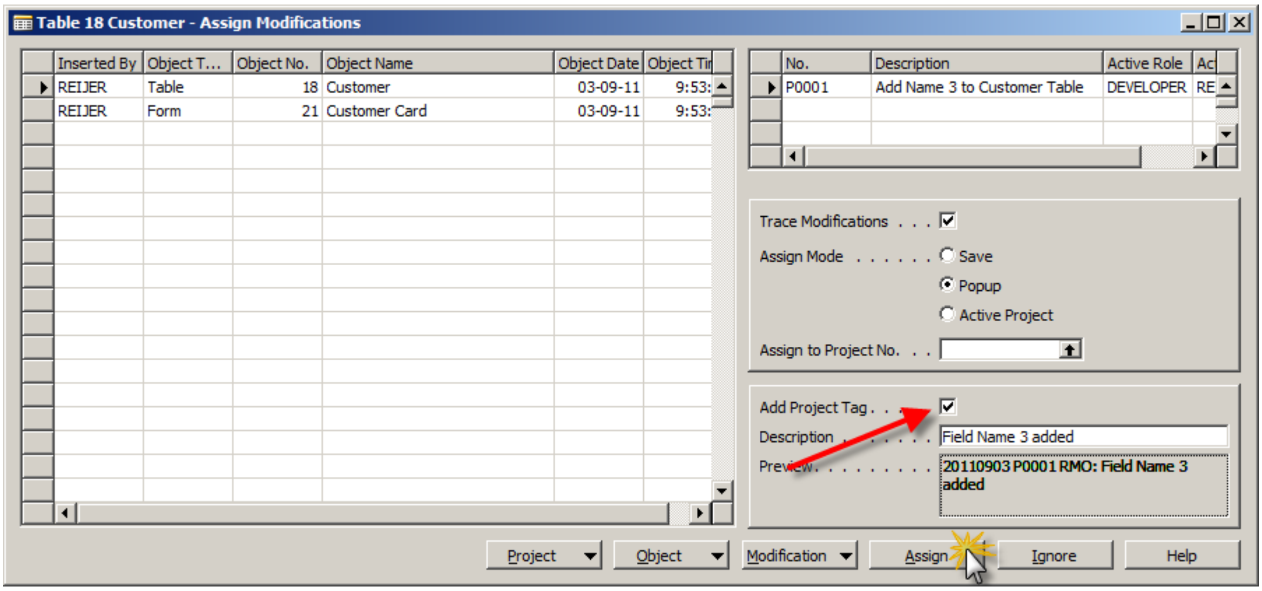


Table 18 Customer - Assign Modifications

Inserted By	Object T...	Object No.	Object Name	Object Date	Object Ti
REIJER	Table	18	Customer	03-09-11	9:53:
REIJER	Form	21	Customer Card	03-09-11	9:53:

No.	Description	Active Role	Ac
P0001	Add Name 3 to Customer Table	DEVELOPER	RE

Trace Modifications . . . ☑

Assign Mode . . . . . . ○ Save

● Popup

○ Active Project

Assign to Project No. . .

Add Project Tag . . ☑

Description . . . . . . Field Name 3 added

Preview. . . . . 20110903 P0001 RMO: Field Name 3 added

The tag will be added to the objects that are assigned.

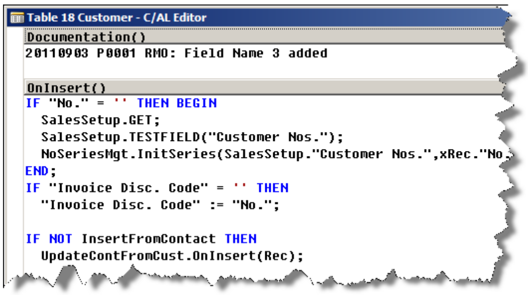
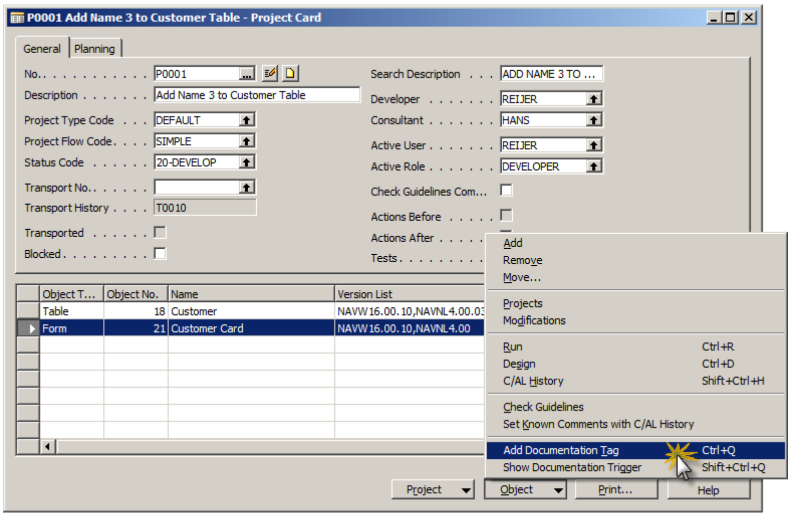


Table 18 Customer - C/AL Editor

```
Documentation()
20110903 P0001 RMO: Field Name 3 added

OnInsert()
IF "No." = '' THEN BEGIN
  SalesSetup.GET;
  SalesSetup.TESTFIELD("Customer Nos.");
  NoSeriesMgt.InitSeries(SalesSetup."Customer Nos.",xRec."No.
END;
IF "Invoice Disc. Code" = '' THEN
  "Invoice Disc. Code" := "No.";

IF NOT InsertFromContact THEN
  UpdateContFromCust.OnInsert(Rec);
```

It is also possible to add a tag to selected objects in the project card.



P0001 Add Name 3 to Customer Table - Project Card

No.. . . . . P0001

Description . . . . Add Name 3 to Customer Table

Project Type Code . . . DEFAULT

Project Flow Code. . . . SIMPLE

Status Code 20-DEVELOP

Transport No.. . . . . .

Transport History T0010

Transported . . . . . . ☐

Blocked. . . . . ☐

Search Description . . . ADD NAME 3 TO ...

Developer . . . . REIJER

Consultant . . . . HANS

Active User . . . . REIJER

Active Role . . . . DEVELOPER

Check Guidelines Com... ☐

Actions Before . . . . . ☐

Actions After . . . . .

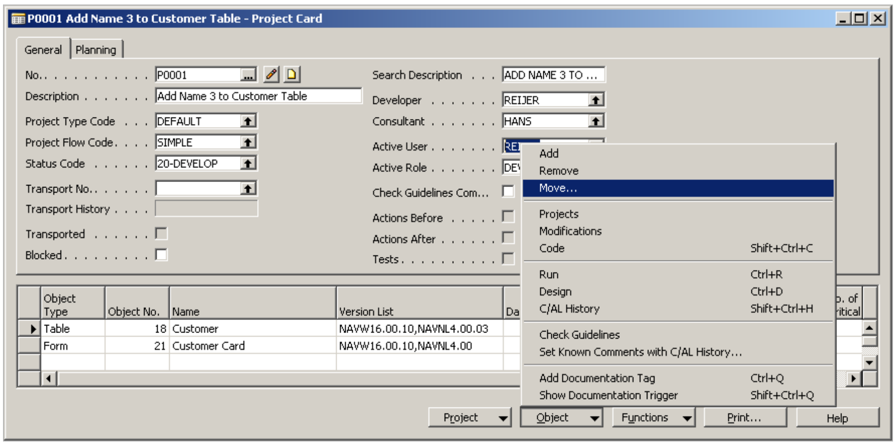
Tests. . . . .

Object T...	Object No.	Name	Version List
Table	18	Customer	NAVW16.00.10,NAVNL4.00.03
Form	21	Customer Card	NAVW16.00.10,NAVNL4.00

Add
Remove
Move...
Projects
Modifications
Run Ctrl+R
Design Ctrl+D
C/AL History Shift+Ctrl+H
Check Guidelines
Set Known Comments with C/AL History
Add Documentation Tag Ctrl+Q
Show Documentation Trigger Shift+Ctrl+Q
Project
Object
Print...
Help

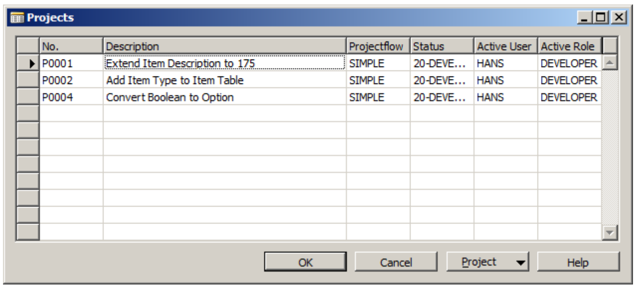
5.8. Move Objects and Modifications to another Project

With this function you move objects to another project. And if there are any modifications present the Object Manager will ask to move them too. This is used when you assigned modifications to the wrong project.

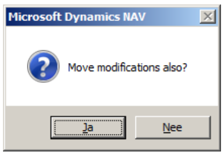

P0001 Add Name 3 to Customer Table - Project Card

No. P0001
Description Add Name 3 to Customer Table
Project Type Code DEFAULT
Project Flow Code SIMPLE
Status Code 20-DEVELOP
Transport No.
Transport History
Transported ☐
Blocked ☐
Search Description ADD NAME 3 TO ...
Developer REIJER
Consultant HANS
Active User REI
Active Role DEV
Check Guidelines Com... ☐
Actions Before ☐
Actions After ☐
Tests ☐

Object Type	Object No.	Name	Version List
Table	18	Customer	NAVW16.00.10,NAVNL4.00.03
Form	21	Customer Card	NAVW16.00.10,NAVNL4.00


Projects

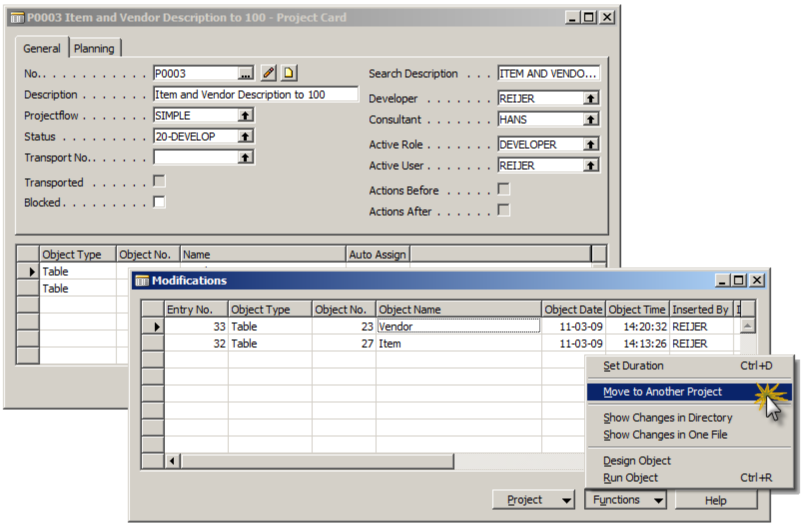
No.	Description	Projectflow	Status	Active User	Active Role
P0001	Extend Item Description to 175	SIMPLE	20-DEVE...	HANS	DEVELOPER
P0002	Add Item Type to Item Table	SIMPLE	20-DEVE...	HANS	DEVELOPER
P0004	Convert Boolean to Option	SIMPLE	20-DEVE...	HANS	DEVELOPER


Microsoft Dynamics NAV

Move modifications also?

When selecting Yes you also move all modifications of the object.

You can also move a particular modification to another project. To do this select the move option in the Modifications form.


P0003 Item and Vendor Description to 100 - Project Card

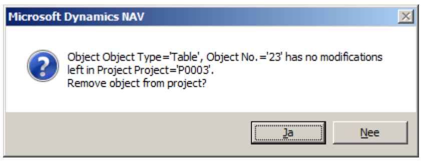
No.	P0003
Description	Item and Vendor Description to 100
Projectflow	SIMPLE
Status	20-DEVELOP
Transport No.	
Transported	☐
Blocked	☐
Search Description	ITEM AND VENDO...
Developer	REIJER
Consultant	HANS
Active Role	DEVELOPER
Active User	REIJER
Actions Before	☐
Actions After	☐

Object Type	Object No.	Name	Auto Assign
Table			
Table			

Modifications

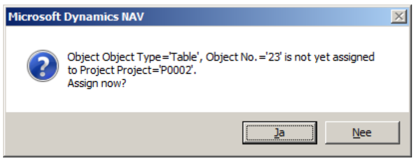
Entry No.	Object Type	Object No.	Object Name	Object Date	Object Time	Inserted By
33	Table	23	Vendor	11-03-09	14:20:32	REIJER
32	Table	27	Item	11-03-09	14:13:26	REIJER

In the Modifications form select the modification you want to move to another project. If there are no more modifications left in the original project you will get the following message.


Microsoft Dynamics NAV

Object Object Type='Table', Object No.='23' has no modifications left in Project Project='P0003'.
Remove object from project?

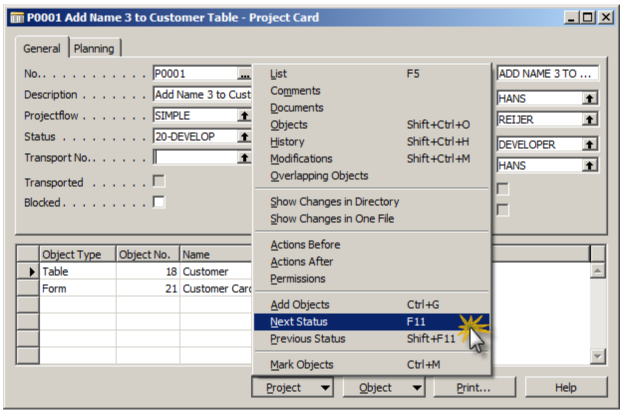
If the object is not yet assigned in the other project you will get the following message.


Microsoft Dynamics NAV

Object Object Type='Table', Object No.='23' is not yet assigned to Project Project='P0002'.
Assign now?

5.9. Change Status of a Project

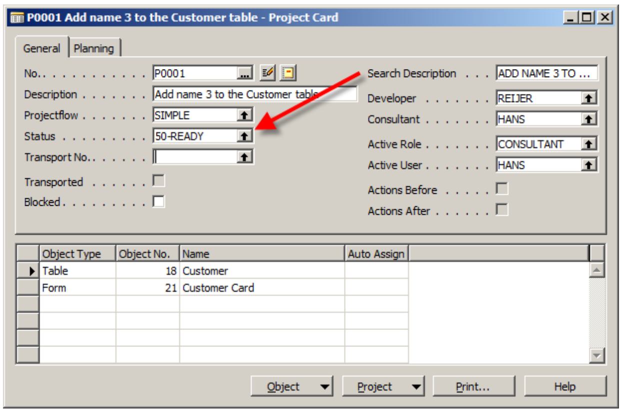
Project > “Next Status” or “Previous Status” to go back.



P0001 Add Name 3 to Customer Table - Project Card

No.. . . . . P0001
Description . . . . Add Name 3 to Cust
Projectflow . . . . SIMPLE
Status 20-DEVELOP
Transport No.. . . . . .
Transported . . . . . . ☐
Blocked. . . . . ☐

Object Type	Object No.	Name
Table	18	Customer
Form	21	Customer Card



P0001 Add name 3 to the Customer table - Project Card

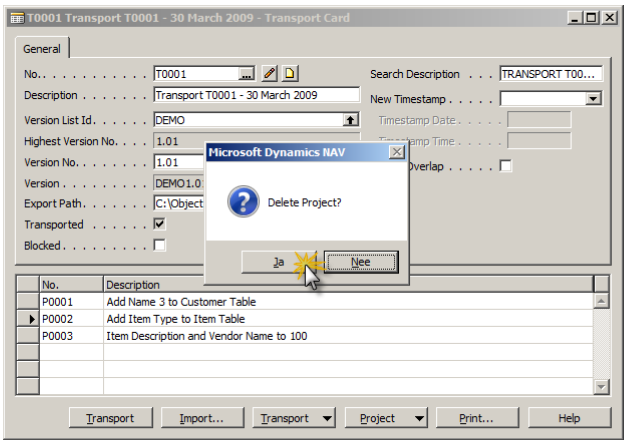
No.. . . . . P0001
Description . . . . Add name 3 to the Customer table
Projectflow . . . . SIMPLE
Status 50-READY
Transport No.. . . . . .
Transported . . . . . . ☐
Blocked. . . . . ☐
Search Description . . . ADD NAME 3 TO ...
Developer . . . . REIJER
Consultant . . . . HANS
Active Role . . . . CONSULTANT
Active User . . . . HANS
Actions Before . . . . . ☐
Actions After . . . . . . ☐

Object Type	Object No.	Name	Auto Assign
Table	18	Customer	
Form	21	Customer Card	

5.10. Reset Project Status

Sometimes you want to re-use a project. If e.g. you have to do some rework on a project. When you assign a modification to a project that is already transported you normally get the warning: “This project is already transported. Are you sure?“.

You can get rid of this warning by using the “Reset Status” option at the project card. First remove the project from the transport. You can do this by clearing the “Transport No.” at the project card or deleting the project from the transport card.



T0001 Transport T0001 - 30 March 2009 - Transport Card

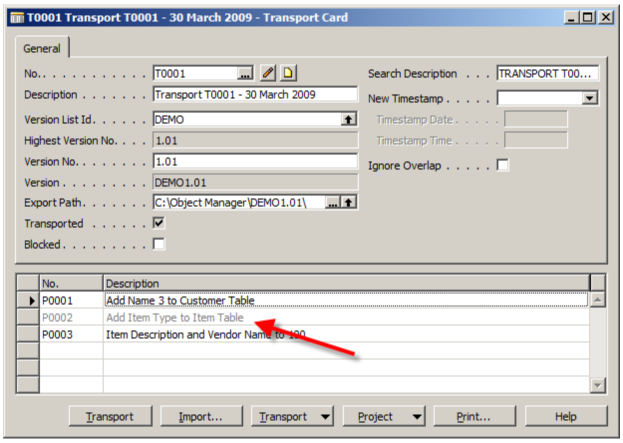
No.. . . . . T0001
Description . . . . Transport T0001 - 30 March 2009
Version List Id. . . . . . DEMO
Highest Version No. . . . 1.01
Version No. 1.01
Version DEMO1.0
Export Path. . . . . . . C:\Object
Transported . . . . . . ☑
Blocked. . . . . ☐
Search Description . . . TRANSPORT T00...
New Timestamp . . . . .
Timestamp Date . . . . .
Timestamp Time . . . . .
Overlap . . . . . ☐

Microsoft Dynamics NAV

Delete Project?

No.	Description
P0001	Add Name 3 to Customer Table
P0002	Add Item Type to Item Table
P0003	Item Description and Vendor Name to 100

You now see that the color of the project is grey. This means that the project is transported in this transport but currently not included in this transport.

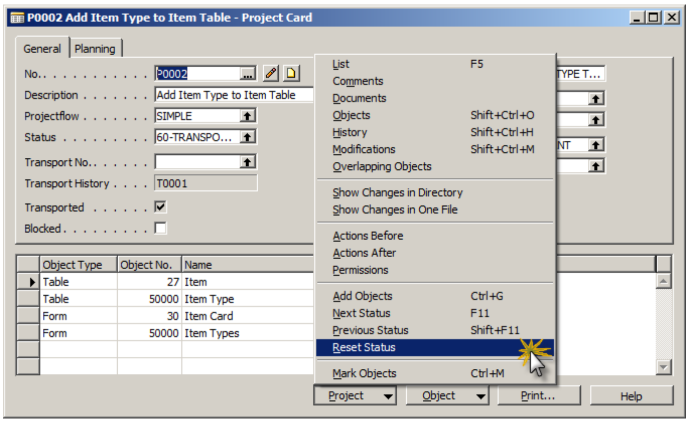


T0001 Transport T0001 - 30 March 2009 - Transport Card

No. T0001
Description Transport T0001 - 30 March 2009
Version List Id. DEMO
Highest Version No. 1.01
Version No. 1.01
Version DEMO1.01
Export Path C:\Object Manager\DEMO1.01\
Transported ☑
Blocked ☐
Search Description TRANSPORT T00...
New Timestamp
Timestamp Date
Timestamp Time
Ignore Overlap ☐

No.	Description
P0001	Add Name 3 to Customer Table
P0002	Add Item Type to Item Table
P0003	Item Description and Vendor Name to 100

Now it is possible to click the “Reset Project” option at the project card.



P0002 Add Item Type to Item Table - Project Card

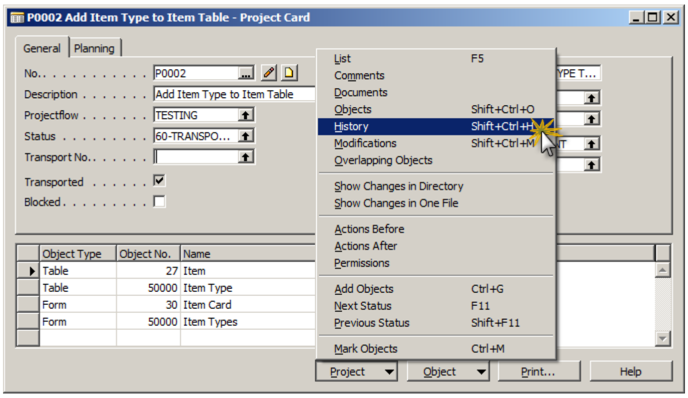
No. P0002
Description Add Item Type to Item Table
Projectflow SIMPLE
Status 60-TRANSPO...
Transport No.
Transport History T0001
Transported ☑
Blocked ☐

Object Type	Object No.	Name
Table	27	Item
Table	50000	Item Type
Form	30	Item Card
Form	50000	Item Types

The “Transported” Boolean is set to false for this project and it got the default status of the project flow. Because the “Transport No.” of the project is cleared it can be included in another transport.

5.11. Project History

In the form “Project History” you have an overview of all the statuses that a project is gone through.

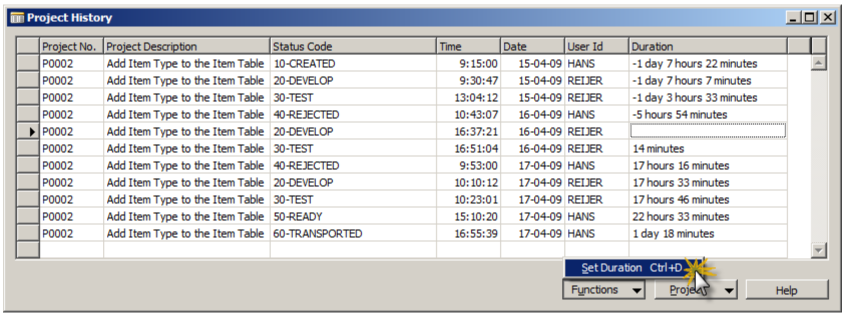


P0002 Add Item Type to Item Table - Project Card

No.	P0002
Description	Add Item Type to Item Table
Projectflow	TESTING
Status	60-TRANSPO...
Transport No.	
Transported	☑
Blocked	☐

Object Type	Object No.	Name
Table	27	Item
Table	50000	Item Type
Form	30	Item Card
Form	50000	Item Types

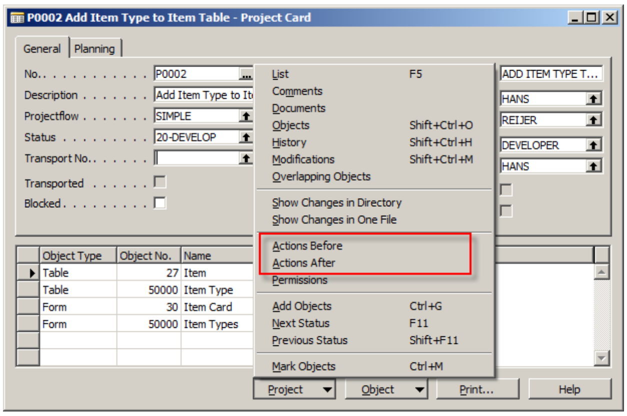
Set the duration if you want to know how long you worked on the project. In this example you see that you have developed 14 minutes on the 16th of April.



Project History

Project No.	Project Description	Status Code	Time	Date	User Id	Duration
P0002	Add Item Type to the Item Table	10-CREATED	9:15:00	15-04-09	HANS	-1 day 7 hours 22 minutes
P0002	Add Item Type to the Item Table	20-DEVELOP	9:30:47	15-04-09	REIJER	-1 day 7 hours 7 minutes
P0002	Add Item Type to the Item Table	30-TEST	13:04:12	15-04-09	REIJER	-1 day 3 hours 33 minutes
P0002	Add Item Type to the Item Table	40-REJECTED	10:43:07	16-04-09	HANS	-5 hours 54 minutes
P0002	Add Item Type to the Item Table	20-DEVELOP	16:37:21	16-04-09	REIJER	
P0002	Add Item Type to the Item Table	30-TEST	16:51:04	16-04-09	REIJER	14 minutes
P0002	Add Item Type to the Item Table	40-REJECTED	9:53:00	17-04-09	HANS	17 hours 16 minutes
P0002	Add Item Type to the Item Table	20-DEVELOP	10:10:12	17-04-09	REIJER	17 hours 33 minutes
P0002	Add Item Type to the Item Table	30-TEST	10:23:01	17-04-09	REIJER	17 hours 46 minutes
P0002	Add Item Type to the Item Table	50-READY	15:10:20	17-04-09	HANS	22 hours 33 minutes
P0002	Add Item Type to the Item Table	60-TRANSPORTED	16:55:39	17-04-09	HANS	1 day 18 minutes

5.12. Add Actions to a Project



P0002 Add Item Type to Item Table - Project Card

No.	P0002
Description	Add Item Type to It
Projectflow	SIMPLE
Status	20-DEVELOP
Transport No.	
Transported	☐
Blocked	☐

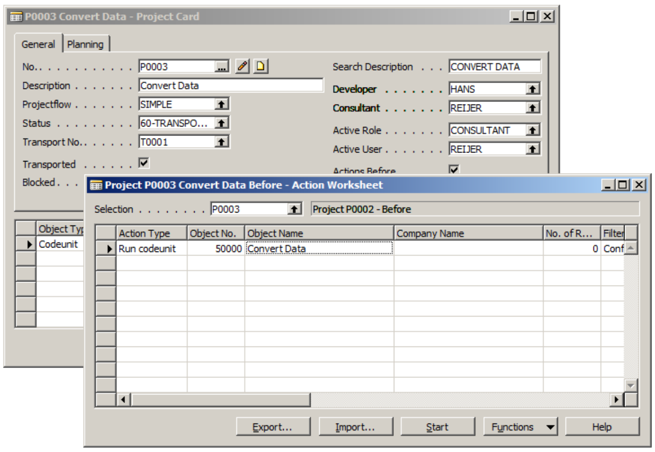
Object Type	Object No.	Name
Table	27	Item
Table	50000	Item Type
Form	30	Item Card
Form	50000	Item Types

Actions Before and Actions After

With this tool you can add certain actions to a project which have to be executed in the customer database. Actions before will be executed before the objects are imported in the customer database. Actions after are executed after the objects are imported.

Example of Action Before

- When you delete a field from a table that is filled in the customer database you will get an error if you import the objects. You can use an action of type “Delete Data” to empty this field before the objects are imported.
- If you want to convert some data before you import the objects in the customer database you can add an action of type “Run Codeunit”. When importing the transport it will first import that single codeunit, run it and then import the rest of the objects.



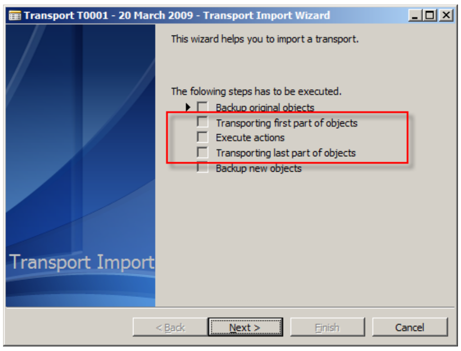
P0003 Convert Data - Project Card

No. P0003
Description Convert Data
Projectflow SIMPLE
Status 60-TRANSPO...
Transport No. T0001
Transported ☑
Search Description CONVERT DATA
Developer HANS
Consultant REIJER
Active Role CONSULTANT
Active User REIJER
Actions Before ☑

Project P0003 Convert Data Before - Action Worksheet

Selection P0003 Project P0002 - Before

Action Type	Object No.	Object Name	Company Name	No. of R...	Filter
Run codeunit	50000	Convert Data		0	Conf



Transport T0001 - 20 March 2009 - Transport Import Wizard

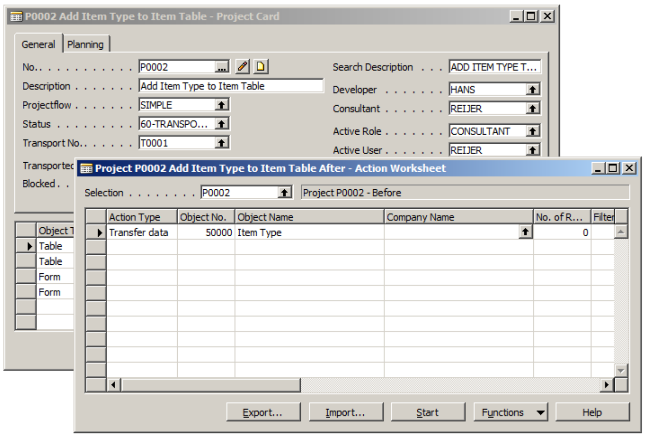
This wizard helps you to import a transport.

The folowing steps has to be executed.

☐ Backup original objects
☐ Transporting first part of objects
☐ Execute actions
☐ Transporting last part of objects
☐ Backup new objects

Example of Action After

- When you have some master data that you want to import in the customer database you can add an action of the type “Transfer Data”.



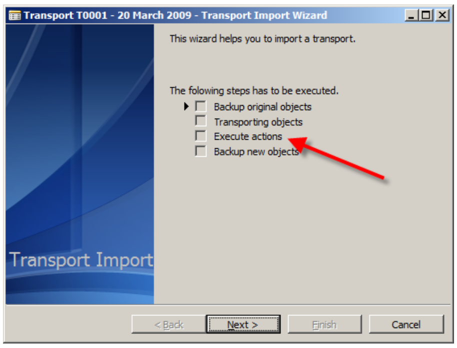
P0002 Add Item Type to Item Table - Project Card

No. P0002
Description Add Item Type to Item Table
Projectflow SIMPLE
Status 60-TRANSPO...
Transport No. T0001
Search Description ADD ITEM TYPE T...
Developer HANS
Consultant REIJER
Active Role CONSULTANT
Active User REIJER

Project P0002 Add Item Type to Item Table After - Action Worksheet

Selection P0002 Project P0002 - Before

Action Type	Object No.	Object Name	Company Name	No. of R...	Filter
Transfer data	50000	Item Type		0	



Transport T0001 - 20 March 2009 - Transport Import Wizard

This wizard helps you to import a transport.

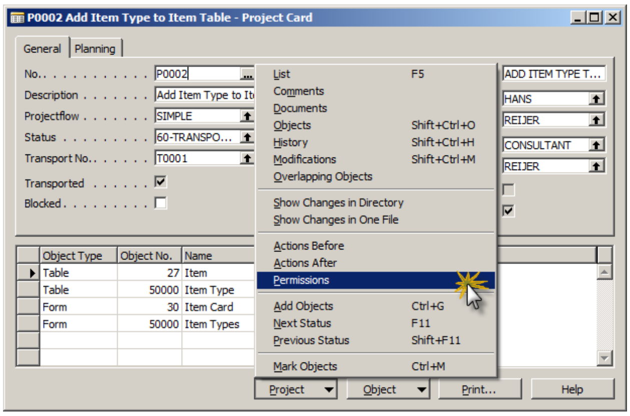
The folowing steps has to be executed.

☐ Backup original objects
☐ Transporting objects
☐ Execute actions
☐ Backup new objects

For more information about actions see chapter *11 - Action Worksheet*.

5.13. Add Permissions to a Project

When you have created some new objects it is required to alter the permissions in the customer database. This can be done by adding permissions to a project. These permissions will be written into the customer database when you transport the project.

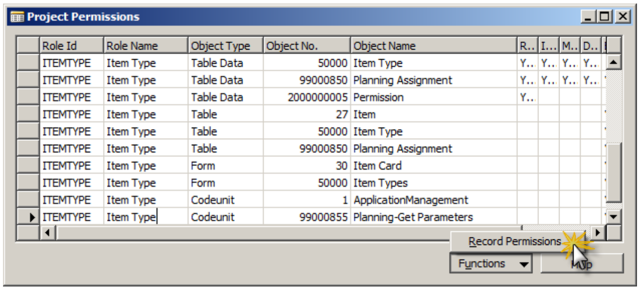

P0002 Add Item Type to Item Table - Project Card

No. P0002
Description . . . . Add Item Type to It
Projectflow . . . . SIMPLE
Status . . . . 60-TRANSPO...
Transport No. T0001
Transported . . . . . . ☑
Blocked . . . . ☐

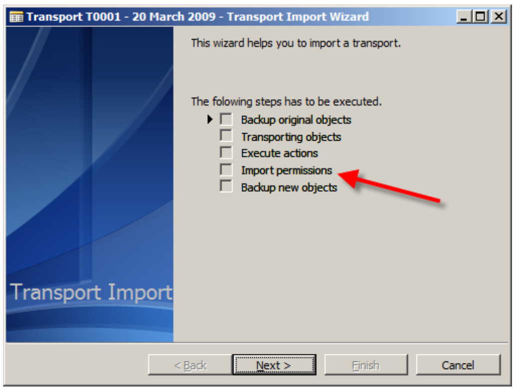
ADD ITEM TYPE T...
HANS
REIJER
CONSULTANT
REIJER
☐
☑

Object Type	Object No.	Name
Table	27	Item
Table	50000	Item Type
Form	30	Item Card
Form	50000	Item Types

The permissions can be recorded with a wizard. When importing the transport file in the customer database there will be an extra step for importing the permissions.


Project Permissions

Role Id	Role Name	Object Type	Object No.	Object Name	R..	I...	M..	D..
ITEMTYPE	Item Type	Table Data	50000	Item Type	Y...	Y...	Y...	Y...
ITEMTYPE	Item Type	Table Data	99000850	Planning Assignment	Y...	Y...	Y...	Y...
ITEMTYPE	Item Type	Table Data	2000000005	Permission	Y...			
ITEMTYPE	Item Type	Table	27	Item				
ITEMTYPE	Item Type	Table	50000	Item Type				
ITEMTYPE	Item Type	Table	99000850	Planning Assignment				
ITEMTYPE	Item Type	Form	30	Item Card				
ITEMTYPE	Item Type	Form	50000	Item Types				
ITEMTYPE	Item Type	Codeunit	1	ApplicationManagement				
ITEMTYPE	Item Type	Codeunit	99000855	Planning-Get Parameters				


Transport T0001 - 20 March 2009 - Transport Import Wizard

This wizard helps you to import a transport.

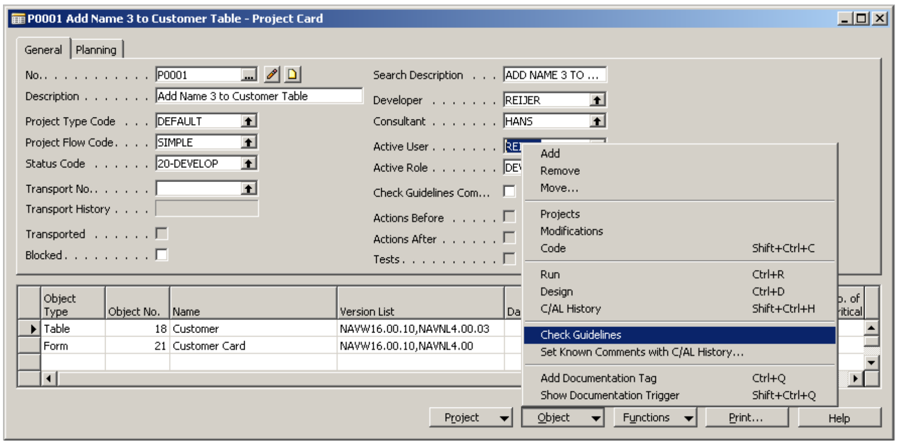
The folowing steps has to be executed.

- ☐ Backup original objects
- ☐ Transporting objects
- ☐ Execute actions
- ☐ Import permissions
- ☐ Backup new objects

For more information see chapter 14 - Record Permission Wizard.

5.14. Check Guidelines

You can check if the Guidelines are met for all the objects in the project.

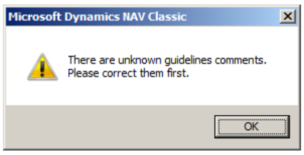

P0001 Add Name 3 to Customer Table - Project Card
General
Planning
Add
Remove
Move...
Projects
Modifications
Code Shift+Ctrl+C
Run Ctrl+R
Design Ctrl+D
C/AL History Shift+Ctrl+H
Check Guidelines
Set Known Comments with C/AL History...
Add Documentation Tag Ctrl+Q
Show Documentation Trigger Shift+Ctrl+Q
Project
Object
Functions
Print...
Help

No.	P0001
Description	Add Name 3 to Customer Table
Project Type Code	DEFAULT
Project Flow Code	SIMPLE
Status Code	20-DEVELOP
Transport No.	
Transport History	
Transported	☐
Blocked	☐
Search Description	ADD NAME 3 TO ...
Developer	REIJER
Consultant	HANS
Active User	RE
Active Role	DEV
Check Guidelines Com...	☐
Actions Before	☐
Actions After	☐
Tests	☐

Object Type	Object No.	Name	Version List
Table	18	Customer	NAVW16.00.10,NAVNL4.00.03
Form	21	Customer Card	NAVW16.00.10,NAVNL4.00

If in the project type the “Check Guidelines at Set Ready Project” is enabled this will be done automatically when the status of the project is set to ready.

If “Check Guidelines Comments” is set on the project card you will not be able to transport the objects until all code in the objects are according to the guidelines or set to known.

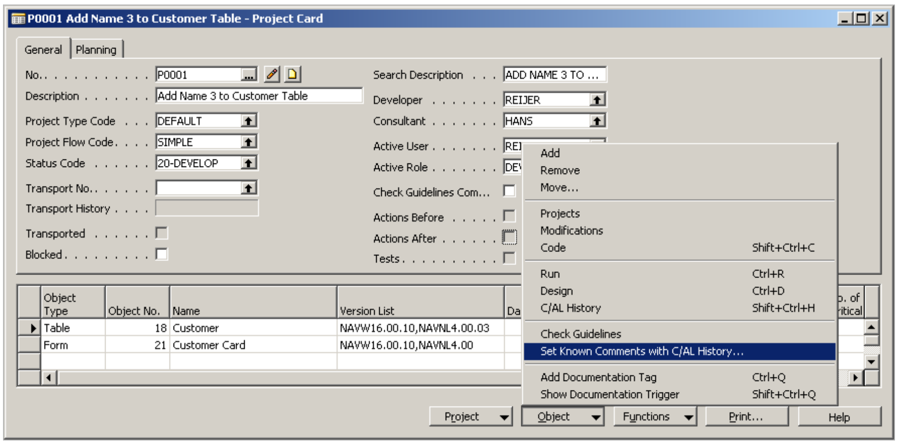


There are unknown guidelines comments.
Please correct them first.

For more information about Check Guidelines see chapter 18 - Check Guidelines.

5.15. Set Known Comments with C/AL History

With this function you can set to known all comments that were already present in a point of time, for objects selected.


P0001 Add Name 3 to Customer Table - Project Card
General
Planning
Add
Remove
Move...
Projects
Modifications
Code Shift+Ctrl+C
Run Ctrl+R
Design Ctrl+D
C/AL History Shift+Ctrl+H
Check Guidelines
Set Known Comments with C/AL History...
Add Documentation Tag Ctrl+Q
Show Documentation Trigger Shift+Ctrl+Q
Project
Object
Functions
Print...
Help

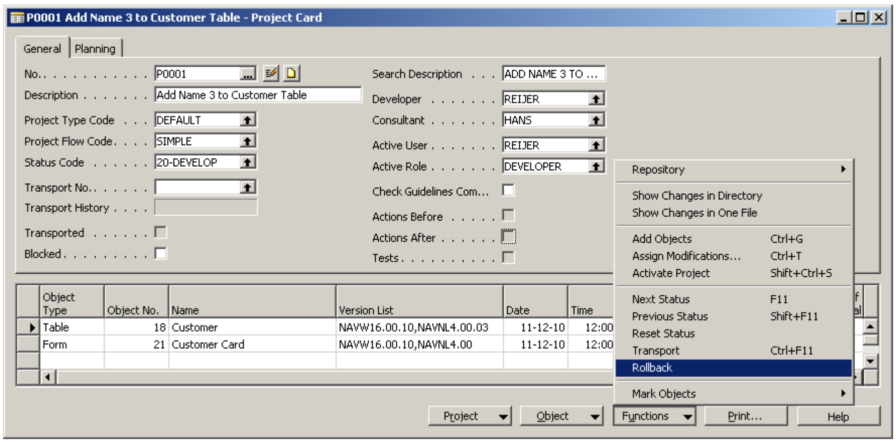
No.	P0001
Description	Add Name 3 to Customer Table
Project Type Code	DEFAULT
Project Flow Code	SIMPLE
Status Code	20-DEVELOP
Transport No.	
Transport History	
Transported	☐
Blocked	☐
Search Description	ADD NAME 3 TO ...
Developer	REIJER
Consultant	HANS
Active User	REI
Active Role	DEV
Check Guidelines Com...	☐
Actions Before	☐
Actions After	☐
Tests	☐

Object Type	Object No.	Name	Version List
Table	18	Customer	NAVW16.00.10,NAVNL4.00.03
Form	21	Customer Card	NAVW16.00.10,NAVNL4.00

For more information about Know Comments see section 18.9 – Known Comment.

5.16. Rollback Objects

To Rollback object changes due to a project, press Functions – Rollback on the project card to open the “Rollback Objects” window.



P0001 Add Name 3 to Customer Table - Project Card

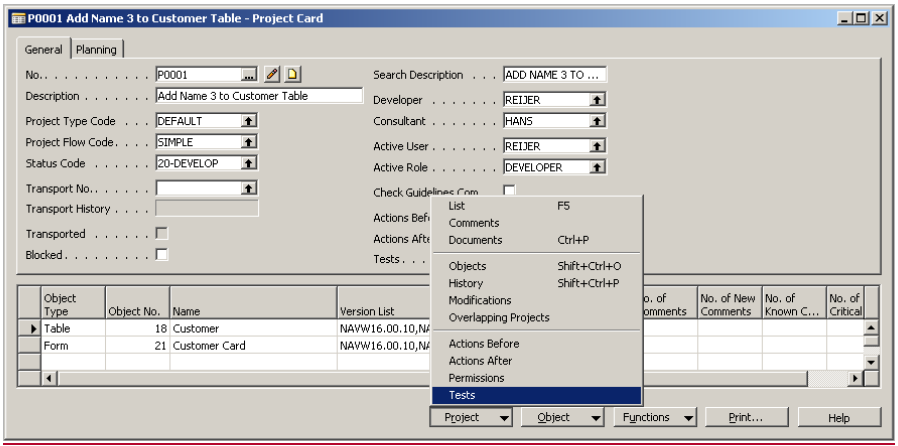
No. P0001
Description . . . . Add Name 3 to Customer Table
Project Type Code . . . DEFAULT
Project Flow Code. . . . SIMPLE
Status Code 20-DEVELOP
Transport No.. . . . . .
Transport History . . . .
Transported . . . . . . ☐
Blocked. . . . . ☐
Search Description . . . ADD NAME 3 TO ...
Developer . . . . REIJER
Consultant . . . . HANS
Active User . . . . REIJER
Active Role . . . . DEVELOPER
Check Guidelines Com... ☐
Actions Before . . . . . ☐
Actions After . . . . . . ☐
Tests. . . . . ☐

Object Type	Object No.	Name	Version List	Date	Time
Table	18	Customer	NAVW16.00.10,NAVNL4.00.03	11-12-10	12:00
Form	21	Customer Card	NAVW16.00.10,NAVNL4.00	11-12-10	12:00

Press Functions – Rollback (or F11) to execute the rollback.
For more information, see section 9.7 – Rollback Objects.

5.17. Test Worksheet

You can add tests to a project for example to check if old functionality still works in the database after transport.



P0001 Add Name 3 to Customer Table - Project Card

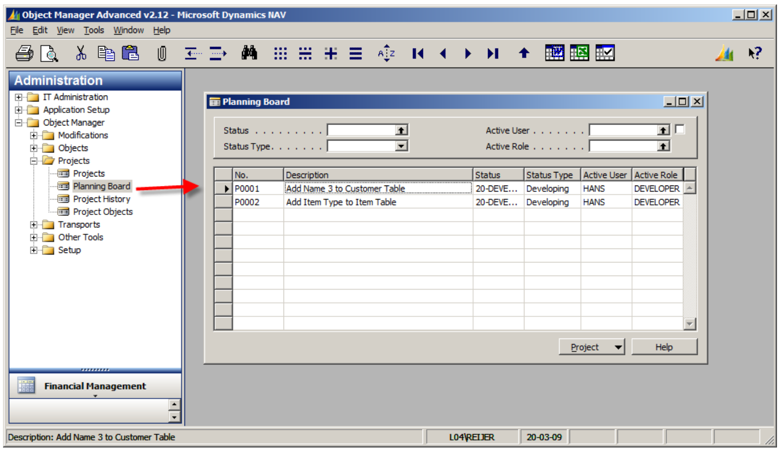
No. P0001
Description . . . . Add Name 3 to Customer Table
Project Type Code . . . DEFAULT
Project Flow Code. . . . SIMPLE
Status Code 20-DEVELOP
Transport No.. . . . . .
Transport History . . . .
Transported . . . . . . ☐
Blocked. . . . . ☐
Search Description . . . ADD NAME 3 TO ...
Developer . . . . REIJER
Consultant . . . . HANS
Active User . . . . REIJER
Active Role . . . . DEVELOPER
Check Guidelines Com ☐
Actions Bef
Actions Afte
Tests. . .

Object Type	Object No.	Name	Version List	o. of omments	No. of New Comments	No. of Known C...	No. of Critical
Table	18	Customer	NAVW16.00.10,NA				
Form	21	Customer Card	NAVW16.00.10,NA				

For more information see chapter 12 - Test Framework.

5.18. Planning Board

With the “Planning Board” you have an overview of the projects and their status.



No.	Description	Status	Status Type	Active User	Active Role
P0001	Add Name 3 to Customer Table	20-DEVE...	Developing	HANS	DEVELOPER
P0002	Add Item Type to Item Table	20-DEVE...	Developing	HANS	DEVELOPER

Description: Add Name 3 to Customer Table

L04\REIJER

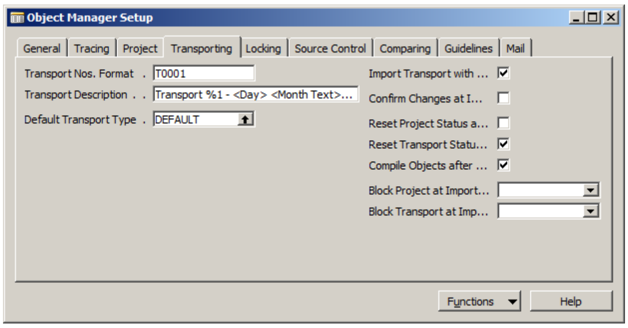
20-03-09

Clicking the checkbox next to the textbox “Active User” on the above right corner shows all the projects where you are the active person.

6. Transport

With a transport you can transfer objects from your development database to the customer database. A transport can also include master data, actions to run such as reports, permissions and documents.

6.1. Setup



Transport Nos. Format . T0001
Transport Description . . Transport %1 - <Day> <Month Text>...
Default Transport Type . DEFAULT
Import Transport with ... ☑
Confirm Changes at I... ☐
Reset Project Status a... ☐
Reset Transport Statu... ☑
Compile Objects after ... ☑
Block Project at Import...
Block Transport at Imp...

- **Transport Nos. Format**

- **Transport Description**
  %1 will be replaced by the "Transport No.".
  You can use date expressions like: <Day> <Month Text> <Year4>.

- **Import Transport with SQL trigger**
  This setting makes it possible to import fields and objects outside the active license.

- **Confirm Changes at Import Transport**
  If your customer has changed any objects since the last transport the object compare sheet will open when you import a transport. For more information see section 6.12 - Confirm Changes at Importing Transport.

- **Reset Project Status at Import Transport**
  All projects will have the default status after you have imported a transport in your test database. For more information see section 6.14 - Reset Project Status at Importing Transport.

- **Compile Objects after Import Transport**
  After you import a transport all objects will be automatically compiled.

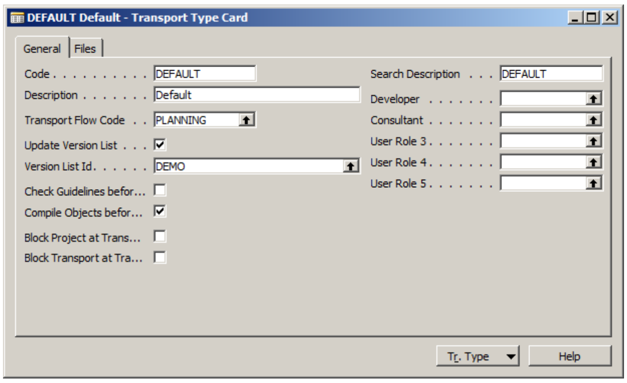
- **Block Project at Import Transport**
  With this setting you can change the blocked status of projects when they are imported with a transport.
  - <Empty>: The status of the project will remain the same as it was when it was transported.

    - No: Projects will be de-blocked when they are imported.
    - Yes: Projects will be blocked when they are imported.

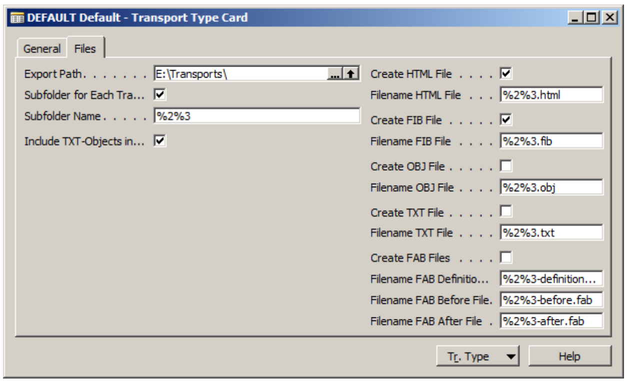
- **Block Transport at Import Transport**
  With this setting you can change the blocked status of a transport when they are imported.
    - <Empty>: The status of the transport will remain the same as it was when it was exported.
    - No: Transport will be de-blocked when it is imported.
    - Yes: Transport will be blocked when it is imported.

6.2. Transport Type

You can differentiate between transports by setting up different types. For example a support issue or hotfix or a release of a functionality.



Code . . . . DEFAULT
Description . . . . Default
Transport Flow Code . . PLANNING
Update Version List . . . ☑
Version List Id. . . . . . DEMO
Check Guidelines befor... ☐
Compile Objects befor... ☑
Block Project at Trans... ☐
Block Transport at Tra... ☐
Search Description . . . DEFAULT
Developer . . . .
Consultant . . . .
User Role 3 . . . .
User Role 4 . . . .
User Role 5 . . . .



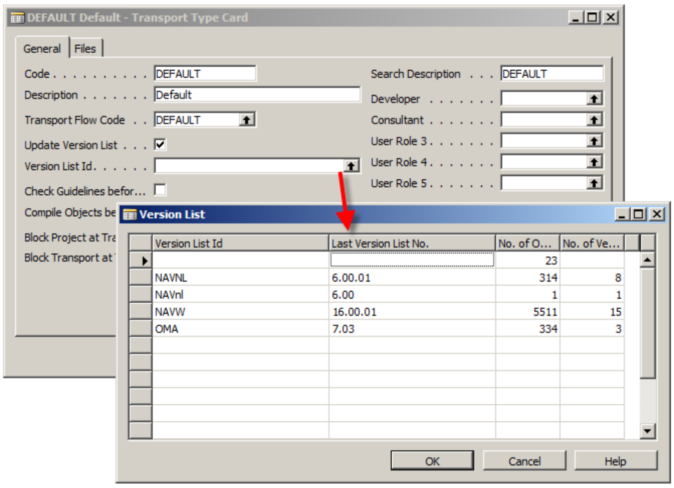
Export Path. . . . . . . E:\Transports\
Subfolder for Each Tra... ☑
Subfolder Name . . . . . %2%3
Include TXT-Objects in... ☑
Create HTML File . . . . ☑
Filename HTML File . . . %2%3.html
Create FIB File . . . . . ☑
Filename FIB File . . . . %2%3.fib
Create OBJ File . . . . . ☐
Filename OBJ File . . . . %2%3.obj
Create TXT File . . . . . ☐
Filename TXT File . . . . %2%3.txt
Create FAB Files . . . . ☐
Filename FAB Definitio... %2%3-definition...
Filename FAB Before File. %2%3-before.fab
Filename FAB After File . %2%3-after.fab

- **Description**
  Description of the transport type.

- **Transport Flow Code**
  For more information see section 6.5 - Transport Flows.

- **Update Version List**
  This is the default setting that is copied into new transports. When this setting is enabled the version list of the objects in the transport will be updated when the transport will be transported.

- **Version List Id**
  Select the default “Version List Id”. This “Version List Id” will be copied into every new transport. If you use the lookup you will see all used Version List Id’s in the current database. It is also possible to fill in another id.



DEFAULT Default - Transport Type Card

Code . . . . DEFAULT
Description . . . . Default
Transport Flow Code . . DEFAULT
Update Version List . . . ☑
Version List Id. . . . . .
Check Guidelines befor… ☐
Compile Objects be
Block Project at Tra
Block Transport at
Search Description . . . DEFAULT
Developer . . . .
Consultant . . . .
User Role 3 . . . .
User Role 4 . . . .
User Role 5 . . . .

Version List

Version List Id	Last Version List No.	No. of O…	No. of Ve…
		23	
NAVNL	6.00.01	314	8
NAVnl	6.00	1	1
NAVW	16.00.01	5511	15
OMA	7.03	334	3

- **Check Guidelines before Transport**
  When this setting is enabled the transport can only be done if there are no unknown comments. For more information see chapter 18 - Check Guidelines.

- **Compile Objects before Transport**
  Objects are compiled when you are transporting. You will get an error if there is an error in an object.

- **Block Project at Transport**
  All projects will be blocked if you transport. Transported projects cannot be de-blocked if this option is enabled.

- **Block Transport at Transport**
  The transport will be blocked at transport. A transported transport cannot be de-blocked if this option is enabled.

- **User Roles**
  You can define the default users here.

- **Export Path**
  This folder will be used to save the transport files.

- **Subfolder for Each Transport**
  The Object Manager creates a new folder for each transport.

  %1 will be replaced by the “Transport No.”.
  %2 will be replaced by the “Version List Id” of the transport.
  %3 will be replaced by the “Version List No.” of the transport.

  It is also possible to use expressions like: <Day> <Month Text> <Year4>.

- **Transport Files**
  You can choose which files are saved to your disk when you transport a transport.
  - HTML: a report with the transport information.

- FIB: contains all transport and project data, actions, document and the objects of the transport.
- OBJ: contains the objects. (See section 7.2 - Import and Export Files).
- TXT: contains the objects in text format.
- FAB: contains all the data of the projects and transport.
- FAB Before: contains the actions that has to be executed before the objects are imported.
- FAB After: contains the actions that has to be executed after the objects are imported.

NOTE: You only need the FIB file when importing a transport in your customer database.

6.3. Comment Groups

Comment Groups are similar to Comment Groups in projects. For more information see section 5.3 - Comment Groups.

6.4. Documents

Documents in transports are similar to Documents in projects. For more information see section 5.4 - Documents.

6.5. Transport Flow

Transport Flows are similar to Flows in projects. For more information see section 5.5 - Project Flow.

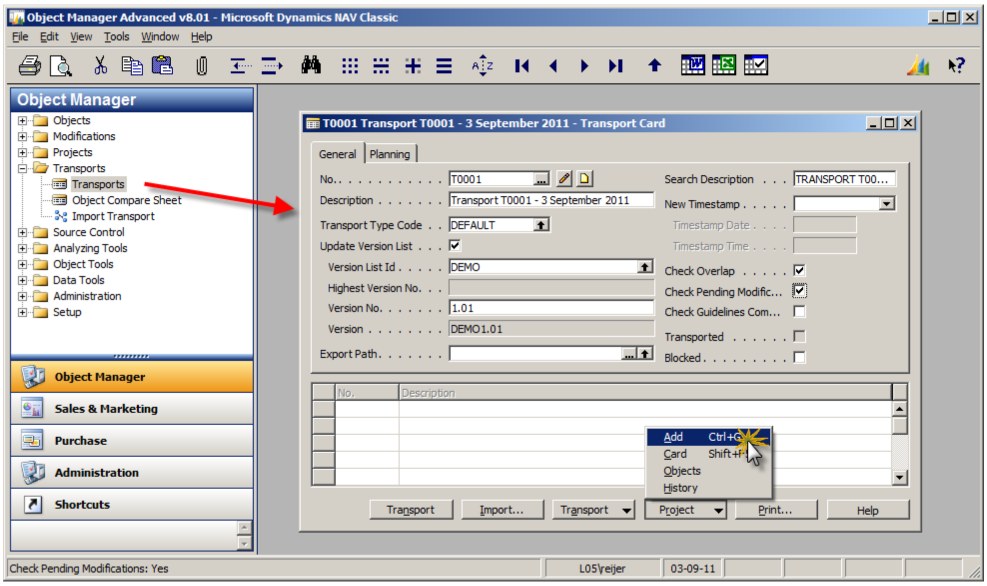
By default a transport gets status type Ready. With this status type you can only add projects that are ready.

If a transport has status type <empty> or Developing you can add all projects. This can be used if you want to plan a future transport and add projects to it that are not yet ready.

6.6. Export Transport

Menu Transports > Make a new Transport Card (F3).

Add projects to the transport.



No. T0001

Description Transport T0001 - 3 September 2011

Transport Type Code DEFAULT

Update Version List ☑

Version List Id DEMO

Highest Version No.

Version No. 1.01

Version DEMO1.01

Export Path

Search Description TRANSPORT T00...

New Timestamp

Timestamp Date

Timestamp Time

Check Overlap ☑

Check Pending Modific... ☑

Check Guidelines Com... ☐

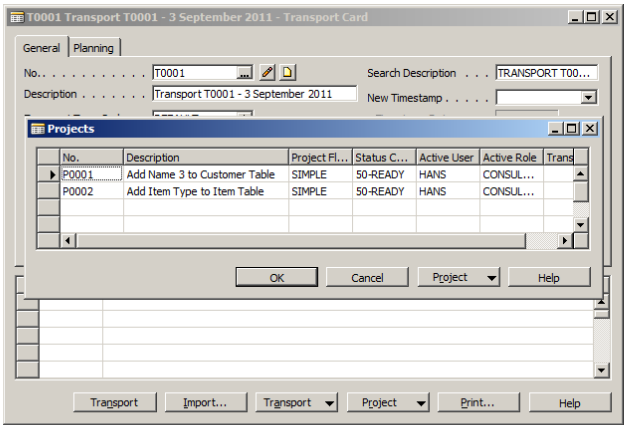
Transported ☐

Blocked ☐

Check Pending Modifications: Yes | L05\reijer | 03-09-11

By default only projects with status ready are visible in this overview. This prevents you from transporting projects that are not completely finished yet.

If you work with transport flows it can be possible that your transport has status type <empty> or developing. With this two status types it is possible to add all projects.



No. T0001

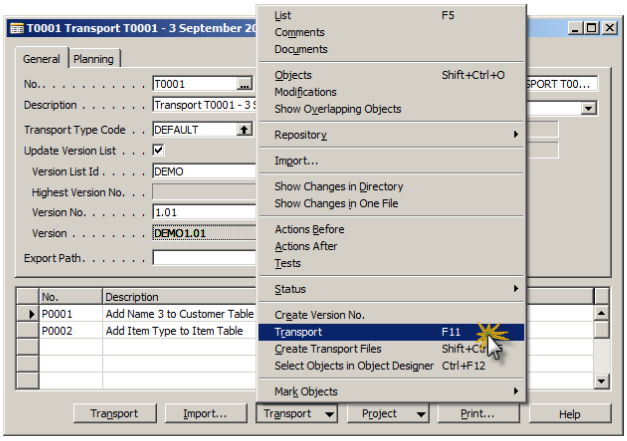
Description Transport T0001 - 3 September 2011

Search Description TRANSPORT T00...

New Timestamp

No.	Description	Project Fl...	Status C...	Active User	Active Role	Trans
P0001	Add Name 3 to Customer Table	SIMPLE	50-READY	HANS	CONSUL...	
P0002	Add Item Type to Item Table	SIMPLE	50-READY	HANS	CONSUL...	

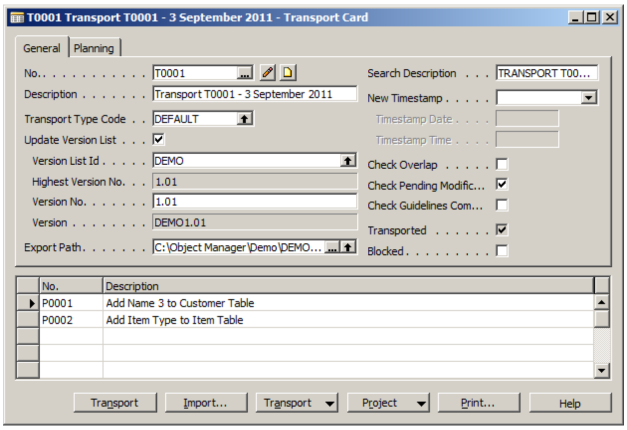
Transport (F11) > The Object Manager creates a new folder with the transport files. The modify flag of the objects will be removed.


T0001 Transport T0001 - 3 September 20

No. T0001
Description Transport T0001 - 3 S
Transport Type Code DEFAULT
Update Version List ☑
Version List Id DEMO
Highest Version No.
Version No. 1.01
Version DEMO1.01
Export Path

No.	Description
P0001	Add Name 3 to Customer Table
P0002	Add Item Type to Item Table

By clicking the assist edit button of textbox “Export Path” you can surf to the folder location.

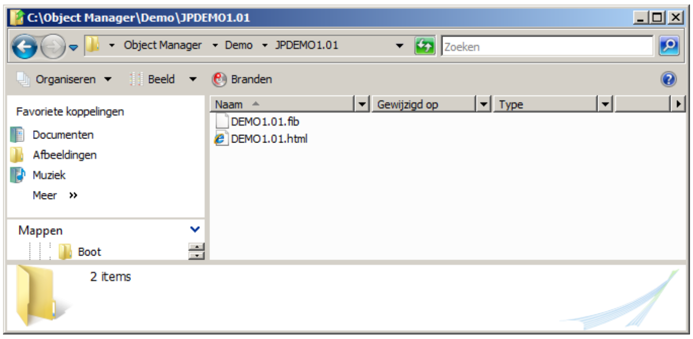

T0001 Transport T0001 - 3 September 2011 - Transport Card

No. T0001
Description Transport T0001 - 3 September 2011
Transport Type Code DEFAULT
Update Version List ☑
Version List Id DEMO
Highest Version No. 1.01
Version No. 1.01
Version DEMO1.01
Export Path C:\Object Manager\Demo\DEMO...

Search Description TRANSPORT T00...
New Timestamp
Timestamp Date
Timestamp Time
Check Overlap ☐
Check Pending Modific... ☑
Check Guidelines Com... ☐
Transported ☑
Blocked ☐

No.	Description
P0001	Add Name 3 to Customer Table
P0002	Add Item Type to Item Table

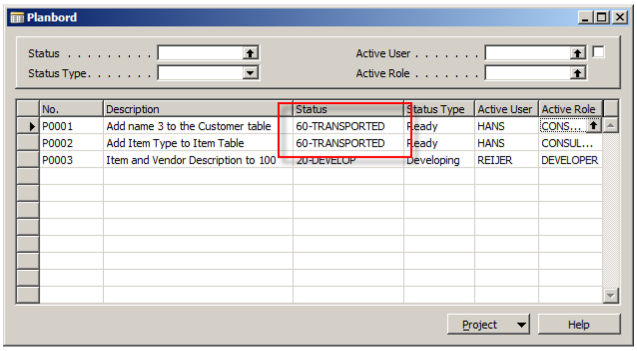
A FIB file is created in the folder as well as the transport report in HTML format and the attached documents of the projects.


C:\Object Manager\Demo\JPDEMO1.01

Naam	Gewijzigd op	Type
DEMO1.01.fib		
DEMO1.01.html		

2 items

In the “Planning Board” you can see that the status of the projects are now set to Transported.



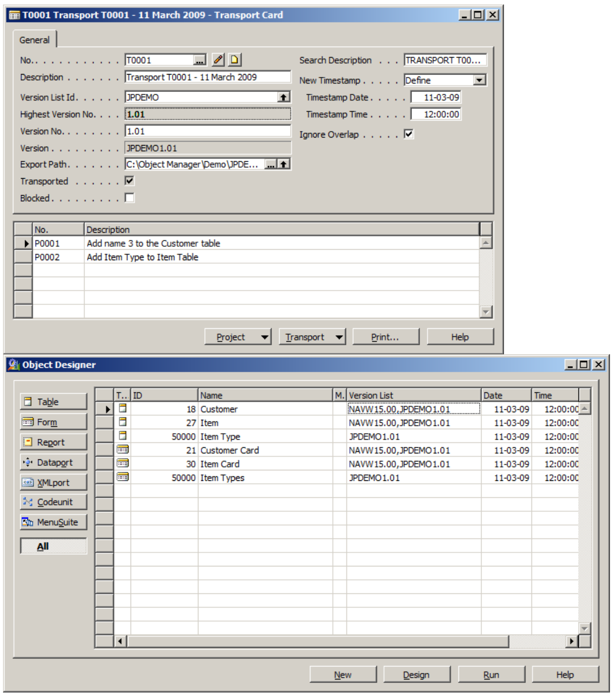
No.	Description	Status	Status Type	Active User	Active Role
P0001	Add name 3 to the Customer table	60-TRANSPORTED	Ready	HANS	CONS...
P0002	Add Item Type to Item Table	60-TRANSPORTED	Ready	HANS	CONSUL...
P0003	Item and Vendor Description to 100	20-DEVELOP	Developing	REIJER	DEVELOPER

6.7. Timestamp

You can also modify the timestamp to the objects when you transport.

- **Timestamp Options**
    - <Empty>: The timestamp of the objects will be the date and time of the last modification.
    - Moment of Transport: Timestamp will set to the time of transport.
    - Define: You can define your own timestamp in fields Timestamp Date and Time.

NOTE: If you differ too much from the real transport date “Version Control” can act strangely. (See chapter 9 - Version and Source Control**)**



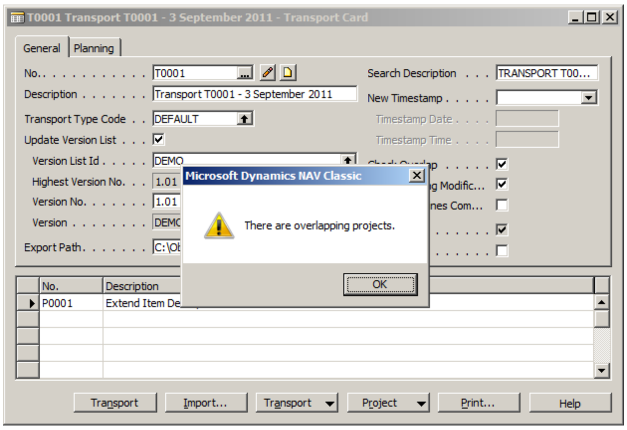
No. T0001
Description . . . . Transport T0001 - 11 March 2009
Version List Id. . . . . . JPDEMO
Highest Version No. . . . 1.01
Version No. 1.01
Version JPDEMO1.01
Export Path. . . . . . . C:\Object Manager\Demo\JPDE...
Transported . . . . . . ☑
Blocked . . . . ☐
Search Description . . . TRANSPORT T00...
New Timestamp . . . . . Define
Timestamp Date 11-03-09
Timestamp Time 12:00:00
Ignore Overlap . . . . . ☑

No.	Description
P0001	Add name 3 to the Customer table
P0002	Add Item Type to Item Table

T..	ID	Name	M.	Version List	Date	Time
	18	Customer		NAVW15.00,JPDEMO1.01	11-03-09	12:00:0
	27	Item		NAVW15.00,JPDEMO1.01	11-03-09	12:00:0
	50000	Item Type		JPDEMO1.01	11-03-09	12:00:0
	21	Customer Card		NAVW15.00,JPDEMO1.01	11-03-09	12:00:0
	30	Item Card		NAVW15.00,JPDEMO1.01	11-03-09	12:00:0
	50000	Item Types		JPDEMO1.01	11-03-09	12:00:0

6.8. *Overlapping Objects*

When transporting projects you sometimes have the problem of overlapping objects. For example: you used table Item as well in project 1 as in project 2 and you want to transport only project 1. The Object Manager checks if the transport has overlapping objects.



T0001 Transport T0001 - 3 September 2011 - Transport Card

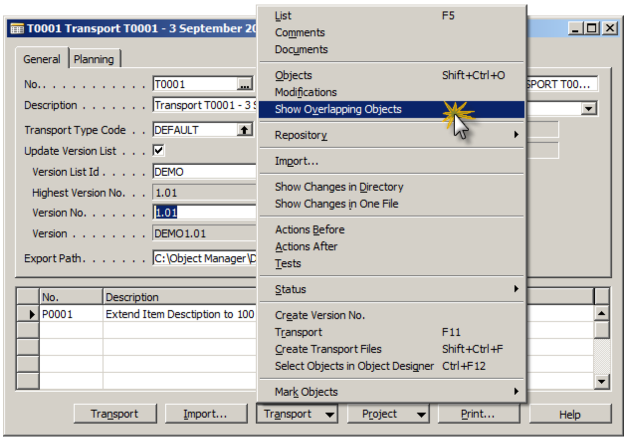
No. T0001
Description Transport T0001 - 3 September 2011
Transport Type Code DEFAULT
Update Version List ☑
Version List Id DEMO
Highest Version No. 1.01
Version No. 1.01
Search Description TRANSPORT T00...

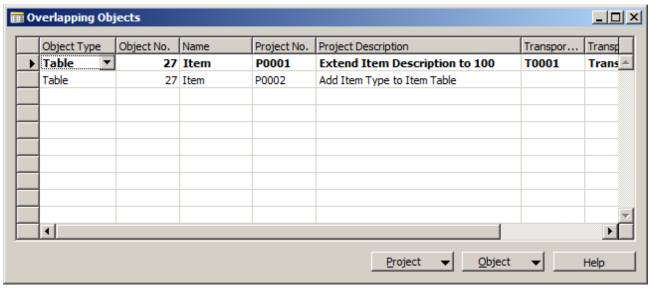
Microsoft Dynamics NAV Classic

There are overlapping projects.

No.	Description
P0001	Extend Item De

You can check where these objects are used with the function “Show Overlapping Objects”.





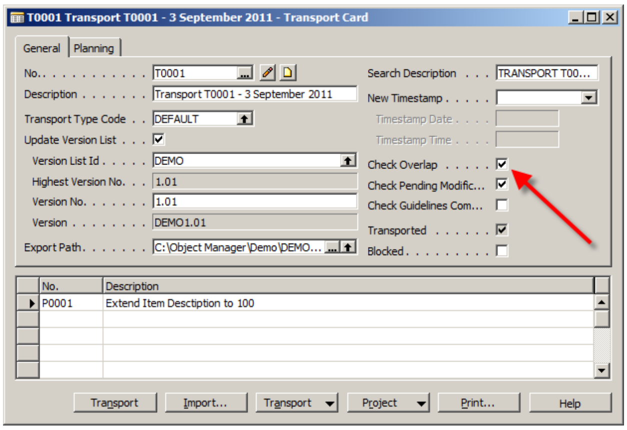
Overlapping Objects

Object Type	Object No.	Name	Project No.	Project Description	Transpor...	Transp
Table	**27**	**Item**	**P0001**	**Extend Item Description to 100**	**T0001**	**Trans**
Table	27	Item	P0002	Add Item Type to Item Table		

The bold lines in this form are in the transport.
Now you have three options:

1. Remove the modifications manually made in project 2 and delete the object from project 2

2. Finish and include project 2 to the transport
3. Disable “Check Overlap” on the “Transport Card” if you know for sure that it would not give any complications in the customer database

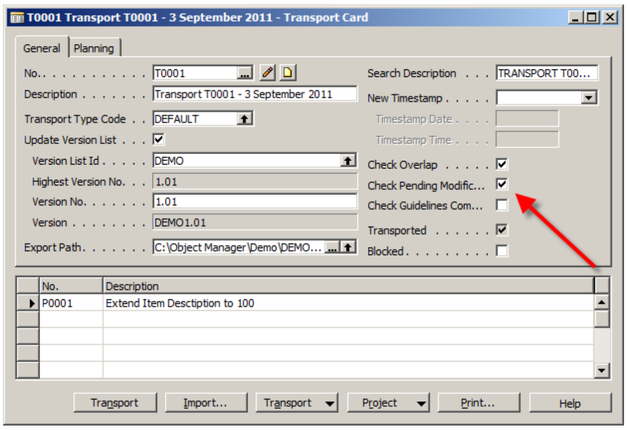

T0001 Transport T0001 - 3 September 2011 - Transport Card

No. T0001
Description Transport T0001 - 3 September 2011
Transport Type Code DEFAULT
Update Version List ☑
Version List Id DEMO
Highest Version No. 1.01
Version No. 1.01
Version DEMO1.01
Export Path C:\Object Manager\Demo\DEMO...
Search Description TRANSPORT T00...
New Timestamp
Timestamp Date
Timestamp Time
Check Overlap ☑
Check Pending Modific... ☑
Check Guidelines Com... ☐
Transported ☑
Blocked ☐

No.	Description
P0001	Extend Item Desctiption to 100

6.9. Pending Modifications

When you transport the objects, the Object Manager checks if there are modifications present that are not yet assigned to a project. This is to be sure that you did not forget any modifications. If you are sure that the pending modifications do not give any complications in your customer database you can disable the “Check Pending Modifications” option on the Transport card.


T0001 Transport T0001 - 3 September 2011 - Transport Card

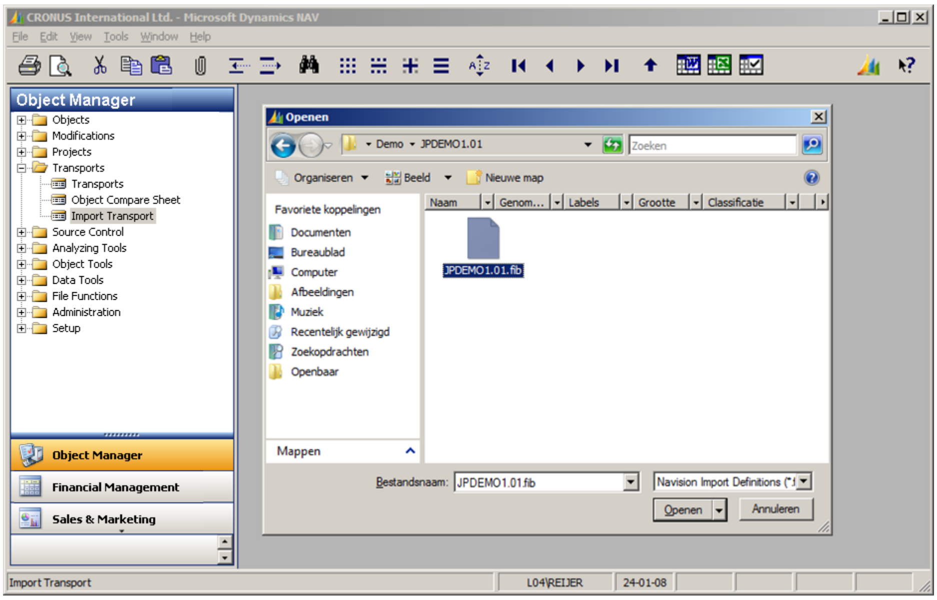
No. T0001
Description Transport T0001 - 3 September 2011
Transport Type Code DEFAULT
Update Version List ☑
Version List Id DEMO
Highest Version No. 1.01
Version No. 1.01
Version DEMO1.01
Export Path C:\Object Manager\Demo\DEMO...
Search Description TRANSPORT T00...
New Timestamp
Timestamp Date
Timestamp Time
Check Overlap ☑
Check Pending Modific... ☑
Check Guidelines Com... ☐
Transported ☑
Blocked ☐

No.	Description
P0001	Extend Item Desctiption to 100

6.10. Import Transport

The transport (FIB file) can be imported in the customer database.

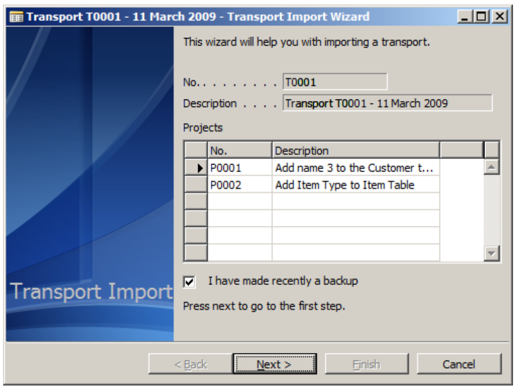
Select Transports > Import Transport and select the FIB file you want to import.



Bestandsnaam: JPDEMO1.01.fib

Navision Import Definitions (*.f

The “Transport Import Wizard” starts with all the steps that need to be executed.



This wizard will help you with importing a transport.

No. T0001

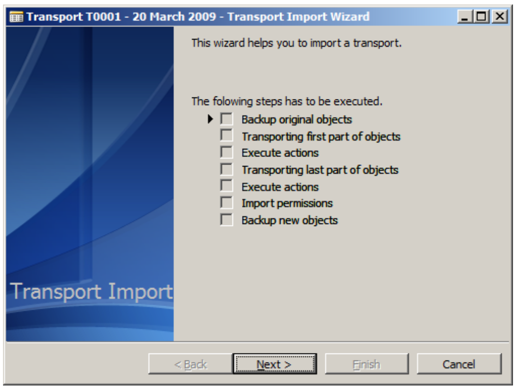
Description . . . . Transport T0001 - 11 March 2009

Projects

No.	Description
P0001	Add name 3 to the Customer t...
P0002	Add Item Type to Item Table

☑ I have made recently a backup

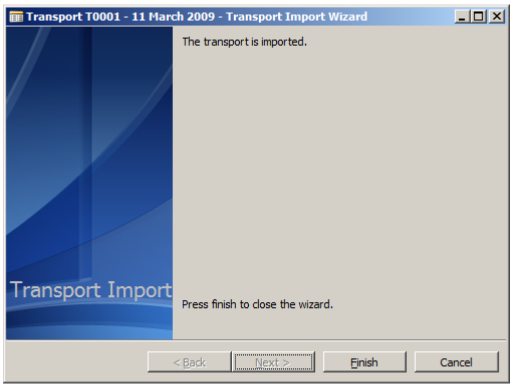
Press next to go to the first step.



This wizard helps you to import a transport.

The folowing steps has to be executed.

- ☐ Backup original objects
- ☐ Transporting first part of objects
- ☐ Execute actions
- ☐ Transporting last part of objects
- ☐ Execute actions
- ☐ Import permissions
- ☐ Backup new objects



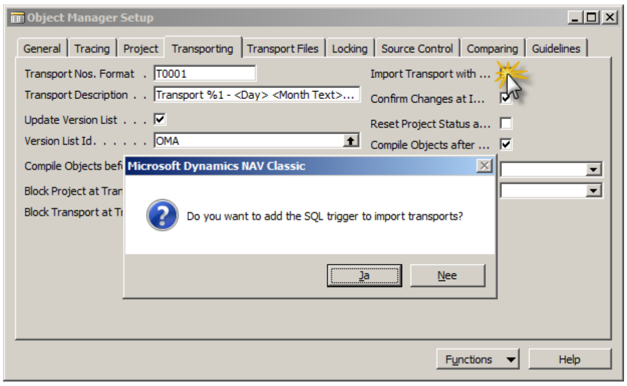
The transport is imported.

Press finish to close the wizard.

NOTE: When importing a FIB file into a database, that does not yet contain the projects and transport included in the FIB file, these projects and transport will be created in this database.

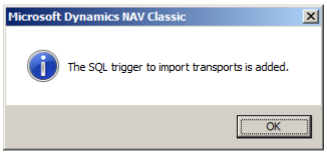
6.11. *Import Transport with SQL Trigger*

If you import a transport with a customer license you sometimes get a license error that a field or object cannot be created. If you enable the option “Import Transport with SQL Trigger” all objects will be imported through a object number available in the customer license. Therefore fields and objects outside the customer license can be imported.



Transport Nos. Format . T0001
Transport Description . . Transport %1 - <Day> <Month Text>...
Update Version List . . . ☑
Version List Id. . . . . . . OMA
Compile Objects befi
Block Project at Trar
Block Transport at Ti
Import Transport with ...
Confirm Changes at I... ☑
Reset Project Status a... ☐
Compile Objects after ... ☑

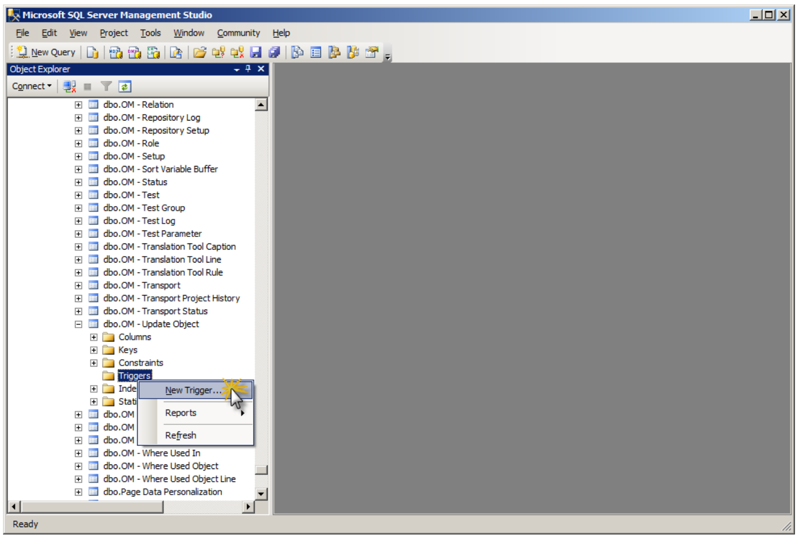
Do you want to add the SQL trigger to import transports?



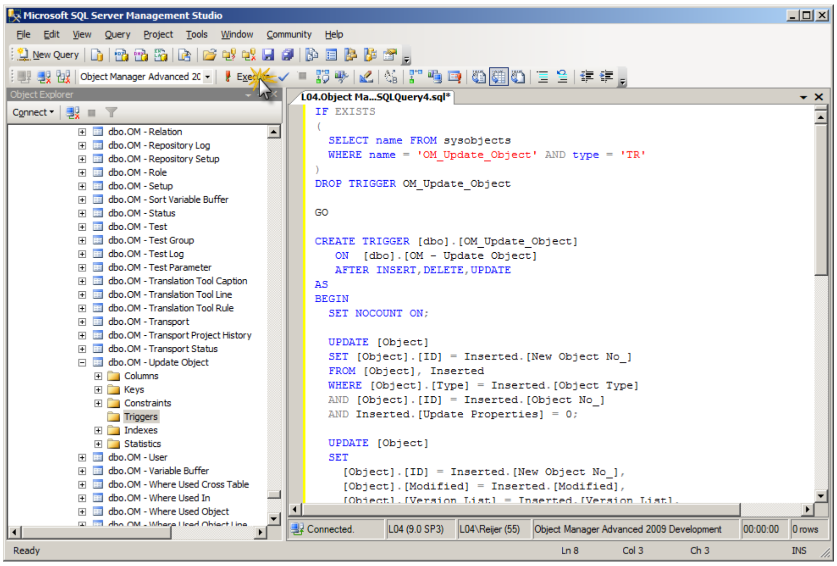
The SQL trigger to import transports is added.

If you enable the option the Object Manager will install a trigger on the table “OM – Update Object”. If the SQL database cannot be found or the active user is no DBO the trigger has to be added manual with e.g. SQL Server Management Studio.

Go to the OM – Update Object table and choose the option to create a new trigger.

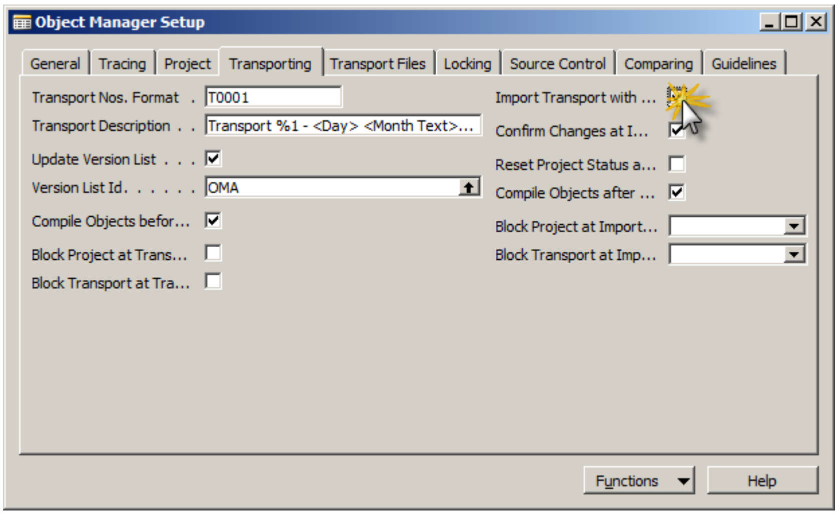


Then paste the SQL statement (see appendix F - Update Object Table SQL Trigger) in the Query window and press execute.


Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Community Help
New Query
Object Manager Advanced 20
Execute
Object Explorer
Connect
dbo.OM - Relation
dbo.OM - Repository Log
dbo.OM - Repository Setup
dbo.OM - Role
dbo.OM - Setup
dbo.OM - Sort Variable Buffer
dbo.OM - Status
dbo.OM - Test
dbo.OM - Test Group
dbo.OM - Test Log
dbo.OM - Test Parameter
dbo.OM - Translation Tool Caption
dbo.OM - Translation Tool Line
dbo.OM - Translation Tool Rule
dbo.OM - Transport
dbo.OM - Transport Project History
dbo.OM - Transport Status
dbo.OM - Update Object
Columns
Keys
Constraints
Triggers
Indexes
Statistics
dbo.OM - User
dbo.OM - Variable Buffer
dbo.OM - Where Used Cross Table
dbo.OM - Where Used In
dbo.OM - Where Used Object
L04.Object Ma...SQLQuery4.sql*
Connected.
L04 (9.0 SP3)
L04\Reijer (55)
Object Manager Advanced 2009 Development
00:00:00
0 rows
Ready
Ln 8
Col 3
Ch 3
INS

```sql
IF EXISTS
(
  SELECT name FROM sysobjects
  WHERE name = 'OM_Update_Object' AND type = 'TR'
)
DROP TRIGGER OM_Update_Object

GO

CREATE TRIGGER [dbo].[OM_Update_Object]
   ON  [dbo].[OM - Update Object]
   AFTER INSERT,DELETE,UPDATE
AS
BEGIN
  SET NOCOUNT ON;

  UPDATE [Object]
  SET [Object].[ID] = Inserted.[New Object No_]
  FROM [Object], Inserted
  WHERE [Object].[Type] = Inserted.[Object Type]
  AND [Object].[ID] = Inserted.[Object No_]
  AND Inserted.[Update Properties] = 0;

  UPDATE [Object]
  SET
    [Object].[ID] = Inserted.[New Object No_],
    [Object].[Modified] = Inserted.[Modified],
    [Object].[Version List] = Inserted.[Version List],
```

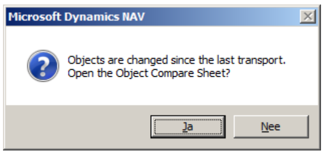
Now it is possible to enable the option “Import Transport with SQL trigger”.



Transport Nos. Format . T0001
Transport Description . . Transport %1 - <Day> <Month Text>...
Update Version List . . . ☑
Version List Id. . . . . . OMA
Compile Objects befor... ☑
Block Project at Trans... ☐
Block Transport at Tra... ☐
Import Transport with ... ☑
Confirm Changes at I... ☑
Reset Project Status a... ☐
Compile Objects after ... ☑
Block Project at Import...
Block Transport at Imp...

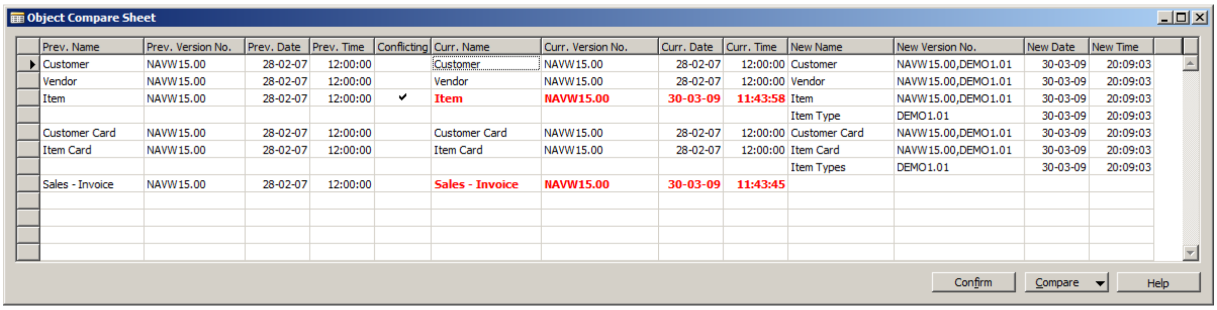
6.12. Confirm Changes at Importing Transport

When your customer has changed one or more objects in the database you get a warning when you import a transport.



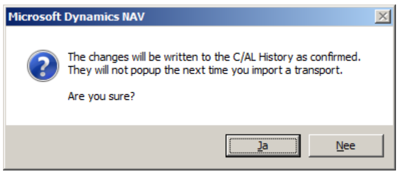
Objects are changed since the last transport.
Open the Object Compare Sheet?

Opening the “Object Compare Sheet” gives you an overview of the changes the customer has made since the previous transport.



Prev. Name	Prev. Version No.	Prev. Date	Prev. Time	Conflicting	Curr. Name	Curr. Version No.	Curr. Date	Curr. Time	New Name	New Version No.	New Date	New Time
Customer	NAVW15.00	28-02-07	12:00:00		Customer	NAVW15.00	28-02-07	12:00:00	Customer	NAVW15.00,DEMO1.01	30-03-09	20:09:03
Vendor	NAVW15.00	28-02-07	12:00:00		Vendor	NAVW15.00	28-02-07	12:00:00	Vendor	NAVW15.00,DEMO1.01	30-03-09	20:09:03
Item	NAVW15.00	28-02-07	12:00:00	✓	**Item**	**NAVW15.00**	**30-03-09**	**11:43:58**	Item	NAVW15.00,DEMO1.01	30-03-09	20:09:03
									Item Type	DEMO1.01	30-03-09	20:09:03
Customer Card	NAVW15.00	28-02-07	12:00:00		Customer Card	NAVW15.00	28-02-07	12:00:00	Customer Card	NAVW15.00,DEMO1.01	30-03-09	20:09:03
Item Card	NAVW15.00	28-02-07	12:00:00		Item Card	NAVW15.00	28-02-07	12:00:00	Item Card	NAVW15.00,DEMO1.01	30-03-09	20:09:03
									Item Types	DEMO1.01	30-03-09	20:09:03
Sales - Invoice	NAVW15.00	28-02-07	12:00:00		**Sales - Invoice**	**NAVW15.00**	**30-03-09**	**11:43:45**				

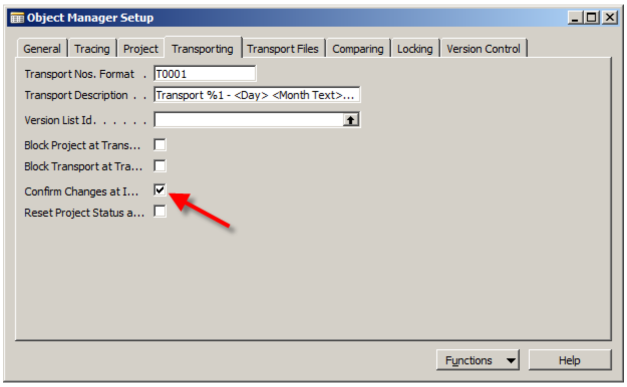
In this example you see that the customer has changed the Item table and the report “Sales - Invoice”. It is possible to analyze the changes with your compare tool or confirm the changes with the confirm button.



The changes will be written to the C/AL History as confirmed.
They will not popup the next time you import a transport.

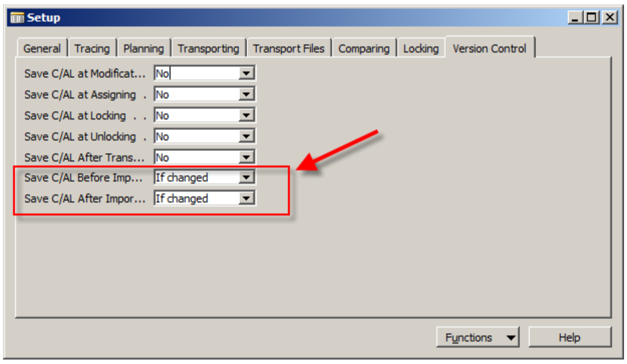
Are you sure?

The option “Confirm Changes at Importing Transport” will always open the “Object Compare Sheet” when importing a transport if objects are changed. All modifications have to be confirmed before you can import a transport.



Transport Nos. Format . T0001
Transport Description . . Transport %1 - <Day> <Month Text>...
Version List Id. . . . . .
Block Project at Trans... ☐
Block Transport at Tra... ☐
Confirm Changes at I... ☑
Reset Project Status a... ☐

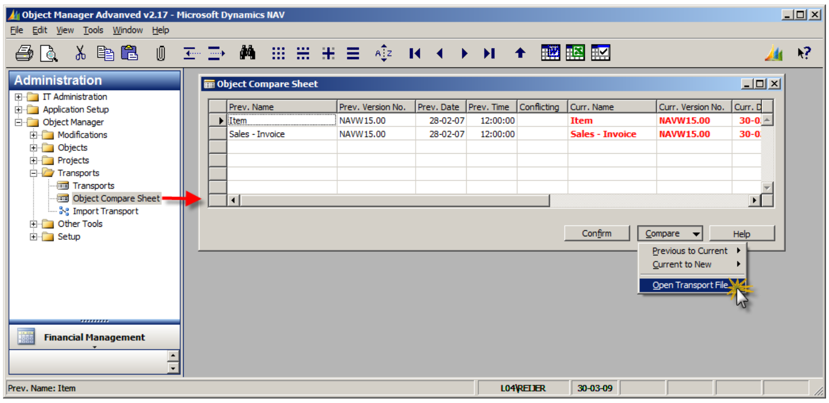
NOTE: Set options “Save C/AL Before Import” and “Save C/AL After Import” on “If changed” or Yes.


Setup
General | Tracing | Planning | Transporting | Transport Files | Comparing | Locking | Version Control
Functions
Help

Save C/AL at Modificat…	No
Save C/AL at Assigning	No
Save C/AL at Locking	No
Save C/AL at Unlocking	No
Save C/AL After Trans…	No
Save C/AL Before Imp…	If changed
Save C/AL After Impor…	If changed

6.13. Object Compare Sheet

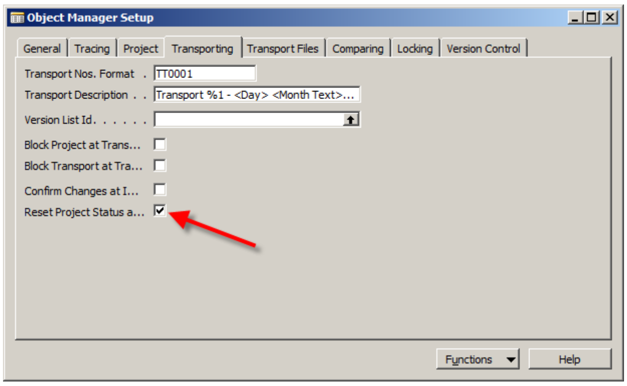
To see what your customer has changed since the last transport you can open the “Object Compare Sheet”. Use the “Open Transport File” option to compare the changes against a transport file. See section 6.12 - Confirm Changes at Importing Transport for more information.



Prev. Name	Prev. Version No.	Prev. Date	Prev. Time	Conflicting	Curr. Name	Curr. Version No.	Curr. D
Item	NAVW15.00	28-02-07	12:00:00		Item	NAVW15.00	30-0
Sales - Invoice	NAVW15.00	28-02-07	12:00:00		Sales - Invoice	NAVW15.00	30-0

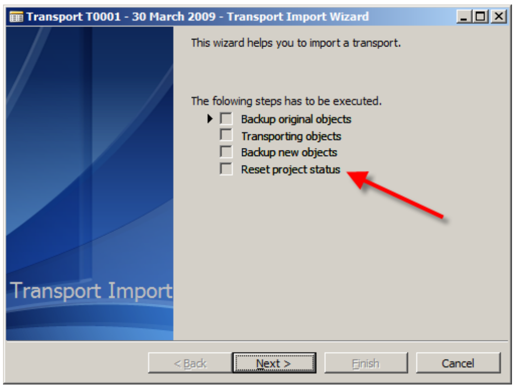
6.14. Reset Project Status at Importing Transport

When you are developing in an environment with a development, test and live database and you want to transport your projects from the test database to the live database you can use the “Reset Project Status at Importing Transport” option in the setup.



Transport Nos. Format . TT0001
Transport Description . . Transport %1 - <Day> <Month Text>...
Version List Id. . . . . .
Block Project at Trans... ☐
Block Transport at Tra... ☐
Confirm Changes at I... ☐
Reset Project Status a... ☑

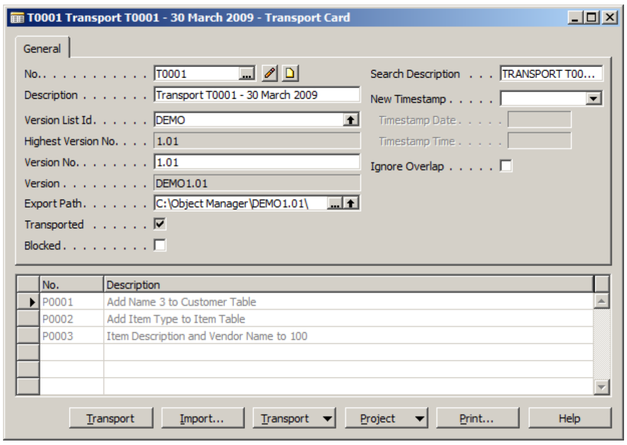
When you import a transport you will see an extra option in the transport import wizard.



This wizard helps you to import a transport.

The folowing steps has to be executed.
- ☐ Backup original objects
- ☐ Transporting objects
- ☐ Backup new objects
- ☐ Reset project status

All the projects that you import with a transport will have the default status of the project flow and the projects will be deleted from the transport. You can recognize this by the grey color in the transport.



No.. . . . . T0001

Description . . . . Transport T0001 - 30 March 2009

Version List Id. . . . . . DEMO

Highest Version No. . . . 1.01

Version No. 1.01

Version DEMO1.01

Export Path. . . . . . . C:\Object Manager\DEMO1.01\

Transported . . . . . . ☑

Blocked. . . . . ☐

Search Description . . . TRANSPORT T00...

New Timestamp . . . . .

Timestamp Date . . . . .

Timestamp Time . . . . .

Ignore Overlap . . . . . ☐

No.	Description
P0001	Add Name 3 to Customer Table
P0002	Add Item Type to Item Table
P0003	Item Description and Vendor Name to 100

Because the field “Transport No.” is cleared on the project card it is possible to include them in another transport.

NOTE: Use another “Transport Nos. Format” in your test database then in your development database.

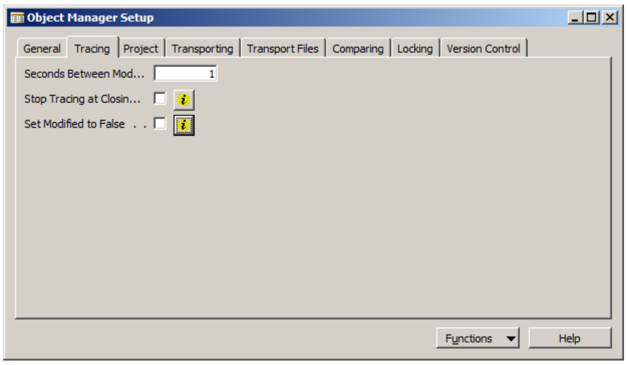
6.15. Rollback Objects

To Rollback object changes due to a transport, press Transport – Rollback to open the “Rollback Objects” window, and then press Functions – Rollback (or F11) to execute the rollback.

For more information about Check Guidelines see section 9.7 – Rollback Objects.

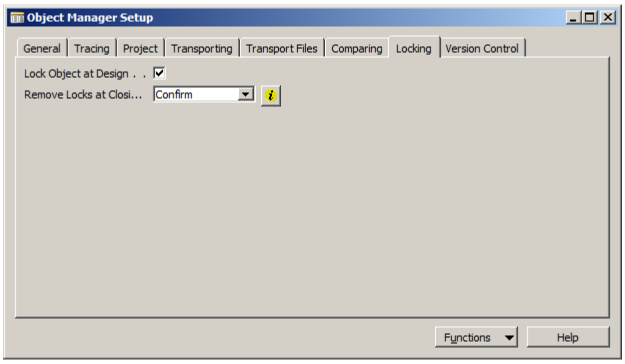
7. Object Explorer

7.1. Setup



Seconds Between Mod... 1
Stop Tracing at Closin... ☐
Set Modified to False . . ☐

- **Stop Tracing at Closing 3.7 Menu and Object Explorer**
  When enabling this setting tracing of modifications will be stopped if you close the Object Explorer.



Lock Object at Design . . ☑
Remove Locks at Closi... Confirm

- **Lock Objects at Design**
  When enabling this setting you will lock each object that you design with the Object Designer. Recommended in an environment with more developers.

- **Remove Locks at Closing 3.7 Menu and Object Explorer**
  When enabling this setting your locks will be removed if you close the “Object Explorer”.

7.2. Auto Open Object Explorer at F12

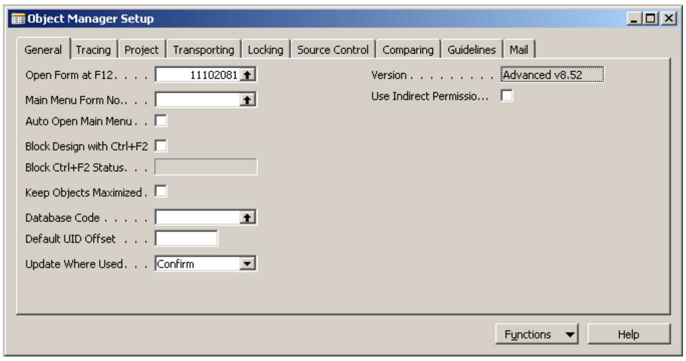
In NAV 3.7 and before you could enter a form number in the User Setup. This form opened when you started NAV or when you pressed F12.

Since NAV version 4 this is no longer possible because this field is deleted from the user setup. But there is a work around to open e.g. the Object Explorer when you press F12.

Add the following code to codeunit 1 – Application Management.

```
PROCEDURE OpenMenu@1() : Integer;
VAR
  Setup@1000 : Record 11102035;
BEGIN
  Setup.CustomGet;
  EXIT(Setup."Open Form at F12");
END;
```

NAV will open the “Open Form at F12” form that you have entered in the setup.



Open Form at F12. . . . 11102081
Main Menu Form No.. . .
Auto Open Main Menu . . ☐
Block Design with Ctrl+F2 ☐
Block Ctrl+F2 Status. . .
Keep Objects Maximized . ☐
Database Code . . . . .
Default UID Offset . . .
Update Where Used. . . Confirm
Version . . . . Advanced v8.52
Use Indirect Permissio... ☐

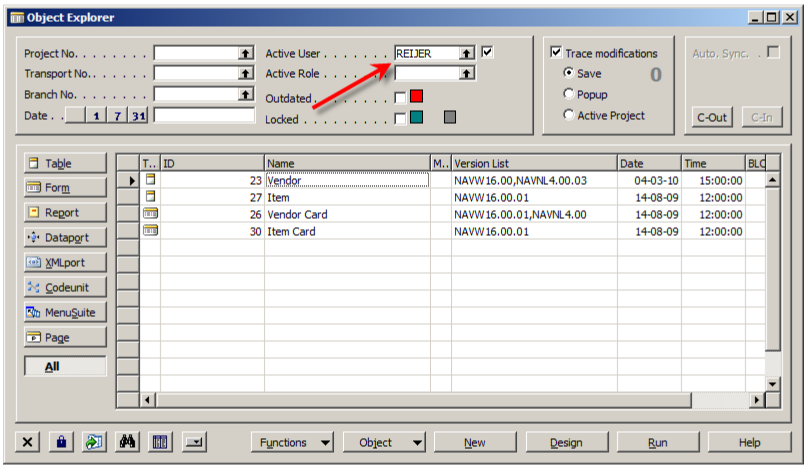
If there is a risk that codeunit 1 will be transported to a customer database without the “OM – Setup” table you will have to add the following function. This will compile without the Object Manager.

```
PROCEDURE OpenMenu@1() : Integer;
VAR
  Object@1000 : Record 2000000001;
  RecRef@1001 : RecordRef;
  FldRef@1002 : FieldRef;
BEGIN
  IF Object.GET(Object.Type::Table, '', DATABASE::"OM - Setup") THEN BEGIN
    RecRef.OPEN(DATABASE::"OM - Setup");
    FldRef := RecRef.FIELD(1);
    FldRef.SETRANGE(UPPERCASE(USERID));
    IF NOT RecRef.FINDFIRST THEN BEGIN
      FldRef.SETRANGE('');
      IF RecRef.FINDFIRST THEN
        ;
    END;

    FldRef := RecRef.FIELD(64);
    EXIT(FldRef.VALUE);
  END;
END;
```

NOTE: Since NAV6.01 NAV uses function id 1 for the local function GetCurrency. You will have to change the id of this function to e.g. 1000 because you cannot have two functions with the same id.

7.3. Filter Objects

Selecting the checkbox on the right or selecting the “Active User” will show all your objects where you are working on.



Active User: REIJER ☑

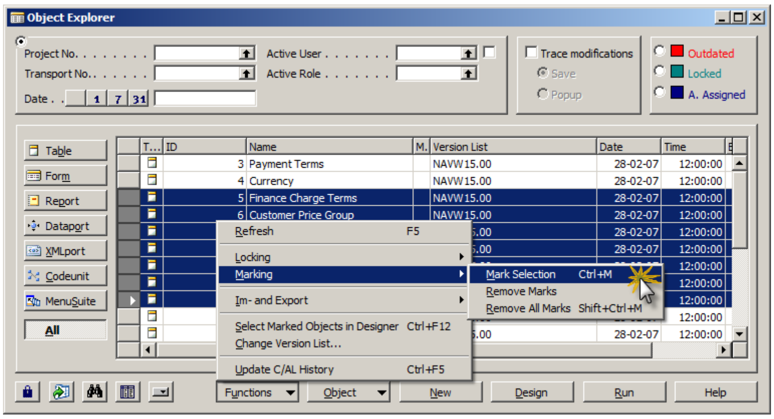
☑ Trace modifications ● Save

T..	ID	Name	M..	Version List	Date	Time
	23	Vendor		NAVW16.00,NAVNL4.00.03	04-03-10	15:00:00
	27	Item		NAVW16.00.01	14-08-09	12:00:00
	26	Vendor Card		NAVW16.00.01,NAVNL4.00	14-08-09	12:00:00
	30	Item Card		NAVW16.00.01	14-08-09	12:00:00

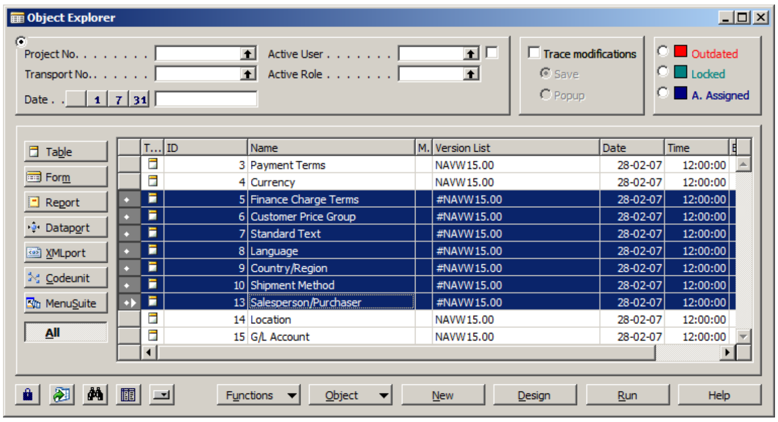
It is also possible to filter on project, transport, branch, date and role.

7.4. Mark Objects

Mark Objects is used when you want to select the objects of a particular project or transport in the Object Explorer.



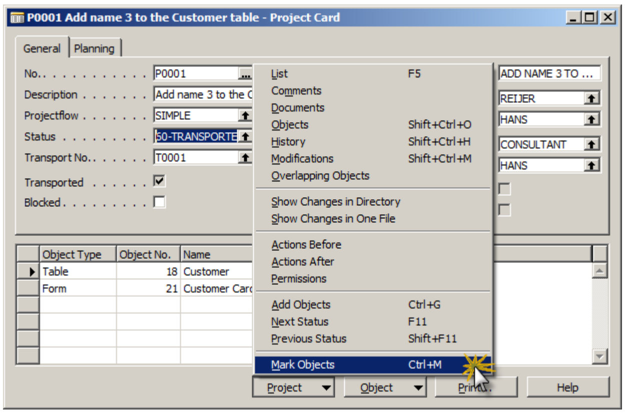
T...	ID	Name	M.	Version List	Date	Time
	3	Payment Terms		NAVW15.00	28-02-07	12:00:00
	4	Currency		NAVW15.00	28-02-07	12:00:00
	5	Finance Charge Terms		NAVW15.00	28-02-07	12:00:00
	6	Customer Price Group		NAVW15.00	28-02-07	12:00:00



ID	Name	Version List	Date	Time
3	Payment Terms	NAVW15.00	28-02-07	12:00:00
4	Currency	NAVW15.00	28-02-07	12:00:00
5	Finance Charge Terms	#NAVW15.00	28-02-07	12:00:00
6	Customer Price Group	#NAVW15.00	28-02-07	12:00:00
7	Standard Text	#NAVW15.00	28-02-07	12:00:00
8	Language	#NAVW15.00	28-02-07	12:00:00
9	Country/Region	#NAVW15.00	28-02-07	12:00:00
10	Shipment Method	#NAVW15.00	28-02-07	12:00:00
13	Salesperson/Purchaser	#NAVW15.00	28-02-07	12:00:00
14	Location	NAVW15.00	28-02-07	12:00:00
15	G/L Account	NAVW15.00	28-02-07	12:00:00

The Object Manager will add a "#" to the "Version List" field of the Objects. This makes it easy to select them in the Object Designer.

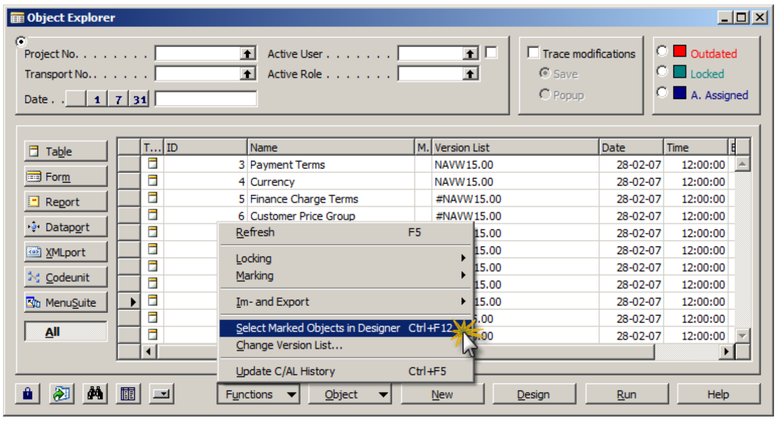
You can mark the objects of a project or transport. Because a "#" is added to the "Version List" you can easely select the objects in the Object Designer. This can be used if you want to make e.g. a FOB File.



No. P0001
Description . . . . Add name 3 to the C
Projectflow . . . . SIMPLE
Status 50-TRANSPORTE
Transport No. T0001
Transported . . . . . . ☑
Blocked . . . . ☐

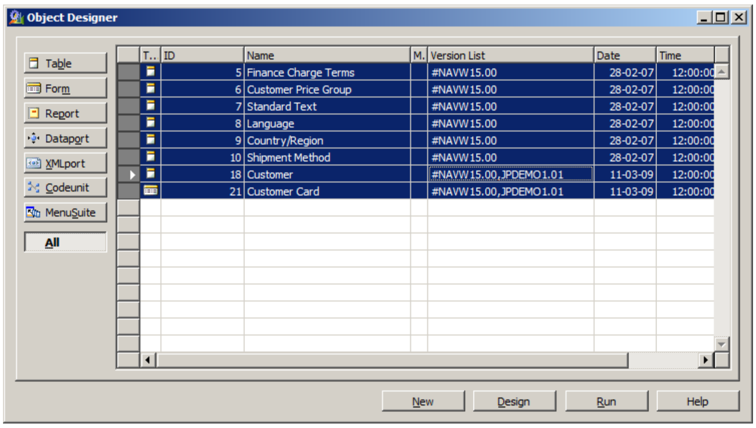
ADD NAME 3 TO ...
REIJER
HANS
CONSULTANT
HANS
☐
☐

Object Type	Object No.	Name
Table	18	Customer
Form	21	Customer Car



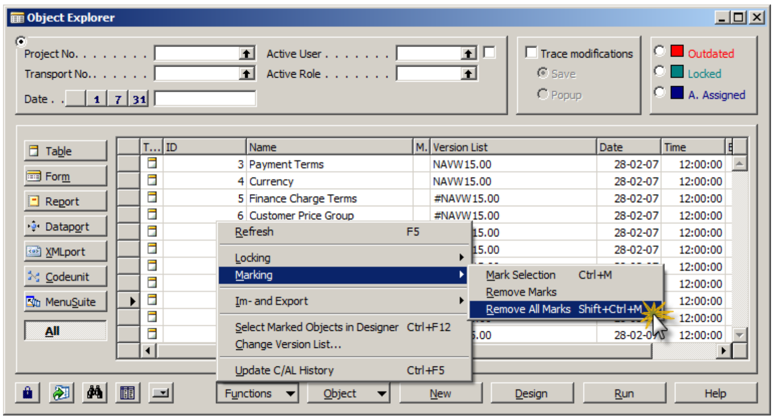
ID	Name	Version List	Date	Time
3	Payment Terms	NAVW15.00	28-02-07	12:00:00
4	Currency	NAVW15.00	28-02-07	12:00:00
5	Finance Charge Terms	#NAVW15.00	28-02-07	12:00:00
6	Customer Price Group	#NAVW15.00	28-02-07	12:00:00

The NAV Object Designer opens and shows the selected set of objects.



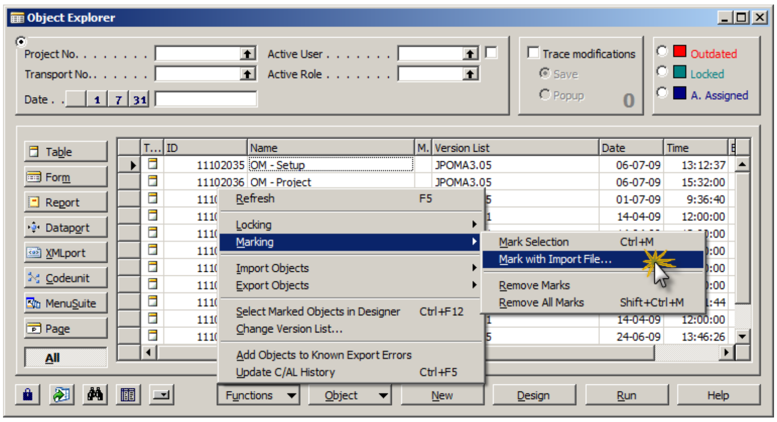
T..	ID	Name	M.	Version List	Date	Time
	5	Finance Charge Terms		#NAVW15.00	28-02-07	12:00:00
	6	Customer Price Group		#NAVW15.00	28-02-07	12:00:00
	7	Standard Text		#NAVW15.00	28-02-07	12:00:00
	8	Language		#NAVW15.00	28-02-07	12:00:00
	9	Country/Region		#NAVW15.00	28-02-07	12:00:00
	10	Shipment Method		#NAVW15.00	28-02-07	12:00:00
	18	Customer		#NAVW15.00,JPDEMO1.01	11-03-09	12:00:00
	21	Customer Card		#NAVW15.00,JPDEMO1.01	11-03-09	12:00:00

With the button Functions > Markings > Remove you can remove any markings you made.

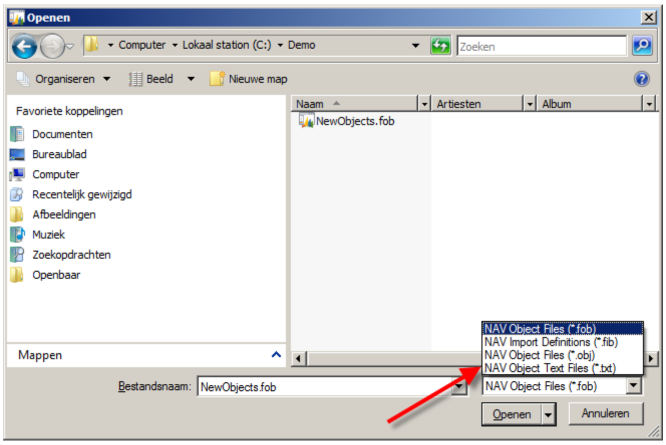


7.5. Mark Objects with Input File

Sometimes when you get a fob file or text file you want to set the original objects aside. Or export the original object in text format to compare/merge them. This can be done with the “Mark with Import File…” option.



Click the option and pick the object file.



NewObjects.fob

Bestandsnaam: NewObjects.fob

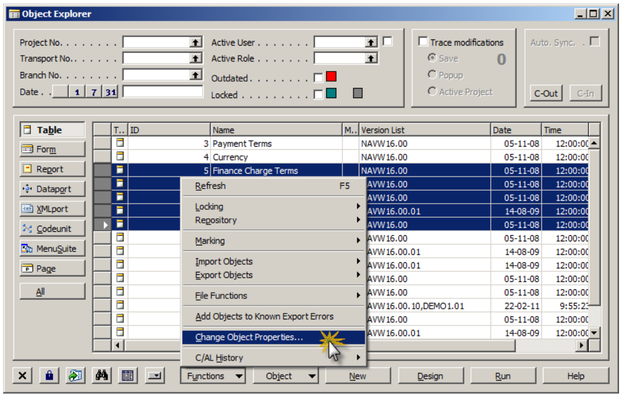
NAV Object Files (*.fob)
NAV Import Definitions (*.fib)
NAV Object Files (*.obj)
NAV Object Text Files (*.txt)

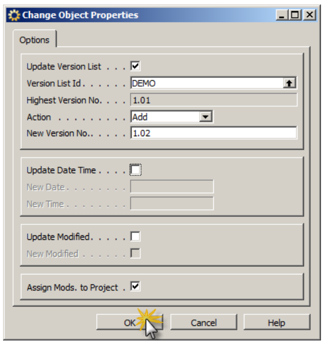
Here you can choose between various formats. The Object Manager scans this file and marks the corresponding objects in your database with a “#”.

7.6. Change Object Properties

With this option you can change the version list, date, time and modified flag of the selected objects without the need of creating projects or transports.

When you want e.g. to add DEMO1.02 to the version list you enable the “Update Version List” and fill the right parameters.




Change Object Properties

Options

Update Version List ☑

Version List Id: DEMO

Highest Version No.: 1.01

Action: Add

New Version No.: 1.02

Update Date Time ☐

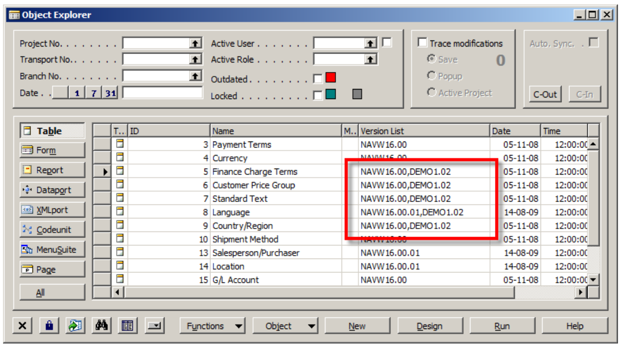
New Date

New Time

Update Modified ☐

New Modified ☐

Assign Mods. to Project ☑



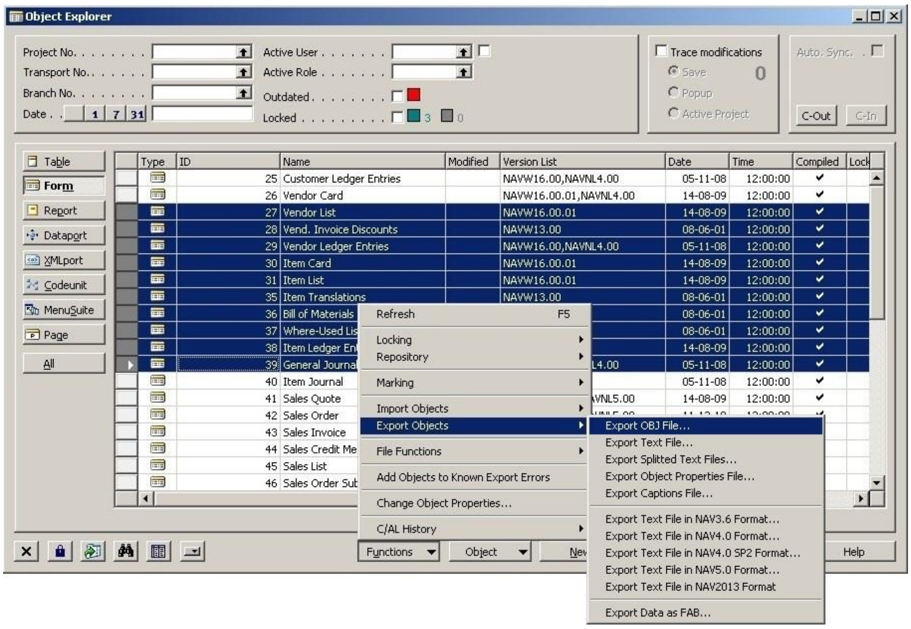
ID	Name	Version List	Date	Time
3	Payment Terms	NAVW16.00	05-11-08	12:00:00
4	Currency	NAVW16.00	05-11-08	12:00:00
5	Finance Charge Terms	NAVW16.00,DEMO1.02	05-11-08	12:00:00
6	Customer Price Group	NAVW16.00,DEMO1.02	05-11-08	12:00:00
7	Standard Text	NAVW16.00,DEMO1.02	05-11-08	12:00:00
8	Language	NAVW16.00.01,DEMO1.02	14-08-09	12:00:00
9	Country/Region	NAVW16.00,DEMO1.02	05-11-08	12:00:00
10	Shipment Method	NAVW16.00	05-11-08	12:00:00
13	Salesperson/Purchaser	NAVW16.00.01	14-08-09	12:00:00
14	Location	NAVW16.00.01	14-08-09	12:00:00
15	G/L Account	NAVW16.00	05-11-08	12:00:00

Is you disable the option “Assign Mods. to Project” the modifications are not traced and not saved to the modification table and therefore not needed to be assigned to a project.

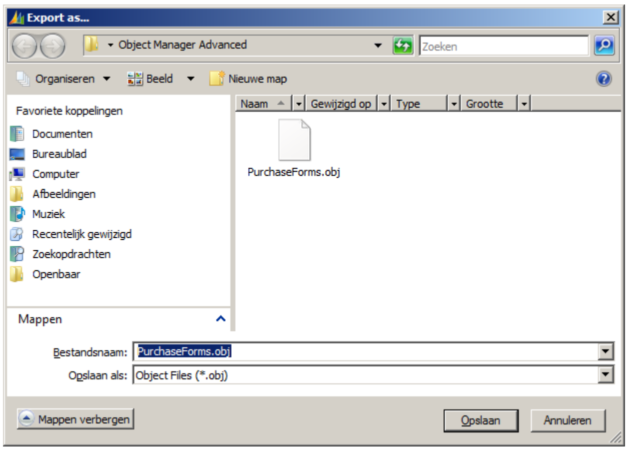
7.7. Import and Export Files

7.7.1. OBJ Files

With the button Functions > “Export Objects“ > “Export OBJ File” you export the selected objects in OBJ format.



Type	ID	Name	Modified	Version List	Date	Time	Compiled	Lock
	25	Customer Ledger Entries		NAVW16.00,NAVNL4.00	05-11-08	12:00:00	✓	
	26	Vendor Card		NAVW16.00.01,NAVNL4.00	14-08-09	12:00:00	✓	
	27	Vendor List		NAVW16.00.01	14-08-09	12:00:00	✓	
	28	Vend. Invoice Discounts		NAVW13.00	08-06-01	12:00:00	✓	
	29	Vendor Ledger Entries		NAVW16.00,NAVNL4.00	05-11-08	12:00:00	✓	
	30	Item Card		NAVW16.00.01	14-08-09	12:00:00	✓	
	31	Item List		NAVW16.00.01	14-08-09	12:00:00	✓	
	35	Item Translations		NAVW13.00	08-06-01	12:00:00	✓	
	36	Bill of Materials			08-06-01	12:00:00	✓	
	37	Where-Used Lis			08-06-01	12:00:00	✓	
	38	Item Ledger En			14-08-09	12:00:00	✓	
	39	General Journa		L4.00	05-11-08	12:00:00	✓	
	40	Item Journal			05-11-08	12:00:00	✓	
	41	Sales Quote		VNL5.00	14-08-09	12:00:00	✓	
	42	Sales Order						
	43	Sales Invoice						
	44	Sales Credit Me						
	45	Sales List						
	46	Sales Order Sub						

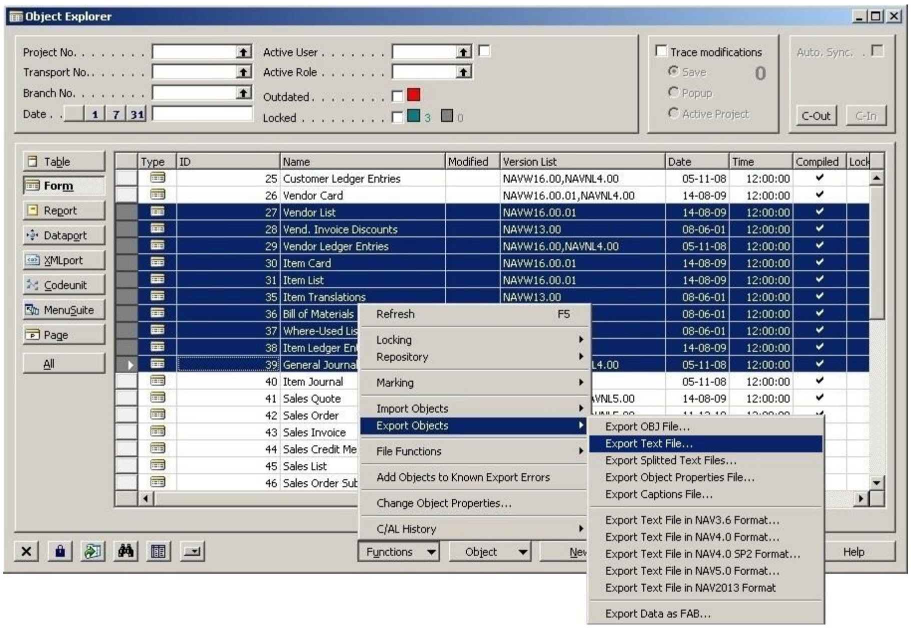


Bestandsnaam: PurchaseForms.obj
Opslaan als: Object Files (*.obj)

When importing an OBJ file you will get the “Object Import Worksheet”.

7.7.2. Text Files

With the button Functions > “Export Objects” > “Export Text File” you export the selected objects in text format.



Type	ID	Name	Modified	Version List	Date	Time	Compiled	Lock
	25	Customer Ledger Entries		NAVW16.00,NAVNL4.00	05-11-08	12:00:00	✓	
	26	Vendor Card		NAVW16.00.01,NAVNL4.00	14-08-09	12:00:00	✓	
	27	Vendor List		NAVW16.00.01	14-08-09	12:00:00	✓	
	28	Vend. Invoice Discounts		NAVW13.00	08-06-01	12:00:00	✓	
	29	Vendor Ledger Entries		NAVW16.00,NAVNL4.00	05-11-08	12:00:00	✓	
	30	Item Card		NAVW16.00.01	14-08-09	12:00:00	✓	
	31	Item List		NAVW16.00.01	14-08-09	12:00:00	✓	
	35	Item Translations		NAVW13.00	08-06-01	12:00:00	✓	
	36	Bill of Materials			08-06-01	12:00:00	✓	
	37	Where-Used Lis			08-06-01	12:00:00	✓	
	38	Item Ledger En			14-08-09	12:00:00	✓	
	39	General Journal		L4.00	05-11-08	12:00:00	✓	
	40	Item Journal			05-11-08	12:00:00	✓	
	41	Sales Quote		VNL5.00	14-08-09	12:00:00	✓	
	42	Sales Order						
	43	Sales Invoice						
	44	Sales Credit Me						
	45	Sales List						
	46	Sales Order Sub						

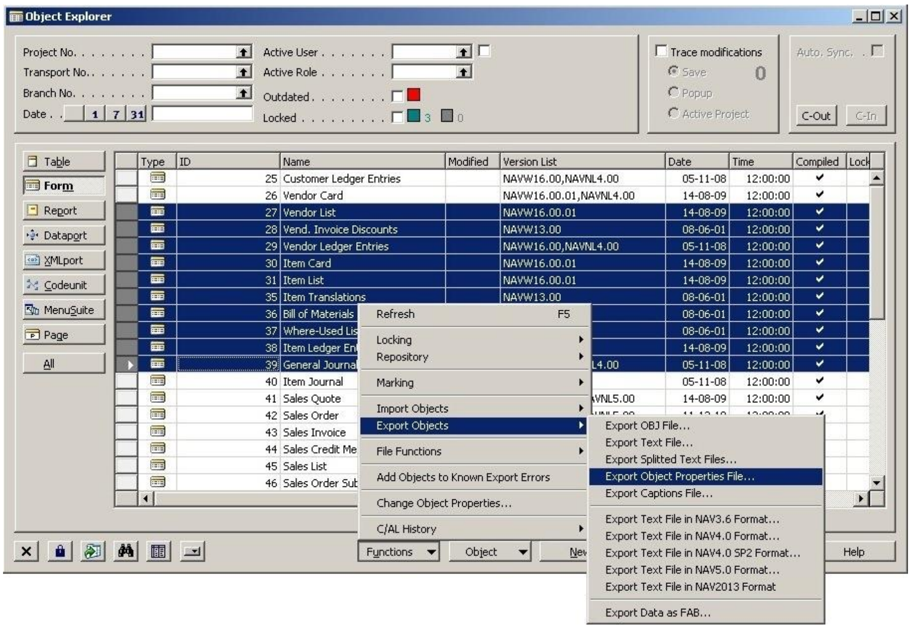
NOTE: When you do not have installed the Object Manager setup you will im- and export the objects from and to the “C/AL History”. Not straight to the object table as you do when im- and exporting from the Object Designer.

With the button Functions > “Import Objects” > “Import Text File” you import a file in text format. This file will be opened in the “Object Import Worksheet”.

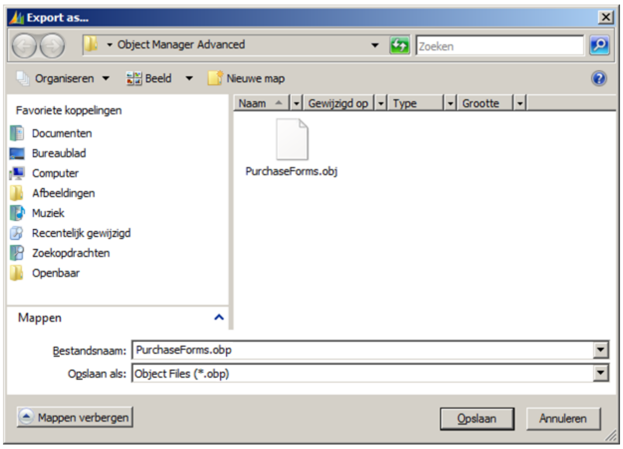
If you have separated text files in a directory you can use “Import Splitted Text Files”. This is often used in combination with the function Split Text Files which will be described below.

7.7.3. Object Properties Files (OBP)

With the button Functions > “Export Objects“ > “Export Object Properties File” you export the object properties, like Date, Time and Version List, of the selected objects in OBP format.



Type	ID	Name	Modified	Version List	Date	Time	Compiled	Lock
	25	Customer Ledger Entries		NAVW16.00,NAVNL4.00	05-11-08	12:00:00	✓	
	26	Vendor Card		NAVW16.00.01,NAVNL4.00	14-08-09	12:00:00	✓	
	27	Vendor List		NAVW16.00.01	14-08-09	12:00:00	✓	
	28	Vend. Invoice Discounts		NAVW13.00	08-06-01	12:00:00	✓	
	29	Vendor Ledger Entries		NAVW16.00,NAVNL4.00	05-11-08	12:00:00	✓	
	30	Item Card		NAVW16.00.01	14-08-09	12:00:00	✓	
	31	Item List		NAVW16.00.01	14-08-09	12:00:00	✓	
	35	Item Translations		NAVW13.00	08-06-01	12:00:00	✓	
	36	Bill of Materials			08-06-01	12:00:00	✓	
	37	Where-Used Lis			08-06-01	12:00:00	✓	
	38	Item Ledger En			14-08-09	12:00:00	✓	
	39	General Journal		L4.00	05-11-08	12:00:00	✓	
	40	Item Journal			05-11-08	12:00:00	✓	
	41	Sales Quote		VNL5.00	14-08-09	12:00:00	✓	
	42	Sales Order						
	43	Sales Invoice						
	44	Sales Credit Me						
	45	Sales List						
	46	Sales Order Sub						



Bestandsnaam: PurchaseForms.obp
Opslaan als: Object Files (*.obp)

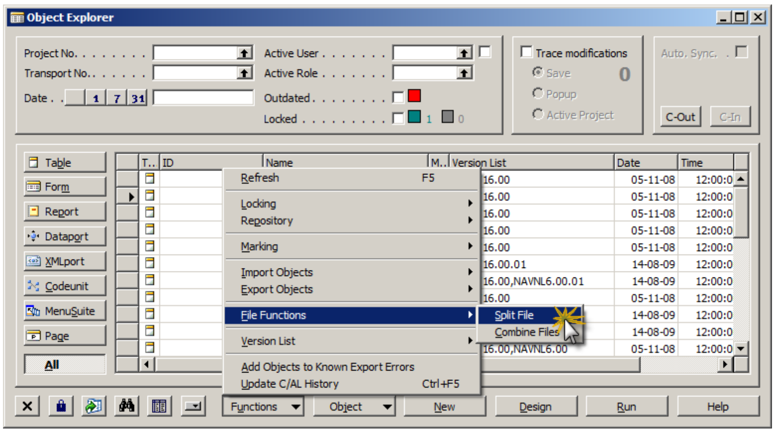
When importing an OBP file you will get the “Object Import Worksheet”.

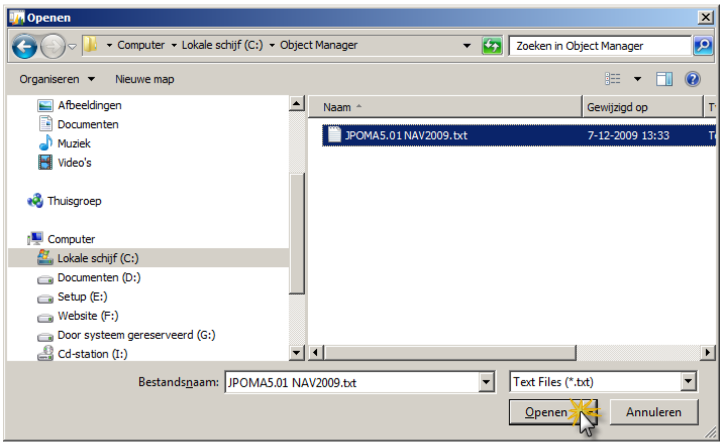
7.8. Split Files

The file function “Split File” allows you to split up a single file containing multiple objects into multiple files each containing one object. This can be used to split up:

- Object text files (.txt)
- Translation files (.txt)
- Import definition files (.fib)

Press Functions > “File Functions” > “Split File”.



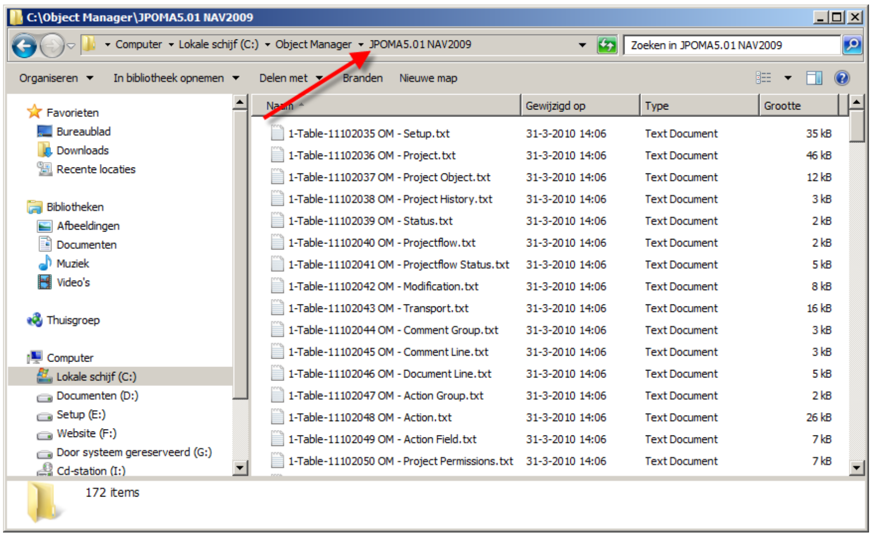


JPOMA5.01 NAV2009.txt 7-12-2009 13:33

Bestandsnaam: JPOMA5.01 NAV2009.txt

Text Files (*.txt)

This will create a new directory with the name of the text file and saves all objects in seperated files.



Naam	Gewijzigd op	Type	Grootte
1-Table-11102035 OM - Setup.txt	31-3-2010 14:06	Text Document	35 kB
1-Table-11102036 OM - Project.txt	31-3-2010 14:06	Text Document	46 kB
1-Table-11102037 OM - Project Object.txt	31-3-2010 14:06	Text Document	12 kB
1-Table-11102038 OM - Project History.txt	31-3-2010 14:06	Text Document	3 kB
1-Table-11102039 OM - Status.txt	31-3-2010 14:06	Text Document	2 kB
1-Table-11102040 OM - Projectflow.txt	31-3-2010 14:06	Text Document	2 kB
1-Table-11102041 OM - Projectflow Status.txt	31-3-2010 14:06	Text Document	5 kB
1-Table-11102042 OM - Modification.txt	31-3-2010 14:06	Text Document	8 kB
1-Table-11102043 OM - Transport.txt	31-3-2010 14:06	Text Document	16 kB
1-Table-11102044 OM - Comment Group.txt	31-3-2010 14:06	Text Document	3 kB
1-Table-11102045 OM - Comment Line.txt	31-3-2010 14:06	Text Document	3 kB
1-Table-11102046 OM - Document Line.txt	31-3-2010 14:06	Text Document	5 kB
1-Table-11102047 OM - Action Group.txt	31-3-2010 14:06	Text Document	2 kB
1-Table-11102048 OM - Action.txt	31-3-2010 14:06	Text Document	26 kB
1-Table-11102049 OM - Action Field.txt	31-3-2010 14:06	Text Document	7 kB
1-Table-11102050 OM - Project Permissions.txt	31-3-2010 14:06	Text Document	7 kB

172 items

NOTE: Another way to get all your objects in seperated object text files is to export them with the function Functions > “Export

Objects" > "Export Splitted Text Files". You will be prompted for a directory and the objects will be exported to that directory.

7.9. Combine Text Files

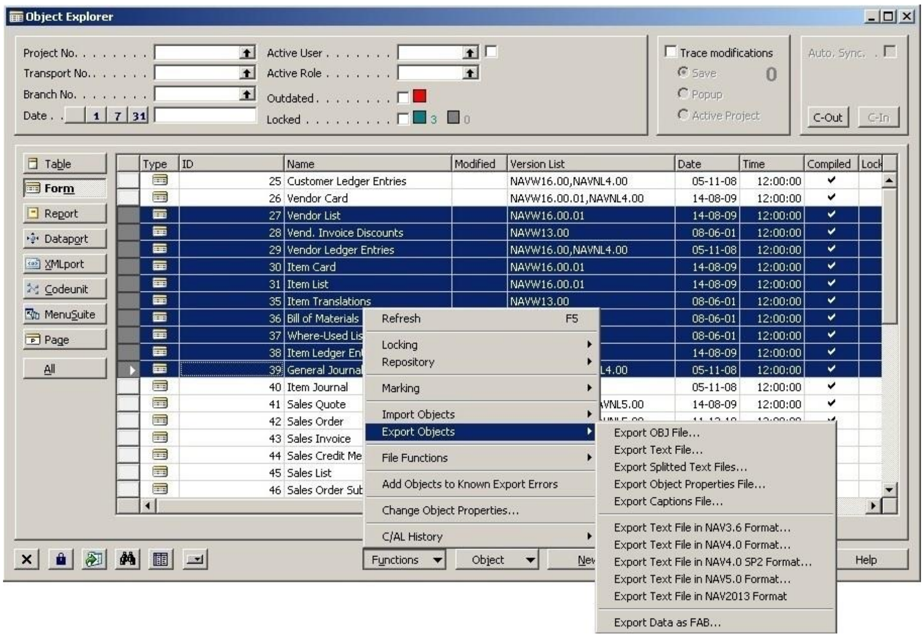
To combine separated objects into one text file you can use the function Functions > "File Functions" > "Combine Files". You will be prompted for a directory and all the text files in that directory will be placed in a new text file with the same name of the chosen directory. This applies to both translation and object text files.

It is also possible to import all seperated text files straight into NAV by selection the function Functions > "Import Objects" > "Import Splitted Text Files". This will import all objects located in the chosen directory.

7.10. Downgrade Objects

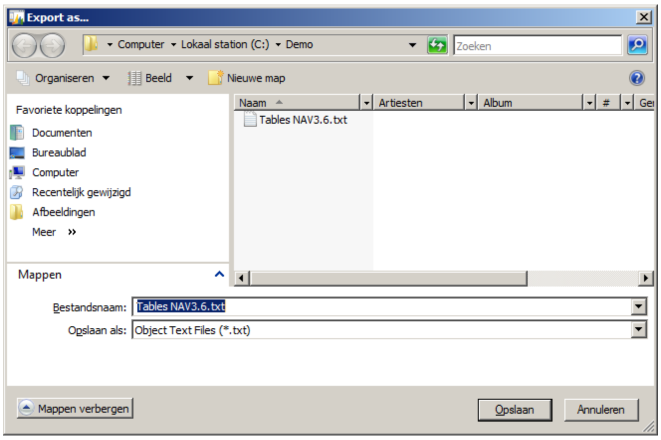
Since NAV2009 it is not possible to run your objects in older NAV versions. NAV will crash if you run your object in e.g. NAV5.0. The only way to import your objects in older versions is in text format. In the Object Explorer you can export your objects in older NAV formats. When you export in e.g. NAV4.0 format the FINDSET en FINDFIRST will be replaced by FIND('-') and non-supported elements and properties will be removed.

Select the objects you want to convert and press one of the NAV formats.



ID	Name	Modified	Version List	Date	Time	Compiled	Lock
25	Customer Ledger Entries		NAVW16.00,NAVNL4.00	05-11-08	12:00:00	✓	
26	Vendor Card		NAVW16.00.01,NAVNL4.00	14-08-09	12:00:00	✓	
27	Vendor List		NAVW16.00.01	14-08-09	12:00:00	✓	
28	Vend. Invoice Discounts		NAVW13.00	08-06-01	12:00:00	✓	
29	Vendor Ledger Entries		NAVW16.00,NAVNL4.00	05-11-08	12:00:00	✓	
30	Item Card		NAVW16.00.01	14-08-09	12:00:00	✓	
31	Item List		NAVW16.00.01	14-08-09	12:00:00	✓	
35	Item Translations		NAVW13.00	08-06-01	12:00:00	✓	
36	Bill of Materials			08-06-01	12:00:00	✓	
37	Where-Used Lis			08-06-01	12:00:00	✓	
38	Item Ledger En			14-08-09	12:00:00	✓	
39	General Journal		L4.00	05-11-08	12:00:00	✓	
40	Item Journal			05-11-08	12:00:00	✓	
41	Sales Quote		VNL5.00	14-08-09	12:00:00	✓	
42	Sales Order						
43	Sales Invoice						
44	Sales Credit Me						
45	Sales List						
46	Sales Order Sub						

You will be prompted to enter a filename:

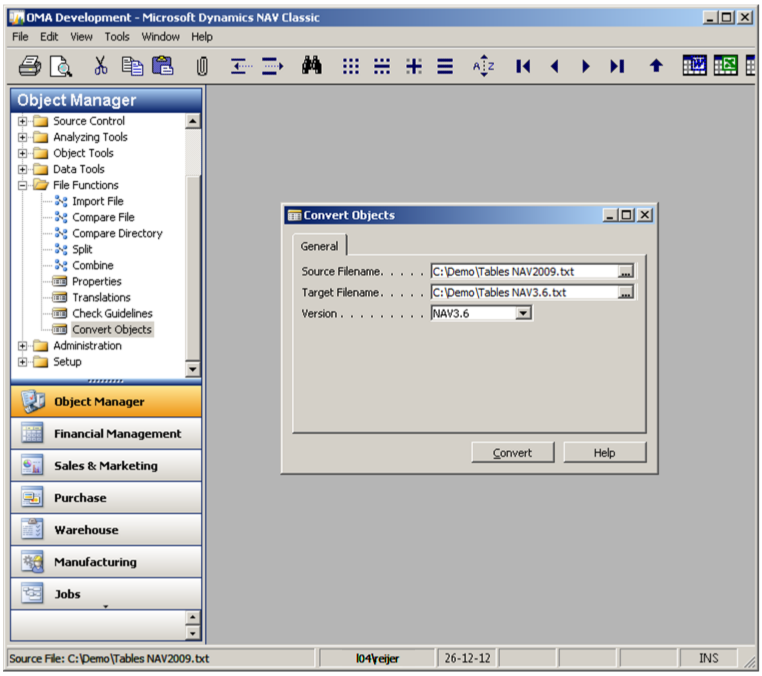


Bestandsnaam: Tables NAV3.6.txt

Opslaan als: Object Text Files (*.txt)

You can import the created file in the Object Designer.

It is also possible to convert an existing file with the "Convert Objects" window.



Source Filename. . . . . C:\Demo\Tables NAV2009.txt

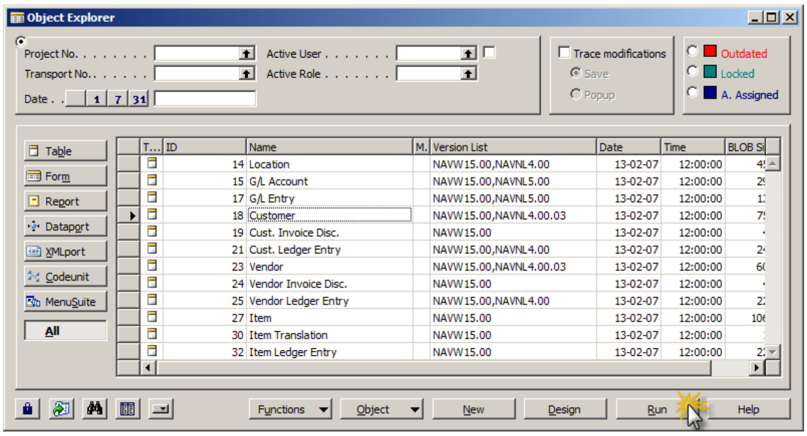
Target Filename. . . . . C:\Demo\Tables NAV3.6.txt

Version NAV3.6

Source File: C:\Demo\Tables NAV2009.txt

7.11. Show Table Data

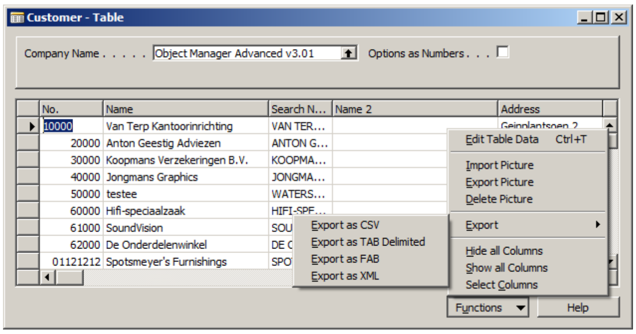
When pressing the Run button at the Object Explorer you will not get the default table view from NAV but a form with more options.



Project No. . . . .
Transport No.. . . . . .
Date . .
Active User . . . .
Active Role . . . .

T...	ID	Name	M.	Version List	Date	Time	BLOB Si
	14	Location		NAVW15.00,NAVNL4.00	13-02-07	12:00:00	4
	15	G/L Account		NAVW15.00,NAVNL5.00	13-02-07	12:00:00	2
	17	G/L Entry		NAVW15.00,NAVNL5.00	13-02-07	12:00:00	1
	18	Customer		NAVW15.00,NAVNL4.00.03	13-02-07	12:00:00	7
	19	Cust. Invoice Disc.		NAVW15.00	13-02-07	12:00:00	
	21	Cust. Ledger Entry		NAVW15.00,NAVNL4.00	13-02-07	12:00:00	2
	23	Vendor		NAVW15.00,NAVNL4.00.03	13-02-07	12:00:00	6
	24	Vendor Invoice Disc.		NAVW15.00	13-02-07	12:00:00	
	25	Vendor Ledger Entry		NAVW15.00,NAVNL4.00	13-02-07	12:00:00	2
	27	Item		NAVW15.00	13-02-07	12:00:00	10
	30	Item Translation		NAVW15.00	13-02-07	12:00:00	
	32	Item Ledger Entry		NAVW15.00	13-02-07	12:00:00	2

In this form it is possible to import and export your BLOB fields or export the data to various formats as CSV and XML.

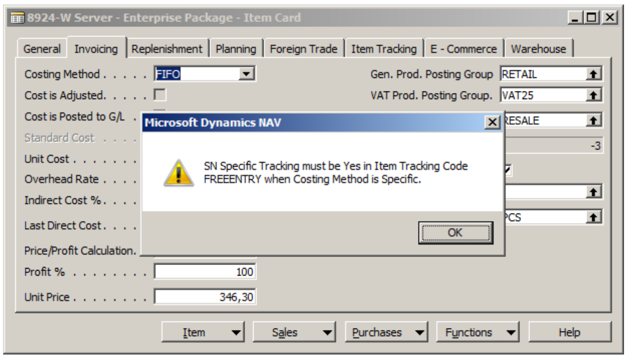


Company Name . . . . . Object Manager Advanced v3.01
Options as Numbers . . . ☐

No.	Name	Search N...	Name 2	Address
10000	Van Terp Kantoorinrichting	VAN TER...		Geinplantsoen 2
20000	Anton Geestig Adviezen	ANTON G...		
30000	Koopmans Verzekeringen B.V.	KOOPMA...		
40000	Jongmans Graphics	JONGMA...		
50000	testee	WATERS...		
60000	Hifi-speciaalzaak	HIFI-SPE...		
61000	SoundVision	SOU		
62000	De Onderdelenwinkel	DE C		
01121212	Spotsmeyer's Furnishings	SPO		

7.12. Edit Table Data

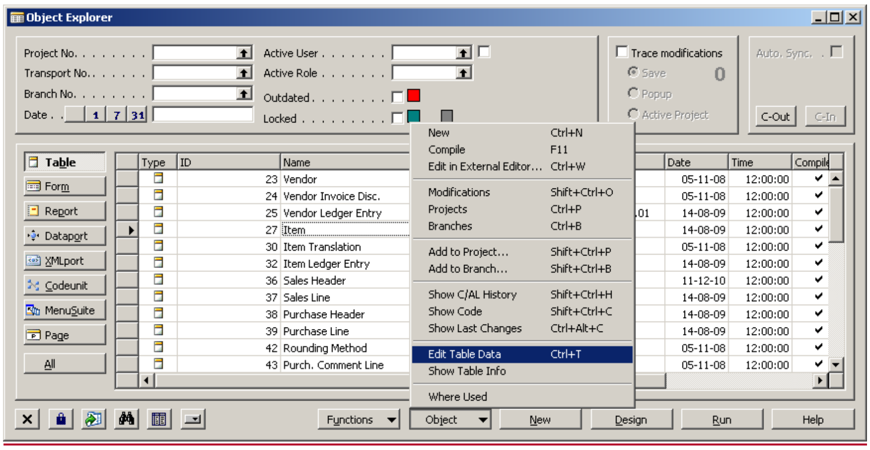
Sometimes you want to change data without it being validated in the database. E.g. when you want to change the “Costing Method” field on the “Item Card” you can get the following error if the “SN Specific Tracking” is off.



Costing Method . . . . . FIFO
Cost is Adjusted. . . . . ☐
Cost is Posted to G/L .
Standard Cost . . . .
Unit Cost . . . .
Overhead Rate . . . .
Indirect Cost %. . . .
Last Direct Cost. . . .
Price/Profit Calculation.
Profit % 100
Unit Price 346,30
Gen. Prod. Posting Group RETAIL
VAT Prod. Posting Group. VAT25

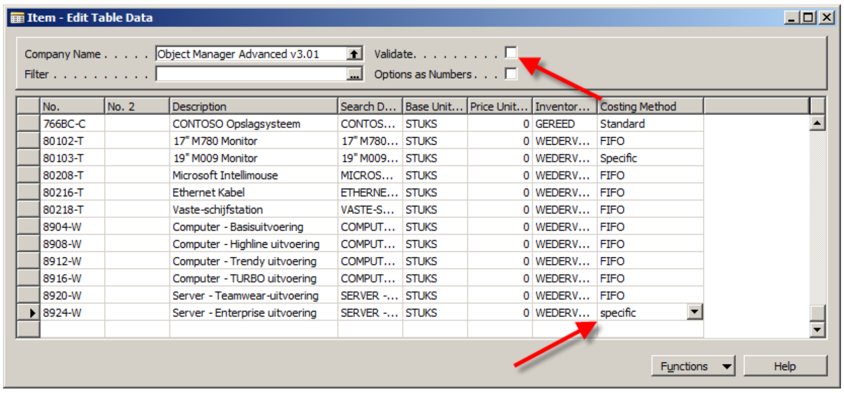
SN Specific Tracking must be Yes in Item Tracking Code FREEENTRY when Costing Method is Specific.

In this example we fill the “Costing Method” field without validation in the Item table using the “Edit Table Data” option in the Object Explorer.



ID	Name	Date	Time	Compile
23	Vendor	05-11-08	12:00:00	✓
24	Vendor Invoice Disc.	05-11-08	12:00:00	✓
25	Vendor Ledger Entry	14-08-09	12:00:00	✓
27	Item	14-08-09	12:00:00	✓
30	Item Translation	05-11-08	12:00:00	✓
32	Item Ledger Entry	14-08-09	12:00:00	✓
36	Sales Header	11-12-10	12:00:00	✓
37	Sales Line	14-08-09	12:00:00	✓
38	Purchase Header	14-08-09	12:00:00	✓
39	Purchase Line	14-08-09	12:00:00	✓
42	Rounding Method	05-11-08	12:00:00	✓
43	Purch. Comment Line	05-11-08	12:00:00	✓

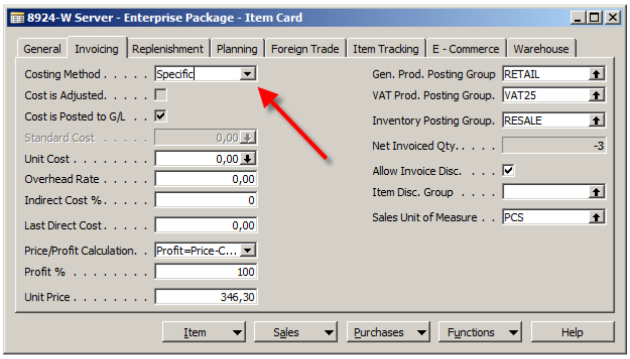
In the table data you can change the “Costing Method” of that specific record manually. With the checkbox validation you can choose to validate or not the changes you make. In our example we leave it off.



Company Name . . . . . Object Manager Advanced v3.01
Filter . . . .
Validate. . . . . ☐
Options as Numbers . . . ☐

No.	No. 2	Description	Search D…	Base Unit…	Price Unit…	Inventor…	Costing Method
766BC-C		CONTOSO Opslagsysteem	CONTOS…	STUKS	0	GEREED	Standard
80102-T		17" M780 Monitor	17" M780…	STUKS	0	WEDERV…	FIFO
80103-T		19" M009 Monitor	19" M009…	STUKS	0	WEDERV…	Specific
80208-T		Microsoft Intellimouse	MICROS…	STUKS	0	WEDERV…	FIFO
80216-T		Ethernet Kabel	ETHERNE…	STUKS	0	WEDERV…	FIFO
80218-T		Vaste-schijfstation	VASTE-S…	STUKS	0	WEDERV…	FIFO
8904-W		Computer - Basisuitvoering	COMPUT…	STUKS	0	WEDERV…	FIFO
8908-W		Computer - Highline uitvoering	COMPUT…	STUKS	0	WEDERV…	FIFO
8912-W		Computer - Trendy uitvoering	COMPUT…	STUKS	0	WEDERV…	FIFO
8916-W		Computer - TURBO uitvoering	COMPUT…	STUKS	0	WEDERV…	FIFO
8920-W		Server - Teamwear-uitvoering	SERVER -…	STUKS	0	WEDERV…	FIFO
8924-W		Server - Enterprise uitvoering	SERVER -…	STUKS	0	WEDERV…	specific

When you return to the “Item Card” itself you will see that the “Costing Method” is changed without that the error that has taken place.

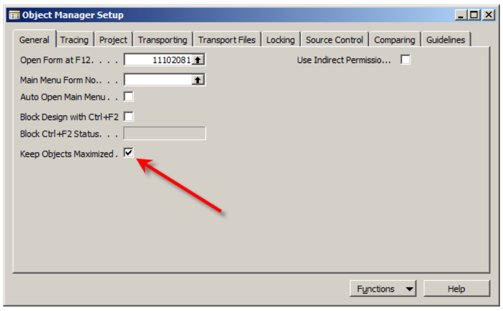


Costing Method . . . . . Specific
Cost is Adjusted. . . . . ☐
Cost is Posted to G/L . . ☑
Standard Cost 0,00
Unit Cost 0,00
Overhead Rate 0,00
Indirect Cost % 0
Last Direct Cost 0,00
Price/Profit Calculation. . Profit=Price-C…
Profit % 100
Unit Price 346,30
Gen. Prod. Posting Group RETAIL
VAT Prod. Posting Group. VAT25
Inventory Posting Group. RESALE
Net Invoiced Qty. . . . . -3
Allow Invoice Disc. . . . ☑
Item Disc. Group . . . .
Sales Unit of Measure . . PCS

NOTE: This option is only meant for your development database. It should not be used in your customer database because you can get inconsistent data.

7.13. Keep Objects Maximized

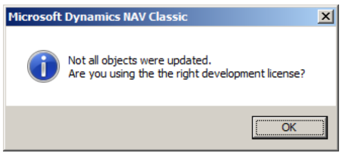
When you press design in the Object Explorer the Object Designer will be minimized. When enabling the setting “Keep Objects Maximized” all windows will remain in their own state.



Open Form at F12. . . . 11102081
Use Indirect Permissio... ☐
Main Menu Form No.. . .
Auto Open Main Menu . . ☐
Block Design with Ctrl+F2 ☐
Block Ctrl+F2 Status. . .
Keep Objects Maximized . ☑

7.14. Known Object Export Errors

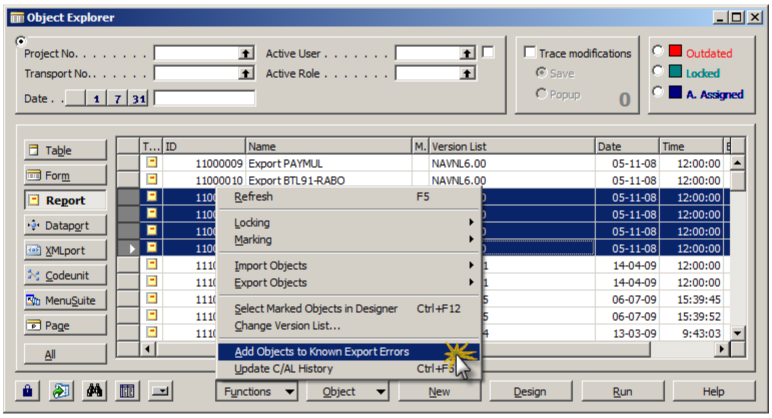
It can happen that some objects cannot be exported in text format. Because it has an error or because it is not in your license. There are several places where the Object Manager wants to export these objects and will show a message like:



Not all objects were updated.
Are you using the the right development license?

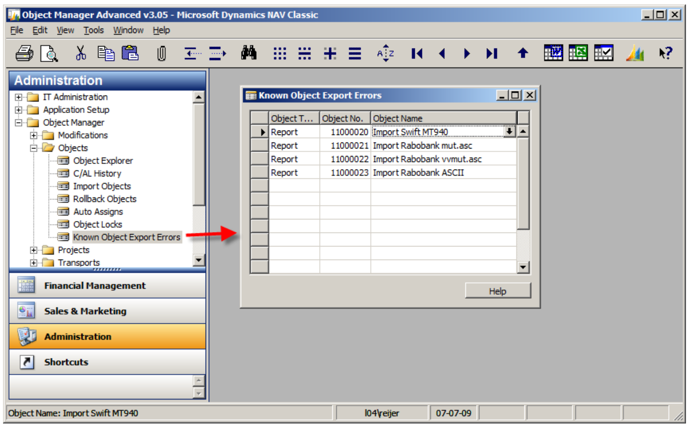
To stop the Object Manager showing these messages you can add these objects to the “Known Object Export Errors”.

Select these objects in the “Object Explorer” and press “Add Objects to Known Export Errors”.



T...	ID	Name	M.	Version List	Date	Time
	11000009	Export PAYMUL		NAVNL6.00	05-11-08	12:00:00
	11000010	Export BTL91-RABO		NAVNL6.00	05-11-08	12:00:00
	1100			0	05-11-08	12:00:00
	1100			0	05-11-08	12:00:00
	1100			0	05-11-08	12:00:00
	1100			0	05-11-08	12:00:00
	111			1	14-04-09	12:00:00
	111			1	14-04-09	12:00:00
	111			5	06-07-09	15:39:45
	111			5	06-07-09	15:39:52
	111			4	13-03-09	9:43:03

The objects are now added to the “Known Object Export Errors” form.

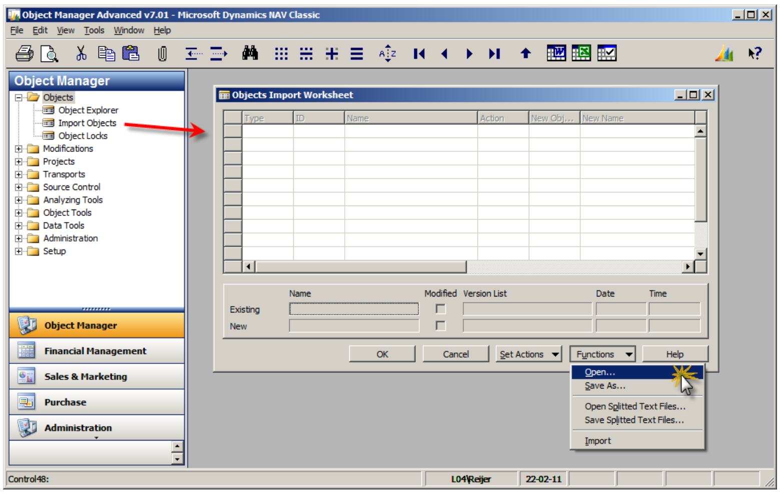


Object T...	Object No.	Object Name
Report	11000020	Import Swift MT940
Report	11000021	Import Rabobank mut.asc
Report	11000022	Import Rabobank vvmut.asc
Report	11000023	Import Rabobank ASCII

Object Name: Import Swift MT940 | l04\reijer | 07-07-09

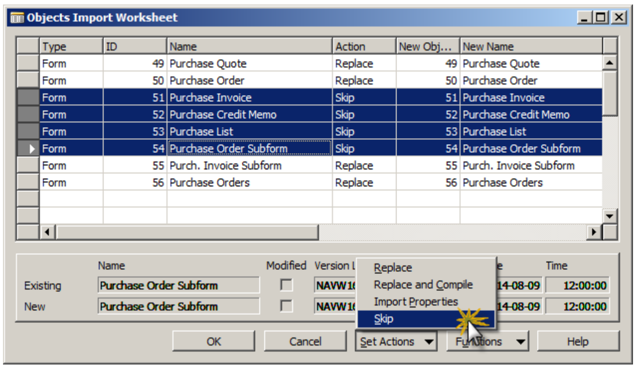
8. Objects Import Worksheet

The Objects Import Worksheet is used to import objects. You can open files in OBJ, TXT, OBP, and FIB formats. It is also possible to import a directory with splitted text files.



8.1. *Import Objects*

8.1.1. Import Options


Objects Import Worksheet

Type	ID	Name	Action	New Obj...	New Name
Form	49	Purchase Quote	Replace	49	Purchase Quote
Form	50	Purchase Order	Replace	50	Purchase Order
Form	51	Purchase Invoice	Skip	51	Purchase Invoice
Form	52	Purchase Credit Memo	Skip	52	Purchase Credit Memo
Form	53	Purchase List	Skip	53	Purchase List
Form	54	Purchase Order Subform	Skip	54	Purchase Order Subform
Form	55	Purch. Invoice Subform	Replace	55	Purch. Invoice Subform
Form	56	Purchase Orders	Replace	56	Purchase Orders

	Name	Modified	Version L	e	Time
Existing	Purchase Order Subform	☐	NAVW1	14-08-09	12:00:00
New	Purchase Order Subform	☐	NAVW1	14-08-09	12:00:00

Before importing the objects you can set the action that has to be executed.

- **Replace**
  Replaces the object without compiling.

- **Replace and Compile**
  Replaces the object and compiles it. This is often used in combination with the text format. When there is an error in one of the objects the complete import will be rolled back.

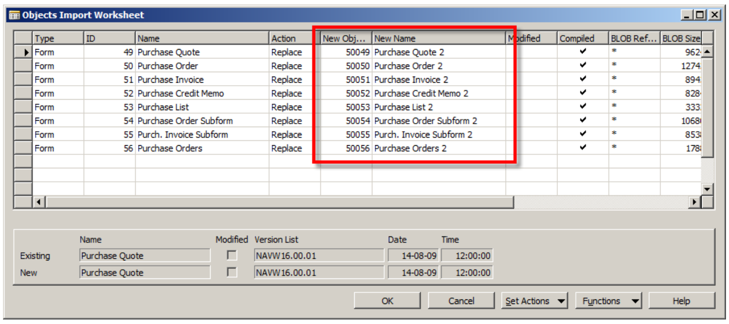
- **Import Properties**

This action only imports the properties of the objects. Properties are name, date, time, version list and modify flag.

- **Skip**
  Skips the object from import.

8.1.2. Change Id and/or Name

It is possible to give the objects another id and/or name.



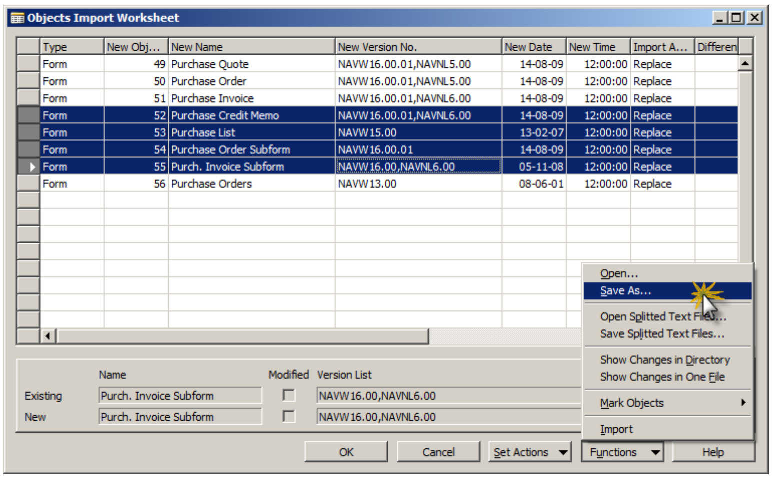
Type	ID	Name	Action	New Obj…	New Name	Modified	Compiled	BLOB Ref…	BLOB Size
Form	49	Purchase Quote	Replace	50049	Purchase Quote 2		✓	*	962
Form	50	Purchase Order	Replace	50050	Purchase Order 2		✓	*	1274
Form	51	Purchase Invoice	Replace	50051	Purchase Invoice 2		✓	*	894
Form	52	Purchase Credit Memo	Replace	50052	Purchase Credit Memo 2		✓	*	828
Form	53	Purchase List	Replace	50053	Purchase List 2		✓	*	333
Form	54	Purchase Order Subform	Replace	50054	Purchase Order Subform 2		✓	*	1068
Form	55	Purch. Invoice Subform	Replace	50055	Purch. Invoice Subform 2		✓	*	853
Form	56	Purchase Orders	Replace	50056	Purchase Orders 2		✓	*	178

	Name	Modified	Version List	Date	Time
Existing	Purchase Quote	☐	NAVW16.00.01	14-08-09	12:00:00
New	Purchase Quote	☐	NAVW16.00.01	14-08-09	12:00:00

NOTE: References to these objects will not be renumbered. Use the "Renumber Objects" tool if you want to renumber with updating all references. (For more information see chapter 21 - Renumber Objects)

8.2. Export Objects

It is also possible to select a number of objects and export them in the corresponding format as they were imported.



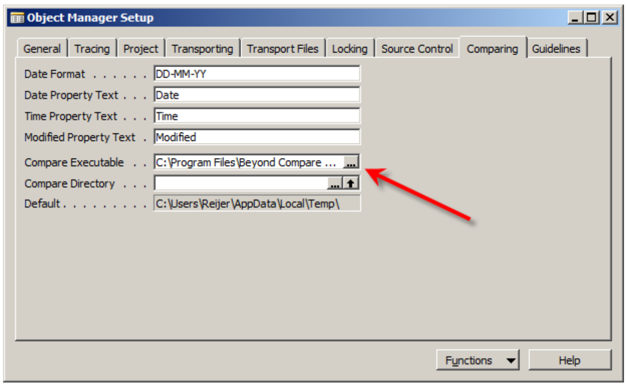
Type	New Obj…	New Name	New Version No.	New Date	New Time	Import A…	Differen
Form	49	Purchase Quote	NAVW16.00.01,NAVNL5.00	14-08-09	12:00:00	Replace	
Form	50	Purchase Order	NAVW16.00.01,NAVNL5.00	14-08-09	12:00:00	Replace	
Form	51	Purchase Invoice	NAVW16.00.01,NAVNL6.00	14-08-09	12:00:00	Replace	
Form	52	Purchase Credit Memo	NAVW16.00.01,NAVNL6.00	14-08-09	12:00:00	Replace	
Form	53	Purchase List	NAVW15.00	13-02-07	12:00:00	Replace	
Form	54	Purchase Order Subform	NAVW16.00.01	14-08-09	12:00:00	Replace	
Form	55	Purch. Invoice Subform	NAVW16.00,NAVNL6.00	05-11-08	12:00:00	Replace	
Form	56	Purchase Orders	NAVW13.00	08-06-01	12:00:00	Replace	

	Name	Modified	Version List
Existing	Purch. Invoice Subform	☐	NAVW16.00,NAVNL6.00
New	Purch. Invoice Subform	☐	NAVW16.00,NAVNL6.00

9. Version and Source Control

With this tool you can save versions on different trigger moments. Also you can start up a compare tool (like Beyond Compare) and compare the code with previous object versions in the system.

9.1. Setup



Date Format . . . . . . DD-MM-YY
Date Property Text . . . Date
Time Property Text . . . Time
Modified Property Text . Modified
Compare Executable . . C:\Program Files\Beyond Compare ...
Compare Directory . . .
Default . . . . C:\Users\Reijer\AppData\Local\Temp\

- **Compare Executable**
  Here you can set the executable that you use for comparing the C/AL code. For example Beyond Compare or UltraCompare.

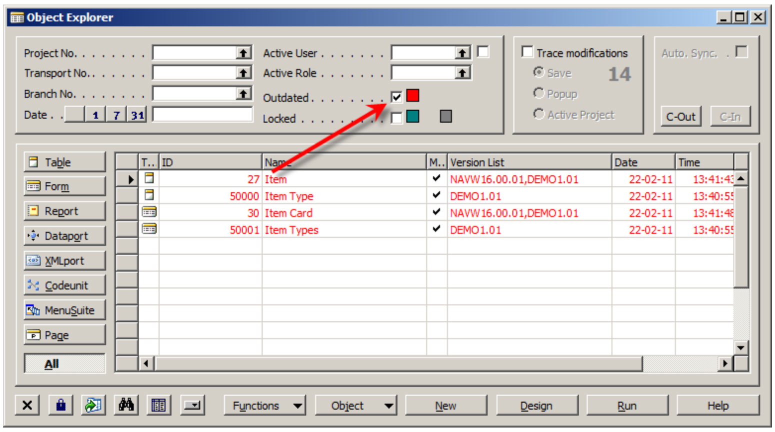
  %1 will be replaced with the left file/directory
  %2 will be replaced with the right file/directory

- **Compare Directory**
  Here you set the directory that the Object Manager uses to save the C/AL code before opening the compare tool. If you leave it empty your default temp directory will be used.

NOTE: The Object Manager creates folders and files in your “Compare Directory”. Periodically you have to clean this folder.

9.2. Update C/AL History

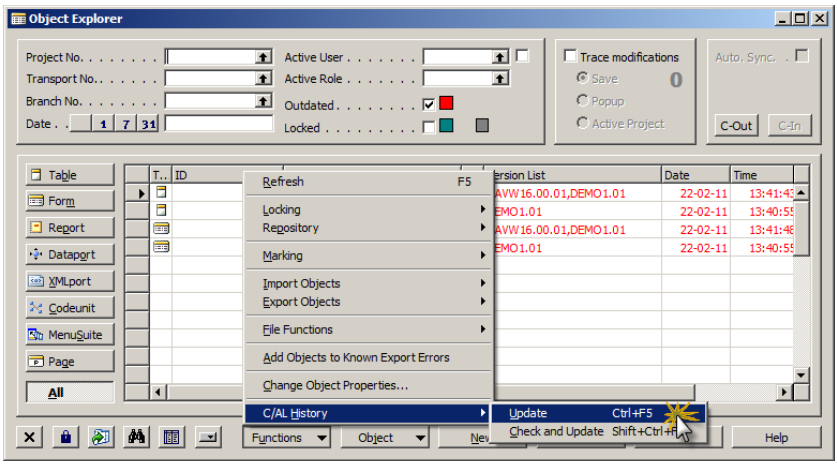
Open the Object Explorer and select outdated objects. An object is outdated when the C/AL History is not up to date.



Outdated ☑
Locked ☐
14

T..	ID	Name	M..	Version List	Date	Time
	27	Item	✓	NAVW16.00.01,DEMO1.01	22-02-11	13:41:4
	50000	Item Type	✓	DEMO1.01	22-02-11	13:40:5
	30	Item Card	✓	NAVW16.00.01,DEMO1.01	22-02-11	13:41:4
	50001	Item Types	✓	DEMO1.01	22-02-11	13:40:5

Press “C/AL History” > Update.



Outdated ☑
Locked ☐
0

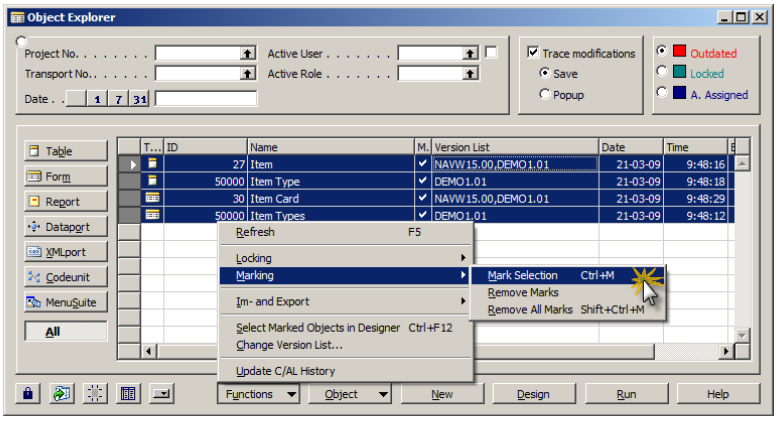
9.3. Check and Update C/AL History

The Object Manager only checks if the properties of an object are changed. If you e.g. imported objects with a text file and the date and time of the objects are still the same but the C/AL of the objects is changed the Object Manager will not update the C/AL History. If you know that there are such changes in your database you will have to use the “Check and Update” option. Now the Object Manager will compare the contents of the objects with the C/AL History. This is a time consuming process.

9.4. Update C/AL History with Text File

When you do not have installed the DLL (for more information see section 2.1- Installing) you can also manually update the “C/AL History” by selecting the outdated objects in the Object Explorer and export the objects in text format and import the objects in the Object Explorer.

1. Select outdated objects.
2. Press “Mark Selection”.

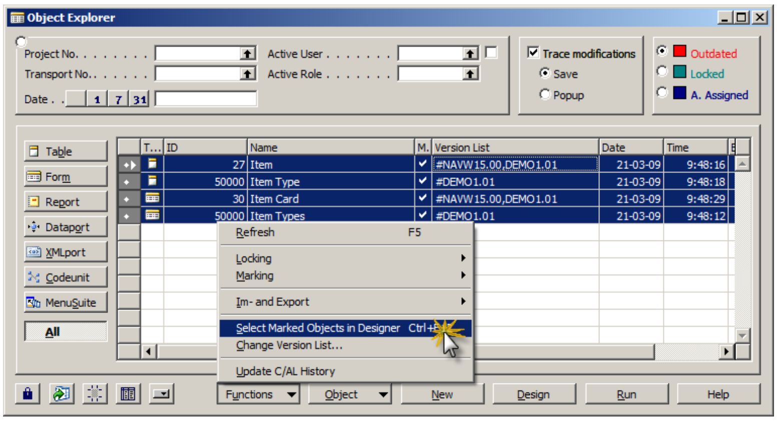


☑ Trace modifications ● Save ○ Popup

● Outdated ○ Locked ○ A. Assigned

T...	ID	Name	M.	Version List	Date	Time
	27	Item	✓	NAVW15.00,DEMO1.01	21-03-09	9:48:16
	50000	Item Type	✓	DEMO1.01	21-03-09	9:48:18
	30	Item Card	✓	NAVW15.00,DEMO1.01	21-03-09	9:48:29
	50000	Item Types	✓	DEMO1.01	21-03-09	9:48:12

3. Press “Select Marked Objects in Designer”.

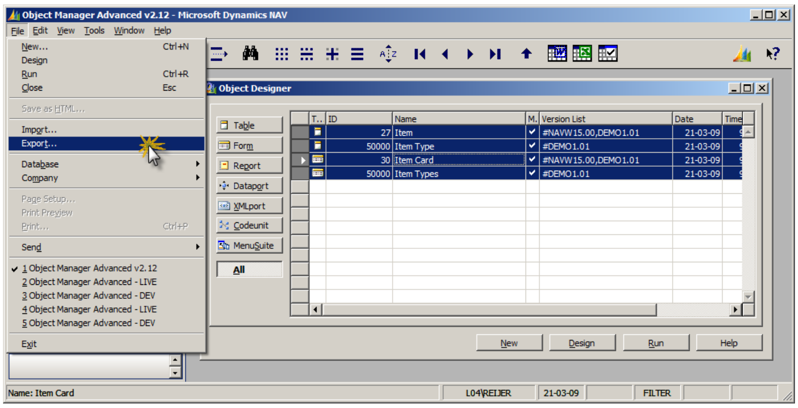


☑ Trace modifications ● Save ○ Popup

● Outdated ○ Locked ○ A. Assigned

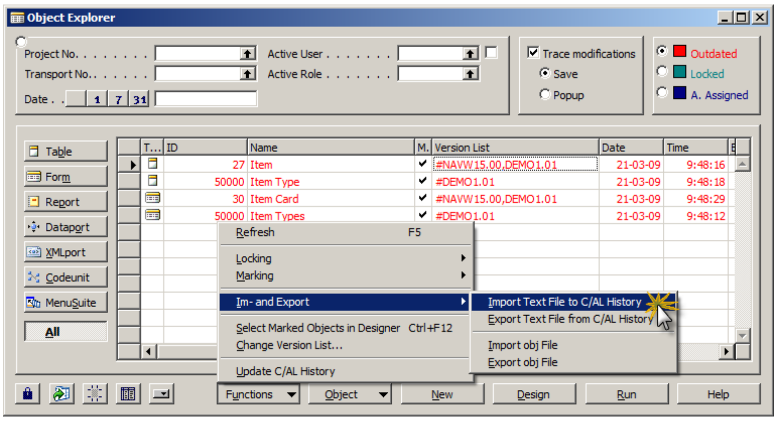
T...	ID	Name	M.	Version List	Date	Time
	27	Item	✓	#NAVW15.00,DEMO1.01	21-03-09	9:48:16
	50000	Item Type	✓	#DEMO1.01	21-03-09	9:48:18
	30	Item Card	✓	#NAVW15.00,DEMO1.01	21-03-09	9:48:29
	50000	Item Types	✓	#DEMO1.01	21-03-09	9:48:12

4. Export the objects in text format.



T..	ID	Name	M.	Version List	Date	Time
	27	Item	✓	#NAVW15.00,DEMO1.01	21-03-09	
	50000	Item Type	✓	#DEMO1.01	21-03-09	
	30	Item Card	✓	#NAVW15.00,DEMO1.01	21-03-09	
	50000	Item Types	✓	#DEMO1.01	21-03-09	

5. Import the text file in the Object Explorer.



T...	ID	Name	M.	Version List	Date	Time
	27	Item	✓	#NAVW15.00,DEMO1.01	21-03-09	9:48:16
	50000	Item Type	✓	#DEMO1.01	21-03-09	9:48:18
	30	Item Card	✓	#NAVW15.00,DEMO1.01	21-03-09	9:48:29
	50000	Item Types	✓	#DEMO1.01	21-03-09	9:48:12

NOTE: When you update the C/AL History with a text file the compiled version of the objects is not saved. Therefore it can be difficult to do a rollback.

9.5. Automatic C/AL Saving

When you perform some actions in the Object Manager it is possible to save the current C/AL to the "C/AL History".

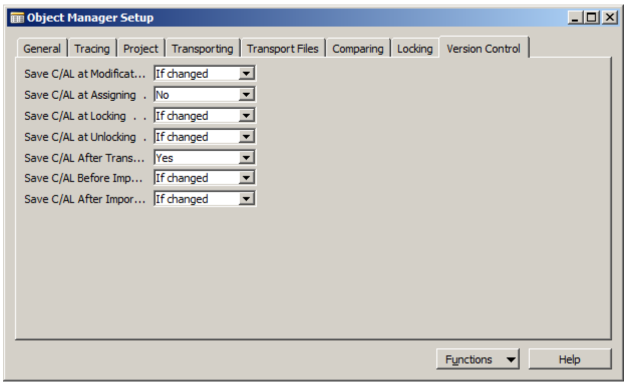
Go to Setup > Source Control and set the triggers:

- Save C/AL at Modification
- Save C/AL at Assigning
- Save C/AL at Locking
- Save C/AL at Unlocking
- Save C/AL at After Transporting
- Save C/AL at Before Importing
- Save C/AL at After Importing

You can set the 7 different triggers on 3 values: "No", "If changed" and "Yes".

- Yes: C/AL will be saved always when trigger action is executed.

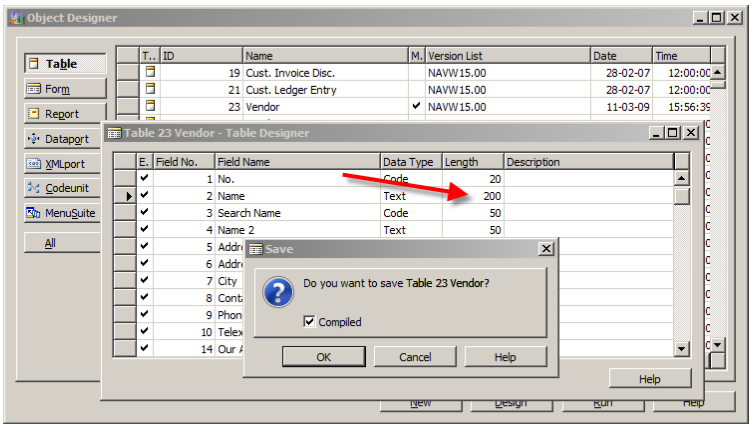
- If changed: C/AL will only be saved when the date or version list of the object has changed.
- No: C/AL will not be saved.



Save C/AL at Modificat...	If changed
Save C/AL at Assigning .	No
Save C/AL at Locking . .	If changed
Save C/AL at Unlocking .	If changed
Save C/AL After Trans...	Yes
Save C/AL Before Imp...	If changed
Save C/AL After Impor...	If changed

Example
If you modify an object with the trigger “Save C/AL at Modification” on “Yes” or “If changed”, you can view the “C/AL History” with the Object Explorer.

In this example we change the length of the field Name from 30 to 200. Save and compile the Vendor table.



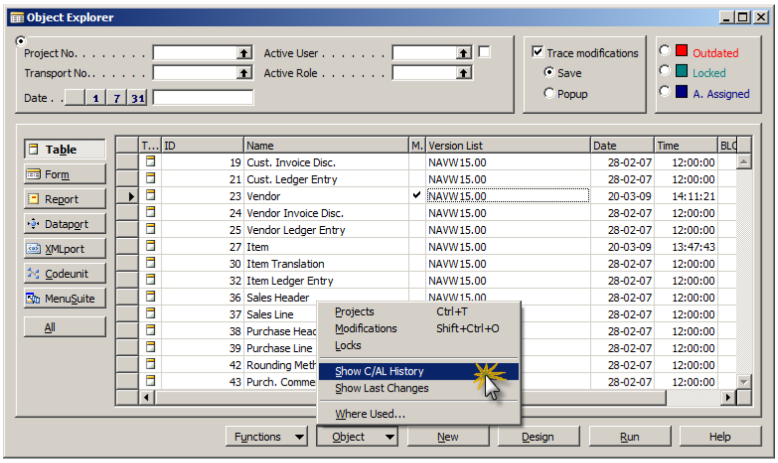
T..	ID	Name	M.	Version List	Date	Time
	19	Cust. Invoice Disc.		NAVW15.00	28-02-07	12:00:00
	21	Cust. Ledger Entry		NAVW15.00	28-02-07	12:00:00
	23	Vendor	✓	NAVW15.00	11-03-09	15:56:39

E.	Field No.	Field Name	Data Type	Length	Description
✓	1	No.	Code	20	
✓	2	Name	Text	200	
✓	3	Search Name	Code	50	
✓	4	Name 2	Text	50	
✓	5	Addr			
✓	6	Addr			
✓	7	City			
✓	8	Cont			
✓	9	Phon			
✓	10	Telex			
✓	14	Our			

Do you want to save Table 23 Vendor?

☑ Compiled

Select the Vendor table and “Show C/AL History” in the Object Explorer. The “C/AL History Objects” form opens.

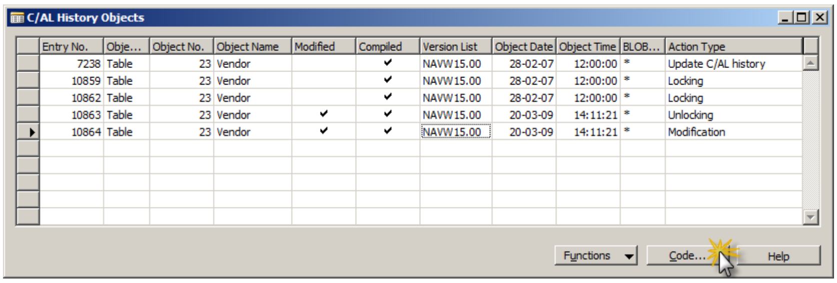


Project No. . . . .
Transport No.. . . . . .
Date . .
Active User . . . . ☐
Active Role . . . .

☑ Trace modifications
● Save
○ Popup

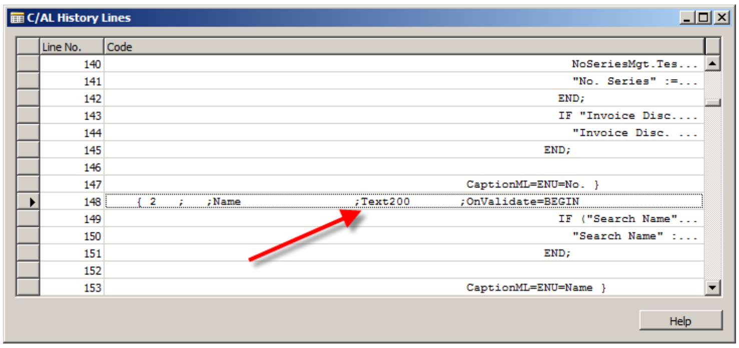
T...	ID	Name	M.	Version List	Date	Time	BLC
	19	Cust. Invoice Disc.		NAVW15.00	28-02-07	12:00:00	
	21	Cust. Ledger Entry		NAVW15.00	28-02-07	12:00:00	
	23	Vendor	✓	NAVW15.00	20-03-09	14:11:21	
	24	Vendor Invoice Disc.		NAVW15.00	28-02-07	12:00:00	
	25	Vendor Ledger Entry		NAVW15.00	28-02-07	12:00:00	
	27	Item		NAVW15.00	20-03-09	13:47:43	
	30	Item Translation		NAVW15.00	28-02-07	12:00:00	
	32	Item Ledger Entry		NAVW15.00	28-02-07	12:00:00	
	36	Sales Header		NAVW15.00	28-02-07	12:00:00	
	37	Sales Line			28-02-07	12:00:00	
	38	Purchase Heac			28-02-07	12:00:00	
	39	Purchase Line			28-02-07	12:00:00	
	42	Rounding Meth			28-02-07	12:00:00	
	43	Purch. Comme			28-02-07	12:00:00	

In this list you can see al changes in the object from the moment you started tracing (see section 3.2 - Trace Modifications). In our example you see the first line is the "original" object, the last line is your last change. The field "Action Type" shows which trigger saved the history. With the button Code you can see the code.



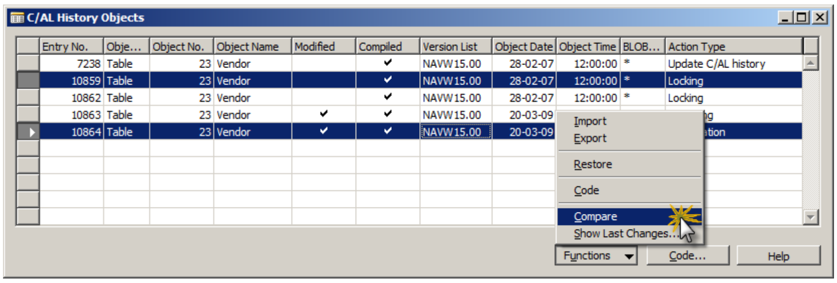
Entry No.	Obje...	Object No.	Object Name	Modified	Compiled	Version List	Object Date	Object Time	BLOB...	Action Type
7238	Table	23	Vendor		✓	NAVW15.00	28-02-07	12:00:00	*	Update C/AL history
10859	Table	23	Vendor		✓	NAVW15.00	28-02-07	12:00:00	*	Locking
10862	Table	23	Vendor		✓	NAVW15.00	28-02-07	12:00:00	*	Locking
10863	Table	23	Vendor	✓	✓	NAVW15.00	20-03-09	14:11:21	*	Unlocking
10864	Table	23	Vendor	✓	✓	NAVW15.00	20-03-09	14:11:21	*	Modification

C/AL History Lines.

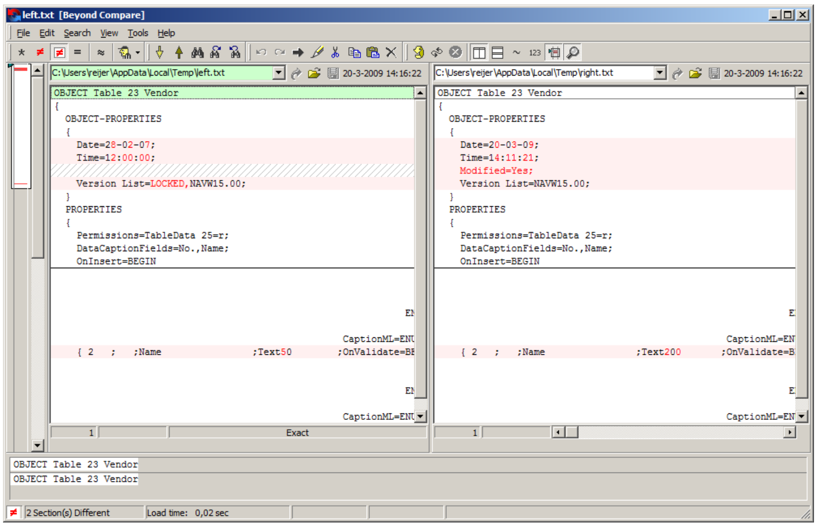


Line No.	Code
140	NoSeriesMgt.Tes...
141	"No. Series" :=...
142	END;
143	IF "Invoice Disc....
144	"Invoice Disc. ...
145	END;
146	
147	CaptionML=ENU=No. }
148	{ 2 ; ;Name ;Text200 ;OnValidate=BEGIN
149	IF ("Search Name"...
150	"Search Name" :...
151	END;
152	
153	CaptionML=ENU=Name }

To analyse the changes you can use the function Compare.



Entry No.	Obje…	Object No.	Object Name	Modified	Compiled	Version List	Object Date	Object Time	BLOB…	Action Type
7238	Table	23	Vendor		✓	NAVW15.00	28-02-07	12:00:00	*	Update C/AL history
10859	Table	23	Vendor		✓	NAVW15.00	28-02-07	12:00:00	*	Locking
10862	Table	23	Vendor		✓	NAVW15.00	28-02-07	12:00:00	*	Locking
10863	Table	23	Vendor	✓	✓	NAVW15.00	20-03-09			g
10864	Table	23	Vendor	✓	✓	NAVW15.00	20-03-09			tion



C:\Users\reijer\AppData\Local\Temp\left.txt 20-3-2009 14:16:22

```
OBJECT Table 23 Vendor
{
  OBJECT-PROPERTIES
  {
    Date=28-02-07;
    Time=12:00:00;

    Version List=LOCKED,NAVW15.00;
  }
  PROPERTIES
  {
    Permissions=TableData 25=r;
    DataCaptionFields=No.,Name;
    OnInsert=BEGIN
```

C:\Users\reijer\AppData\Local\Temp\right.txt 20-3-2009 14:16:22

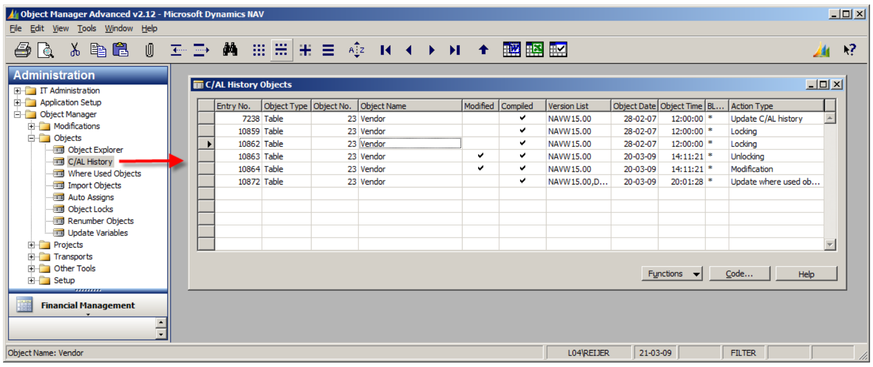
```
OBJECT Table 23 Vendor
{
  OBJECT-PROPERTIES
  {
    Date=20-03-09;
    Time=14:11:21;
    Modified=Yes;
    Version List=NAVW15.00;
  }
  PROPERTIES
  {
    Permissions=TableData 25=r;
    DataCaptionFields=No.,Name;
    OnInsert=BEGIN
```

```
{ 2   ;   ;Name                    ;Text50         ;OnValidate=BE
```

```
{ 2   ;   ;Name                    ;Text200        ;OnValidate=B
```

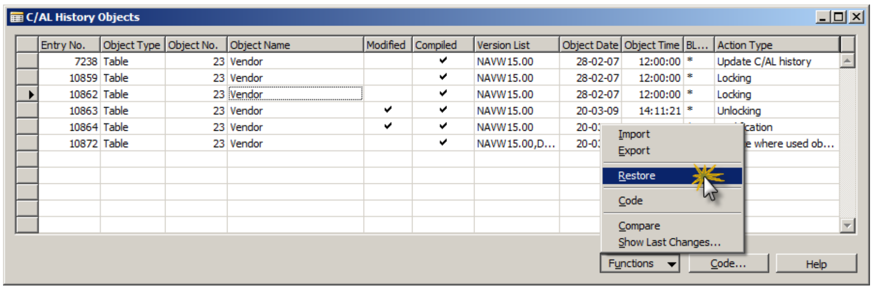
9.6. Restore an Object

In the “C/AL History Objects” form you can restore an object to a previous version.



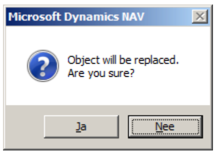
Entry No.	Object Type	Object No.	Object Name	Modified	Compiled	Version List	Object Date	Object Time	BL…	Action Type
7238	Table	23	Vendor		✓	NAVW15.00	28-02-07	12:00:00	*	Update C/AL history
10859	Table	23	Vendor		✓	NAVW15.00	28-02-07	12:00:00	*	Locking
10862	Table	23	Vendor		✓	NAVW15.00	28-02-07	12:00:00	*	Locking
10863	Table	23	Vendor	✓	✓	NAVW15.00	20-03-09	14:11:21	*	Unlocking
10864	Table	23	Vendor	✓	✓	NAVW15.00	20-03-09	14:11:21	*	Modification
10872	Table	23	Vendor		✓	NAVW15.00,D…	20-03-09	20:01:28	*	Update where used ob…

Press Restore.



C/AL History Objects

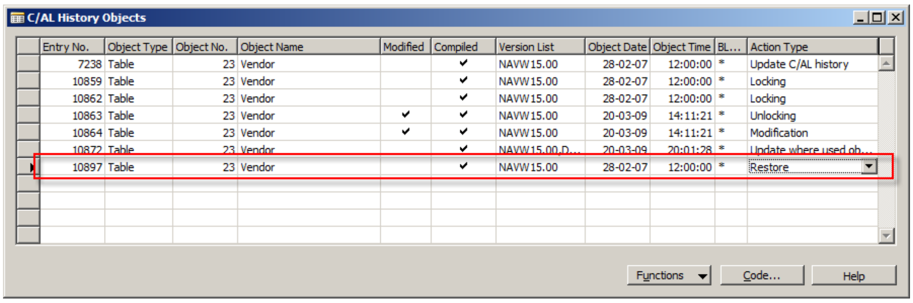
Entry No.	Object Type	Object No.	Object Name	Modified	Compiled	Version List	Object Date	Object Time	BL…	Action Type
7238	Table	23	Vendor		✓	NAVW15.00	28-02-07	12:00:00	*	Update C/AL history
10859	Table	23	Vendor		✓	NAVW15.00	28-02-07	12:00:00	*	Locking
10862	Table	23	Vendor		✓	NAVW15.00	28-02-07	12:00:00	*	Locking
10863	Table	23	Vendor	✓	✓	NAVW15.00	20-03-09	14:11:21	*	Unlocking
10864	Table	23	Vendor	✓	✓	NAVW15.00	20-0			cation
10872	Table	23	Vendor		✓	NAVW15.00,D…	20-0			e where used ob…



Microsoft Dynamics NAV

Object will be replaced.
Are you sure?

A new line of “Action Type” Restore is added to the “C/AL History” table.

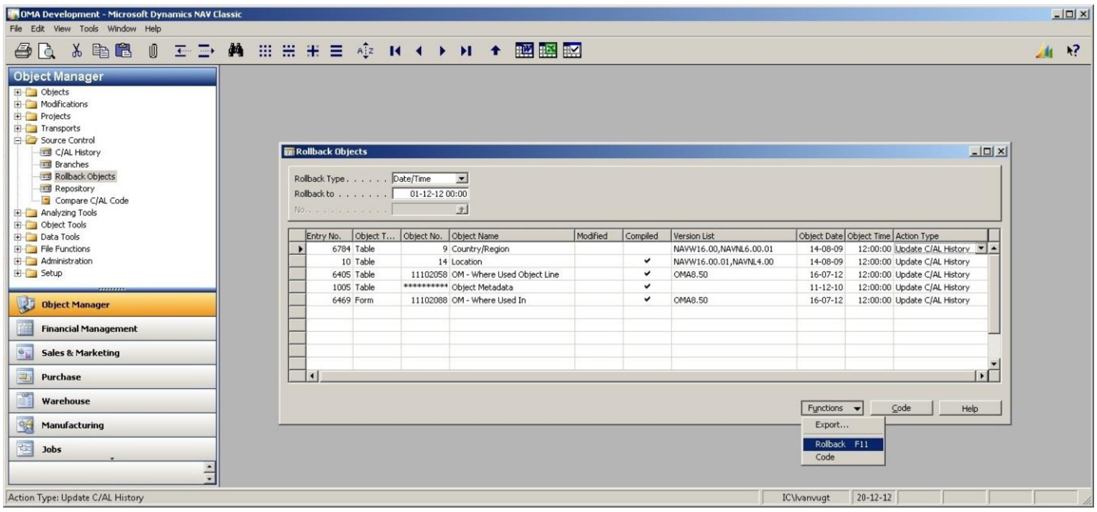


C/AL History Objects

Entry No.	Object Type	Object No.	Object Name	Modified	Compiled	Version List	Object Date	Object Time	BL…	Action Type
7238	Table	23	Vendor		✓	NAVW15.00	28-02-07	12:00:00	*	Update C/AL history
10859	Table	23	Vendor		✓	NAVW15.00	28-02-07	12:00:00	*	Locking
10862	Table	23	Vendor		✓	NAVW15.00	28-02-07	12:00:00	*	Locking
10863	Table	23	Vendor	✓	✓	NAVW15.00	20-03-09	14:11:21	*	Unlocking
10864	Table	23	Vendor	✓	✓	NAVW15.00	20-03-09	14:11:21	*	Modification
10872	Table	23	Vendor		✓	NAVW15.00,D…	20-03-09	20:01:28	*	Update where used ob…
10897	Table	23	Vendor		✓	NAVW15.00	28-02-07	12:00:00	*	Restore

9.7. Rollback Objects

When you want to rollback object changes you open the “Rollback Objects” window. Using the “Rollback Type” you can rollback to a certain date/time combination, for a specific project or specific transport.



OMA Development - Microsoft Dynamics NAV Classic

Rollback Objects

Rollback Type . . . . . . Date/Time
Rollback to 01-12-12 00:00
No. . . . .

Entry No.	Object T…	Object No.	Object Name	Modified	Compiled	Version List	Object Date	Object Time	Action Type
6784	Table	9	Country/Region			NAVW16.00,NAVNL6.00.01	14-08-09	12:00:00	Update C/AL History
10	Table	14	Location		✓	NAVW16.00.01,NAVNL4.00	14-08-09	12:00:00	Update C/AL History
6405	Table	11102058	OM - Where Used Object Line		✓	OMA8.50	16-07-12	12:00:00	Update C/AL History
1005	Table	**********	Object Metadata		✓		11-12-10	12:00:00	Update C/AL History
6469	Form	11102088	OM - Where Used In		✓	OMA8.50	16-07-12	12:00:00	Update C/AL History

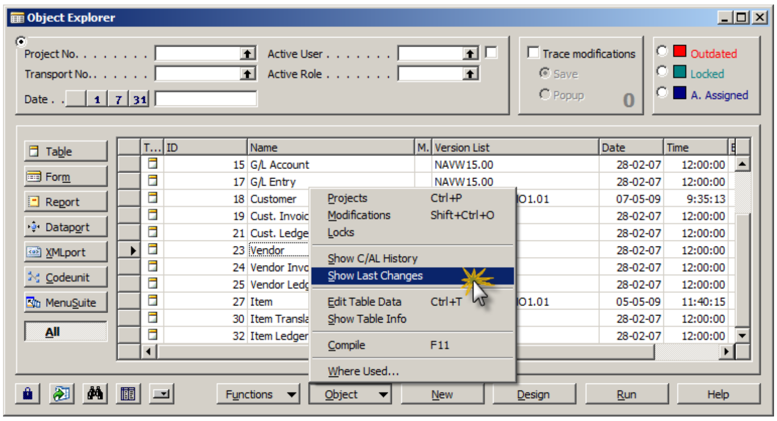
Action Type: Update C/AL History

- Date/Time
  Enter a date/time combination in the “Rollback to” field
- Project or Transport
  Enter a project/transport number in the “No.” field

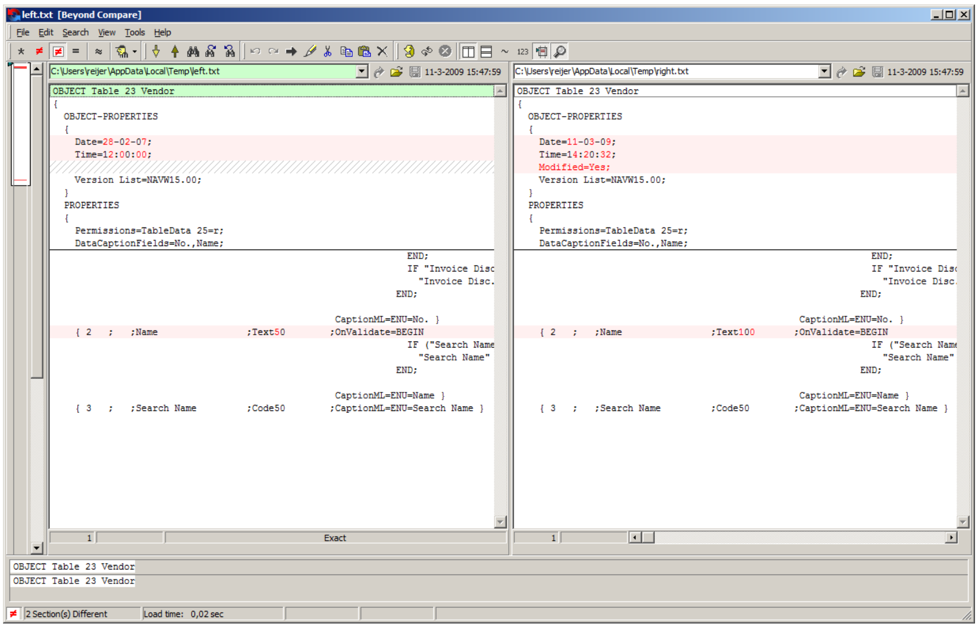
Object Manager will retrieve all the objects that were changed since that date or for that specific project/transport.
Press Functions – Rollback (or F11) to execute the rollback.

9.8. Compare two Versions

“Show Last Changes” will open your compare tool and show the changes you made compared with the previous version of the object.



T...	ID	Name	M.	Version List	Date	Time
	15	G/L Account		NAVW15.00	28-02-07	12:00:00
	17	G/L Entry		NAVW15.00	28-02-07	12:00:00
	18	Customer		O1.01	07-05-09	9:35:13
	19	Cust. Invoic			28-02-07	12:00:00
	21	Cust. Ledge			28-02-07	12:00:00
	23	Vendor			28-02-07	12:00:00
	24	Vendor Invo			28-02-07	12:00:00
	25	Vendor Ledg			28-02-07	12:00:00
	27	Item		O1.01	05-05-09	11:40:15
	30	Item Transla			28-02-07	12:00:00
	32	Item Ledger			28-02-07	12:00:00



C:\Users\reijer\AppData\Local\Temp\left.txt 11-3-2009 15:47:59

```
OBJECT Table 23 Vendor
{
  OBJECT-PROPERTIES
  {
    Date=28-02-07;
    Time=12:00:00;
    Version List=NAVW15.00;
  }
  PROPERTIES
  {
    Permissions=TableData 25=r;
    DataCaptionFields=No.,Name;
                                                      END;
                                                      IF "Invoice Disc
                                                        "Invoice Disc.
                                                    END;

                                            CaptionML=ENU=No. }
    { 2   ;   ;Name                ;Text50       ;OnValidate=BEGIN
                                                      IF ("Search Name
                                                        "Search Name"
                                                    END;

                                            CaptionML=ENU=Name }
    { 3   ;   ;Search Name         ;Code50       ;CaptionML=ENU=Search Name }
```

C:\Users\reijer\AppData\Local\Temp\right.txt 11-3-2009 15:47:59

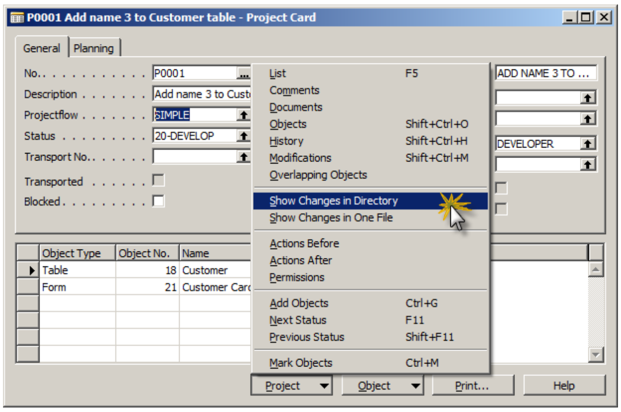
```
OBJECT Table 23 Vendor
{
  OBJECT-PROPERTIES
  {
    Date=11-03-09;
    Time=14:20:32;
    Modified=Yes;
    Version List=NAVW15.00;
  }
  PROPERTIES
  {
    Permissions=TableData 25=r;
    DataCaptionFields=No.,Name;
                                                      END;
                                                      IF "Invoice Disc
                                                        "Invoice Disc.
                                                    END;

                                            CaptionML=ENU=No. }
    { 2   ;   ;Name                ;Text100      ;OnValidate=BEGIN
                                                      IF ("Search Name
                                                        "Search Name"
                                                    END;

                                            CaptionML=ENU=Name }
    { 3   ;   ;Search Name         ;Code50       ;CaptionML=ENU=Search Name }
```

```
OBJECT Table 23 Vendor
OBJECT Table 23 Vendor
```

2 Section(s) Different
Load time: 0,02 sec

9.9. Analyze Modifications Made in a Project

You can compare changes of a project by using the function “Show Changes in Directory”.



No. P0001

Description . . . . Add name 3 to Custi

Projectflow . . . . SIMPLE

Status 20-DEVELOP

Transport No. . . . . . .

Transported . . . . . . ☐

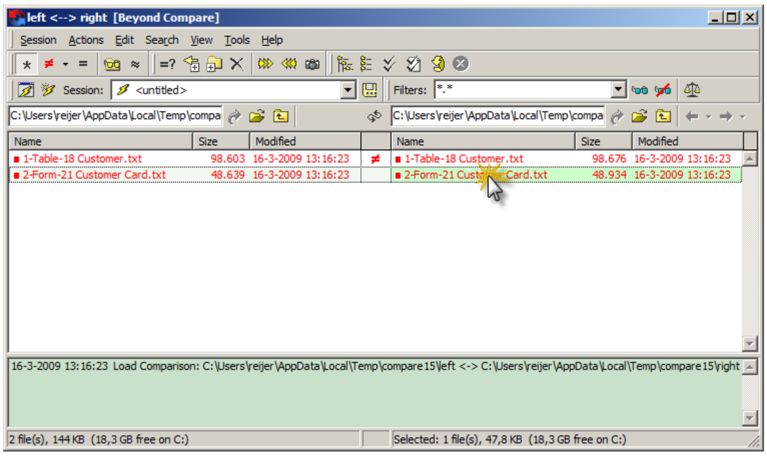
Blocked. . . . . ☐

ADD NAME 3 TO ...

DEVELOPER

Object Type	Object No.	Name
Table	18	Customer
Form	21	Customer Carc

This function opens the compare tool and shows you the differences in objects and code of the specific project.



C:\Users\reijer\AppData\Local\Temp\compa

C:\Users\reijer\AppData\Local\Temp\compa

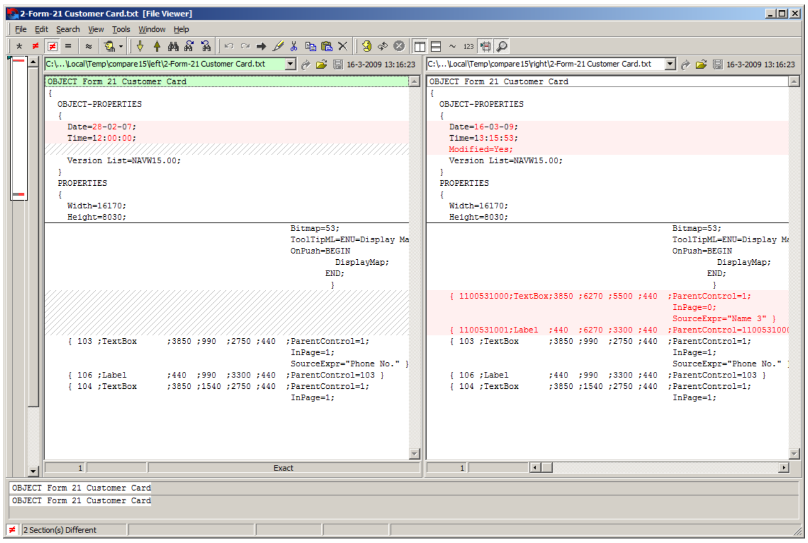
Name	Size	Modified		Name	Size	Modified
1-Table-18 Customer.txt	98.603	16-3-2009 13:16:23	≠	1-Table-18 Customer.txt	98.676	16-3-2009 13:16:23
2-Form-21 Customer Card.txt	48.639	16-3-2009 13:16:23		2-Form-21 Customer Card.txt	48.934	16-3-2009 13:16:23

```
16-3-2009 13:16:23 Load Comparison: C:\Users\reijer\AppData\Local\Temp\compare15\left <-> C:\Users\reijer\AppData\Local\Temp\compare15\right
```

2 file(s), 144 KB (18,3 GB free on C:)

Selected: 1 file(s), 47,8 KB (18,3 GB free on C:)

It is also possible to see the changes in one file. For example if you use UltraCompare which does not have the option of comparing directories.


2-Form-21 Customer Card.txt [File Viewer]
File Edit Search View Tools Window Help
C:\...\Local\Temp\compare15\left\2-Form-21 Customer Card.txt
16-3-2009 13:16:23
C:\...\Local\Temp\compare15\right\2-Form-21 Customer Card.txt
16-3-2009 13:16:23
1
Exact
1
2 Section(s) Different

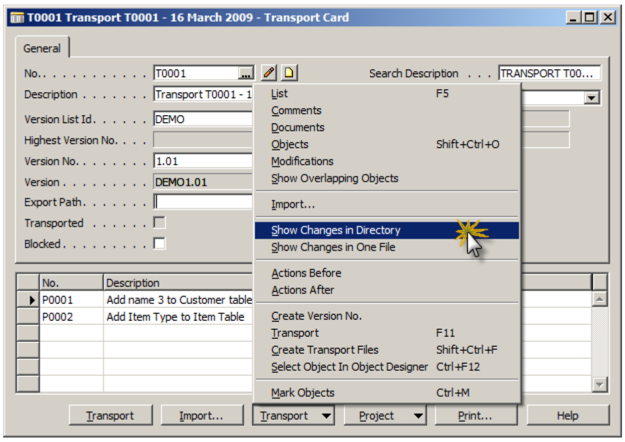
```
OBJECT Form 21 Customer Card
{
  OBJECT-PROPERTIES
  {
    Date=28-02-07;
    Time=12:00:00;

    Version List=NAVW15.00;
  }
  PROPERTIES
  {
    Width=16170;
    Height=8030;
                                                 Bitmap=53;
                                                 ToolTipML=ENU=Display Ma
                                                 OnPush=BEGIN
                                                          DisplayMap;
                                                       END;
                                                  }



    { 103 ;TextBox      ;3850 ;990  ;2750 ;440  ;ParentControl=1;
                                                 InPage=1;
                                                 SourceExpr="Phone No." }
    { 106 ;Label        ;440  ;990  ;3300 ;440  ;ParentControl=103 }
    { 104 ;TextBox      ;3850 ;1540 ;2750 ;440  ;ParentControl=1;
                                                 InPage=1;
```

```
OBJECT Form 21 Customer Card
{
  OBJECT-PROPERTIES
  {
    Date=16-03-09;
    Time=13:15:53;
    Modified=Yes;
    Version List=NAVW15.00;
  }
  PROPERTIES
  {
    Width=16170;
    Height=8030;
                                                 Bitmap=53;
                                                 ToolTipML=ENU=Display Ma
                                                 OnPush=BEGIN
                                                          DisplayMap;
                                                       END;
                                                  }
    { 1100531000;TextBox;3850 ;6270 ;5500 ;440  ;ParentControl=1;
                                                 InPage=0;
                                                 SourceExpr="Name 3" }
    { 1100531001;Label  ;440  ;6270 ;3300 ;440  ;ParentControl=1100531000
    { 103 ;TextBox      ;3850 ;990  ;2750 ;440  ;ParentControl=1;
                                                 InPage=1;
                                                 SourceExpr="Phone No." }
    { 106 ;Label        ;440  ;990  ;3300 ;440  ;ParentControl=103 }
    { 104 ;TextBox      ;3850 ;1540 ;2750 ;440  ;ParentControl=1;
                                                 InPage=1;
```

OBJECT Form 21 Customer Card
OBJECT Form 21 Customer Card

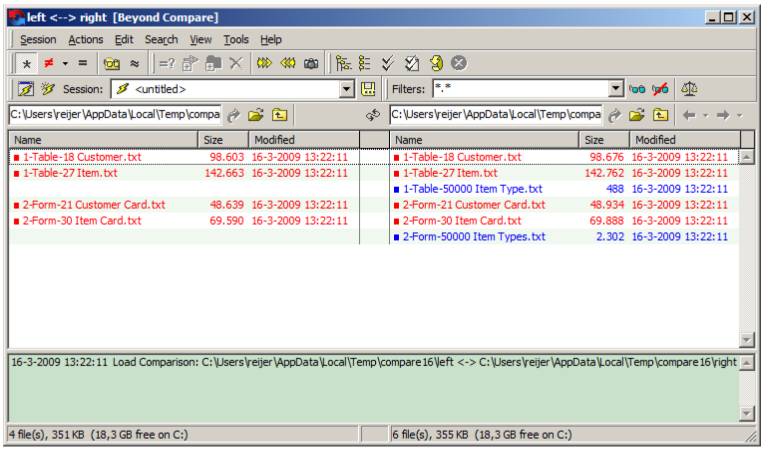
9.10. Analyze Modifications Made in a Transport

You can compare changes in a transport by using the function "Show Changes in Directory".



No. T0001
Description . . . . Transport T0001 - 1
Version List Id. . . . . . DEMO
Highest Version No. . . .
Version No. 1.01
Version DEMO1.01
Export Path . . . .
Transported . . . . . . ☐
Blocked . . . . ☐
Search Description . . . TRANSPORT T00...

No.	Description
P0001	Add name 3 to Customer table
P0002	Add Item Type to Item Table


left <--> right [Beyond Compare]
Session Actions Edit Search View Tools Help
Session: <untitled>
Filters: *.*
C:\Users\reijer\AppData\Local\Temp\compa
C:\Users\reijer\AppData\Local\Temp\compa

Name	Size	Modified	Name	Size	Modified
1-Table-18 Customer.txt	98.603	16-3-2009 13:22:11	1-Table-18 Customer.txt	98.676	16-3-2009 13:22:11
1-Table-27 Item.txt	142.663	16-3-2009 13:22:11	1-Table-27 Item.txt	142.762	16-3-2009 13:22:11
			1-Table-50000 Item Type.txt	488	16-3-2009 13:22:11
2-Form-21 Customer Card.txt	48.639	16-3-2009 13:22:11	2-Form-21 Customer Card.txt	48.934	16-3-2009 13:22:11
2-Form-30 Item Card.txt	69.590	16-3-2009 13:22:11	2-Form-30 Item Card.txt	69.888	16-3-2009 13:22:11
			2-Form-50000 Item Types.txt	2.302	16-3-2009 13:22:11

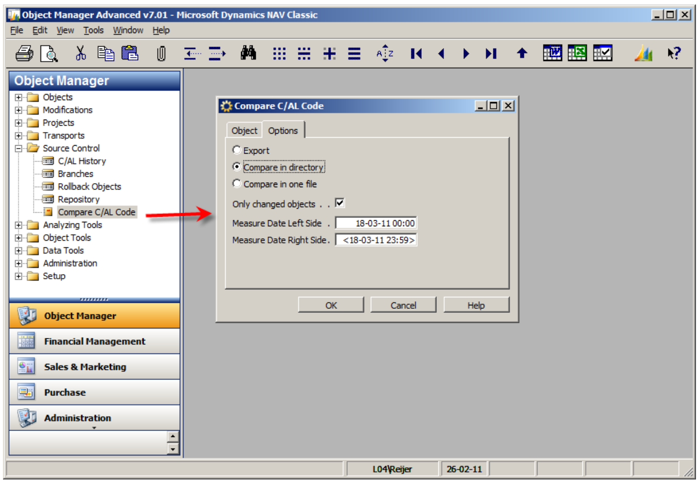
16-3-2009 13:22:11 Load Comparison: C:\Users\reijer\AppData\Local\Temp\compare16\left <-> C:\Users\reijer\AppData\Local\Temp\compare16\right

4 file(s), 351 KB (18,3 GB free on C:)
6 file(s), 355 KB (18,3 GB free on C:)

This function opens the compare tool and shows you the differences in objects and code of the specific transport.

9.11. Analyze Modifications Made in a Period of Time

With the report Compare C/AL Code you can analyze all the modifications that are done in a specific period of time.

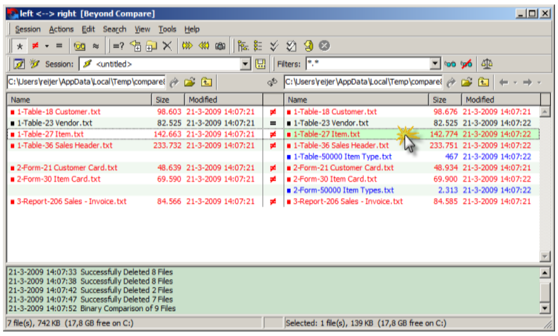


○ Export
● Compare in directory
○ Compare in one file

Only changed objects . . ☑

Measure Date Left Side . 18-03-11 00:00

Measure Date Right Side. <18-03-11 23:59>


left <--> right [Beyond Compare]
Session Actions Edit Search View Tools Help
Session: <untitled>
Filters: *.*

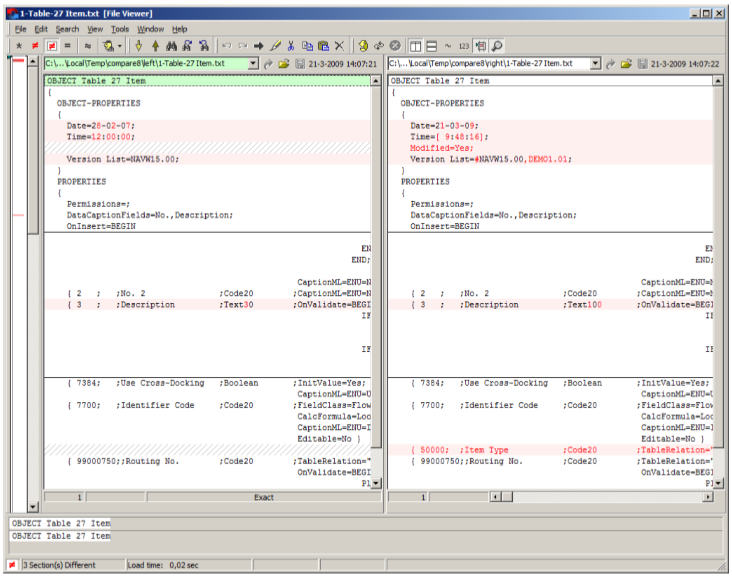
C:\Users\reijer\AppData\Local\Temp\compare

Name	Size	Modified		Name	Size	Modified
1-Table-18 Customer.txt	98.603	21-3-2009 14:07:21	≠	1-Table-18 Customer.txt	98.676	21-3-2009 14:07:21
1-Table-23 Vendor.txt	82.525	21-3-2009 14:07:21	=	1-Table-23 Vendor.txt	82.525	21-3-2009 14:07:22
1-Table-27 Item.txt	142.663	21-3-2009 14:07:21	≠	1-Table-27 Item.txt	142.774	21-3-2009 14:07:22
1-Table-36 Sales Header.txt	233.732	21-3-2009 14:07:21	≠	1-Table-36 Sales Header.txt	233.751	21-3-2009 14:07:22
				1-Table-50000 Item Type.txt	467	21-3-2009 14:07:22
2-Form-21 Customer Card.txt	48.639	21-3-2009 14:07:21	≠	2-Form-21 Customer Card.txt	48.934	21-3-2009 14:07:21
2-Form-30 Item Card.txt	69.590	21-3-2009 14:07:21	≠	2-Form-30 Item Card.txt	69.900	21-3-2009 14:07:22
				2-Form-50000 Item Types.txt	2.313	21-3-2009 14:07:22
3-Report-206 Sales - Invoice.txt	84.566	21-3-2009 14:07:21	≠	3-Report-206 Sales - Invoice.txt	84.585	21-3-2009 14:07:21

```
21-3-2009 14:07:33  Successfully Deleted 8 Files
21-3-2009 14:07:38  Successfully Deleted 8 Files
21-3-2009 14:07:42  Successfully Deleted 2 Files
21-3-2009 14:07:47  Successfully Deleted 7 Files
21-3-2009 14:07:52  Binary Comparison of 9 Files
```

7 file(s), 742 KB (17,8 GB free on C:)

Selected: 1 file(s), 139 KB (17,8 GB free on C:)


1-Table-27 Item.txt [File Viewer]
File Edit Search View Tools Window Help
C:\...\Local\Temp\compare8\left\1-Table-27 Item.txt
21-3-2009 14:07:21
C:\...\Local\Temp\compare8\right\1-Table-27 Item.txt
21-3-2009 14:07:22
Exact

```
OBJECT Table 27 Item
{
  OBJECT-PROPERTIES
  {
    Date=28-02-07;
    Time=12:00:00;

    Version List=NAVW15.00;
  }
  PROPERTIES
  {
    Permissions=;
    DataCaptionFields=No.,Description;
    OnInsert=BEGIN

                                                                   EN
                                                                 END;

                                                        CaptionML=ENU=N
    { 2   ;   ;No. 2               ;Code20        ;CaptionML=ENU=N
    { 3   ;   ;Description         ;Text30        ;OnValidate=BEGI
                                                                   IF

                                                                   IF

    { 7384;   ;Use Cross-Docking   ;Boolean       ;InitValue=Yes;
                                                   CaptionML=ENU=U
    { 7700;   ;Identifier Code     ;Code20        ;FieldClass=Flow
                                                   CalcFormula=Loc
                                                   CaptionML=ENU=I
                                                   Editable=No }

    { 99000750;;Routing No.        ;Code20        ;TableRelation="
                                                   OnValidate=BEGI
                                                                   Pl
```

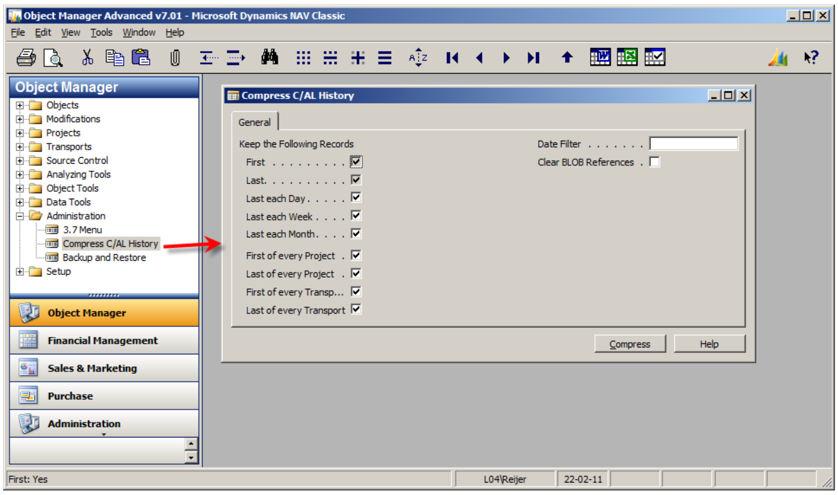
```
OBJECT Table 27 Item
{
  OBJECT-PROPERTIES
  {
    Date=21-03-09;
    Time=[ 9:48:16];
    Modified=Yes;
    Version List=#NAVW15.00,DEMO1.01;
  }
  PROPERTIES
  {
    Permissions=;
    DataCaptionFields=No.,Description;
    OnInsert=BEGIN

                                                                   EI
                                                                 END;

                                                        CaptionML=ENU=N
    { 2   ;   ;No. 2               ;Code20        ;CaptionML=ENU=N
    { 3   ;   ;Description         ;Text100       ;OnValidate=BEGI
                                                                   II

                                                                   II

    { 7384;   ;Use Cross-Docking   ;Boolean       ;InitValue=Yes;
                                                   CaptionML=ENU=U
    { 7700;   ;Identifier Code     ;Code20        ;FieldClass=Flow
                                                   CalcFormula=Loc
                                                   CaptionML=ENU=I
                                                   Editable=No }
    { 50000;  ;Item Type           ;Code20        ;TableRelation="
    { 99000750;;Routing No.        ;Code20        ;TableRelation="
                                                   OnValidate=BEGI
                                                                   Pl
```

```
OBJECT Table 27 Item
OBJECT Table 27 Item
```

3 Section(s) Different | Load time: 0,02 sec

9.12. Compress C/AL History

After a while the C/AL history can be very big. To compress these tables you can use the option “Compress C/AL History” in the Administration menu.



Keep the Following Records
- First ☑
- Last ☑
- Last each Day ☑
- Last each Week ☑
- Last each Month ☑
- First of every Project ☑
- Last of every Project ☑
- First of every Transp... ☑
- Last of every Transport ☑

Date Filter

Clear BLOB References ☐

First: Yes | L04\Reijer | 22-02-11

Check the options you want to keep and press Compress to delete all records that does not comply to one of the options. You can also enter a Date Filter to only compress the C/AL History that is older than e.g. a couple of months. If you choose the option “Clear BLOB References” then all compiled version of the objects in the history will be deleted. After this you cannot restore an object or do a rollback.

9.13. Merge C/AL History

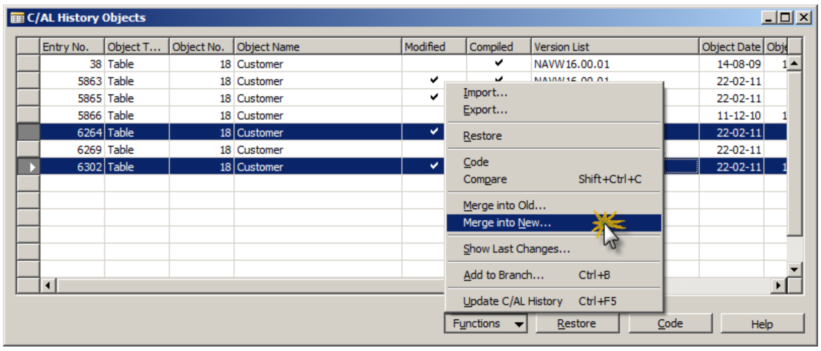
With this option it is possible to merge two C/AL History records in an external editor and save your changes automatically back into the database.

There are 2 options:

- **Merge into Old**
  The oldest C/AL History record will be imported in the database when you are done.

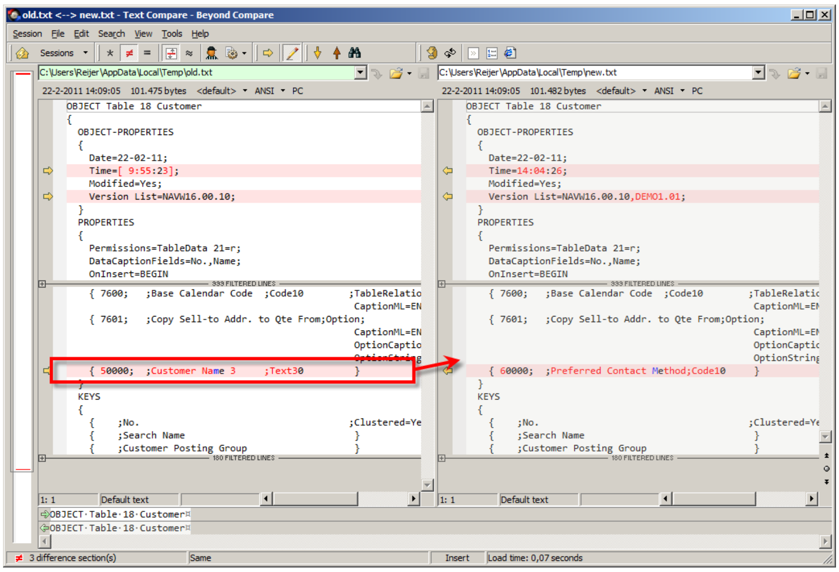
- **Merge into New**
  The newest C/AL History record will be imported in the database when you are done.

Select the two history records you want to merge and press Merge into New.



Entry No.	Object T...	Object No.	Object Name	Modified	Compiled	Version List	Object Date	Obje
38	Table	18	Customer		✓	NAVW16.00.01	14-08-09	1
5863	Table	18	Customer	✓	✓		22-02-11	
5865	Table	18	Customer	✓			22-02-11	
5866	Table	18	Customer				11-12-10	1
6264	Table	18	Customer	✓			22-02-11	
6269	Table	18	Customer				22-02-11	
6302	Table	18	Customer	✓			22-02-11	1

The merge tool will open where you can merge the differences. In our example the customer table overwritten by a colleague and your change was lost.


old.txt <--> new.txt - Text Compare - Beyond Compare
Session File Edit Search View Tools Help
Sessions
999 FILTERED LINES
180 FILTERED LINES
1: 1
Default text
Insert

C:\Users\Reijer\AppData\Local\Temp\old.txt
22-2-2011 14:09:05 101.475 bytes <default> ANSI PC

```
OBJECT Table 18 Customer
{
  OBJECT-PROPERTIES
  {
    Date=22-02-11;
    Time=[ 9:55:23];
    Modified=Yes;
    Version List=NAVW16.00.10;
  }
  PROPERTIES
  {
    Permissions=TableData 21=r;
    DataCaptionFields=No.,Name;
    OnInsert=BEGIN
    { 7600;   ;Base Calendar Code  ;Code10         ;TableRelatio
                                                    CaptionML=EN
    { 7601;   ;Copy Sell-to Addr. to Qte From;Option;
                                                    CaptionML=EN
                                                    OptionCaptio
                                                    OptionString
    { 50000;  ;Customer Name 3     ;Text30          }
  }
  KEYS
  {
    {    ;No.                                      ;Clustered=Ye
    {    ;Search Name                               }
    {    ;Customer Posting Group                    }
```

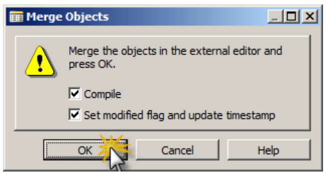
C:\Users\Reijer\AppData\Local\Temp\new.txt
22-2-2011 14:09:05 101.482 bytes <default> ANSI PC

```
OBJECT Table 18 Customer
{
  OBJECT-PROPERTIES
  {
    Date=22-02-11;
    Time=14:04:26;
    Modified=Yes;
    Version List=NAVW16.00.10,DEMO1.01;
  }
  PROPERTIES
  {
    Permissions=TableData 21=r;
    DataCaptionFields=No.,Name;
    OnInsert=BEGIN
    { 7600;   ;Base Calendar Code  ;Code10         ;TableRelatic
                                                    CaptionML=EN
    { 7601;   ;Copy Sell-to Addr. to Qte From;Option;
                                                    CaptionML=EN
                                                    OptionCaptic
                                                    OptionString
    { 60000;  ;Preferred Contact Method;Code10     }
  }
  KEYS
  {
    {    ;No.                                      ;Clustered=Ye
    {    ;Search Name                               }
    {    ;Customer Posting Group                    }
```

```
OBJECT·Table·18·Customer¤
OBJECT·Table·18·Customer¤
```

3 difference section(s) | Same | Insert | Load time: 0,07 seconds

We merge the change into the new version and close the merge tool. Now you see the dialog in NAV that is waiting at your confirm.



Merge Objects

Merge the objects in the external editor and press OK.

☑ Compile

☑ Set modified flag and update timestamp

Compile

Object will be compiled.

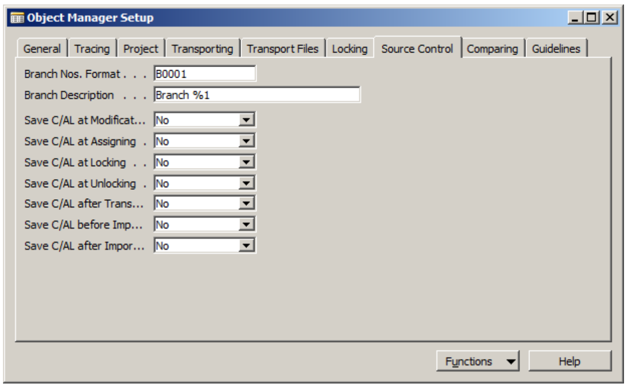
Set Modified Flag

The timestamp of the object will be changed to the current date and time and the modify flag will be set. If this option is disabled the date and time of the merged text file will be used.

10. Branches

A branch is what happens when your development team needs to work on two distinct copies of a set objects at the same time. With branches it is possible to reserve, group, develop and take a deeper look at selected versions of objects. It makes it possible to work on reserved versions of objects or even develop simultaneously on the same object.

10.1. Setup



Branch Nos. Format . . . B0001
Branch Description . . . Branch %1
Save C/AL at Modificat... No
Save C/AL at Assigning . No
Save C/AL at Locking . . No
Save C/AL at Unlocking . No
Save C/AL after Trans... No
Save C/AL before Imp... No
Save C/AL after Impor... No

- **Branch Nos. Format**

- **Branch Description**
  %1 will be replaced by the “Branch No.”.
  You can use date expressions like: <Day> <Month Text> <Year4>.

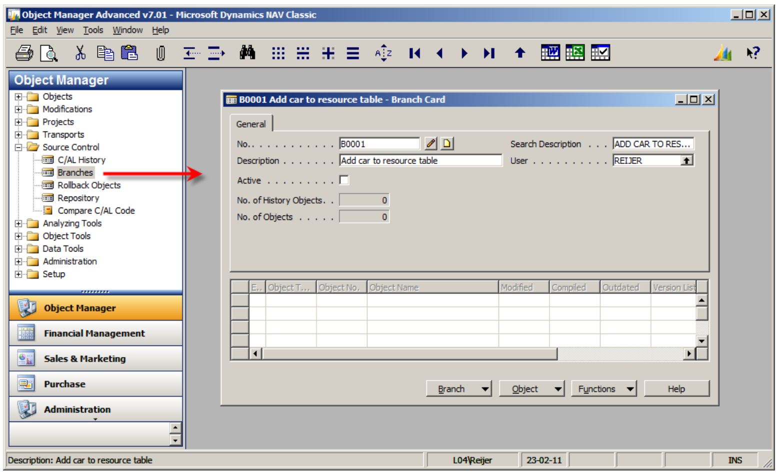
10.2. Comment Groups

Comment Groups are similar to Comment Groups in projects. For more information see section 5.3 - Comment Groups.

10.3. Documents

Documents in branches are similar to Documents in projects. For more information see section 5.4 - Documents.

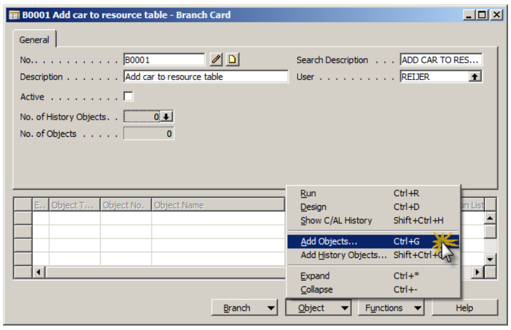
10.4. Add Objects to a Branch



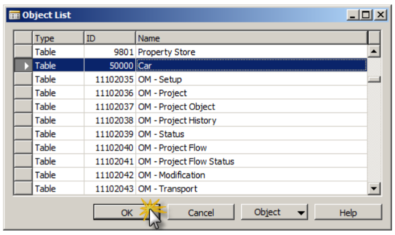
No. B0001
Description: Add car to resource table
Active ☐
No. of History Objects 0
No. of Objects 0
Search Description: ADD CAR TO RES...
User: REIJER

Description: Add car to resource table | L04\Reijer | 23-02-11 | INS

You can add different objects to a branch. The current version is retrieved.

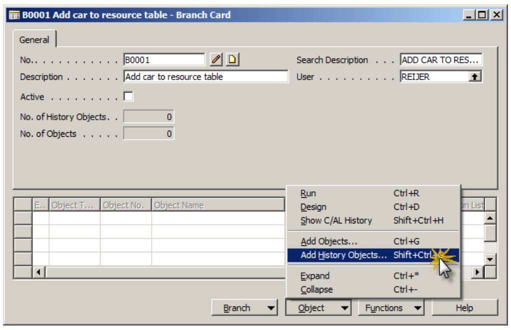

B0001 Add car to resource table - Branch Card
General
E.. Object T... Object No. Object Name
Run Ctrl+R
Design Ctrl+D
Show C/AL History Shift+Ctrl+H
Add Objects... Ctrl+G
Add History Objects... Shift+Ctrl+G
Expand Ctrl+*
Collapse Ctrl+-
Branch
Object
Functions
Help

No. B0001
Description: Add car to resource table
Active ☐
No. of History Objects 0
No. of Objects 0
Search Description: ADD CAR TO RES...
User: REIJER

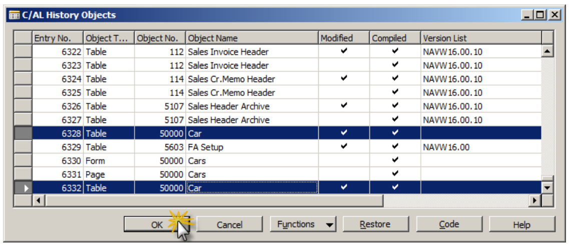


Type	ID	Name
Table	9801	Property Store
Table	50000	Car
Table	11102035	OM - Setup
Table	11102036	OM - Project
Table	11102037	OM - Project Object
Table	11102038	OM - Project History
Table	11102039	OM - Status
Table	11102040	OM - Project Flow
Table	11102041	OM - Project Flow Status
Table	11102042	OM - Modification
Table	11102043	OM - Transport

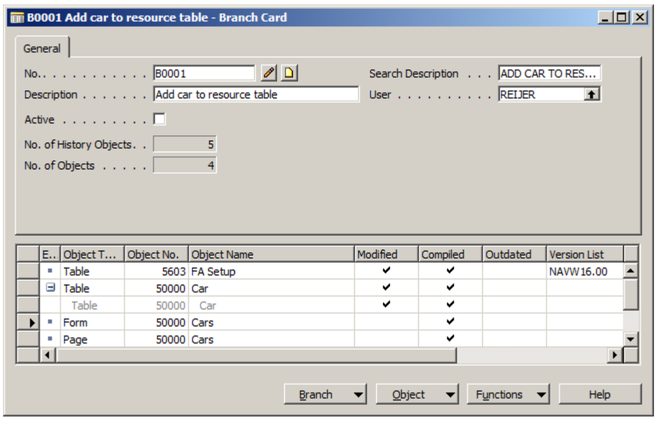
Or you can add a selection from the “C/AL History” table to your branch if needed.


B0001 Add car to resource table - Branch Card
General
E.. Object T... Object No. Object Name
Run Ctrl+R
Design Ctrl+D
Show C/AL History Shift+Ctrl+H
Add Objects... Ctrl+G
Add History Objects... Shift+Ctrl
Expand Ctrl+*
Collapse Ctrl+-
Branch
Object
Functions
Help

No. B0001
Description: Add car to resource table
Active ☐
No. of History Objects 0
No. of Objects 0
Search Description ADD CAR TO RES...
User REIJER



Entry No.	Object T...	Object No.	Object Name	Modified	Compiled	Version List
6322	Table	112	Sales Invoice Header	✓	✓	NAVW16.00.10
6323	Table	112	Sales Invoice Header		✓	NAVW16.00.10
6324	Table	114	Sales Cr.Memo Header	✓	✓	NAVW16.00.10
6325	Table	114	Sales Cr.Memo Header		✓	NAVW16.00.10
6326	Table	5107	Sales Header Archive	✓	✓	NAVW16.00.10
6327	Table	5107	Sales Header Archive		✓	NAVW16.00.10
6328	Table	50000	Car	✓	✓	
6329	Table	5603	FA Setup	✓	✓	NAVW16.00
6330	Form	50000	Cars		✓	
6331	Page	50000	Cars		✓	
6332	Table	50000	Car	✓	✓	



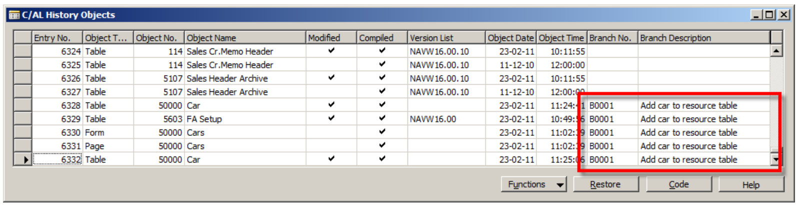
B0001 Add car to resource table - Branch Card

No.	B0001
Description	Add car to resource table
Active	☐
No. of History Objects	5
No. of Objects	4
Search Description	ADD CAR TO RES...
User	REIJER

E..	Object T...	Object No.	Object Name	Modified	Compiled	Outdated	Version List
▪	Table	5603	FA Setup	✓	✓		NAVW16.00
⊟	Table	50000	Car	✓	✓		
	Table	50000	Car	✓	✓		
▪	Form	50000	Cars		✓		
▪	Page	50000	Cars		✓		

In the example above you see two lines for the table car. The first one is the latest version. The second line is an older version. It is possible to expand and collapse an object to see only the latest version.

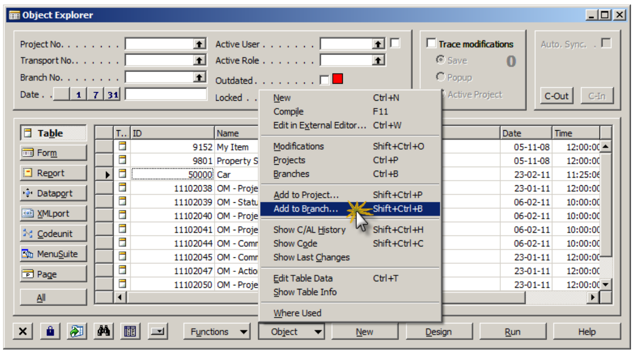
In the C/AL History table the branch is added to the selected versions.



C/AL History Objects

Entry No.	Object T...	Object No.	Object Name	Modified	Compiled	Version List	Object Date	Object Time	Branch No.	Branch Description
6324	Table	114	Sales Cr.Memo Header	✓	✓	NAVW16.00.10	23-02-11	10:11:55		
6325	Table	114	Sales Cr.Memo Header		✓	NAVW16.00.10	11-12-10	12:00:00		
6326	Table	5107	Sales Header Archive	✓	✓	NAVW16.00.10	23-02-11	10:11:55		
6327	Table	5107	Sales Header Archive		✓	NAVW16.00.10	11-12-10	12:00:00		
6328	Table	50000	Car	✓	✓		23-02-11	11:24:41	B0001	Add car to resource table
6329	Table	5603	FA Setup	✓	✓	NAVW16.00	23-02-11	10:49:56	B0001	Add car to resource table
6330	Form	50000	Cars		✓		23-02-11	11:02:[illegible]9	B0001	Add car to resource table
6331	Page	50000	Cars		✓		23-02-11	11:02:[illegible]9	B0001	Add car to resource table
6332	Table	50000	Car	✓	✓		23-02-11	11:25:06	B0001	Add car to resource table

You can add an object from the Object Explorer and add C/AL History Lines from the C/AL History to a Branch.



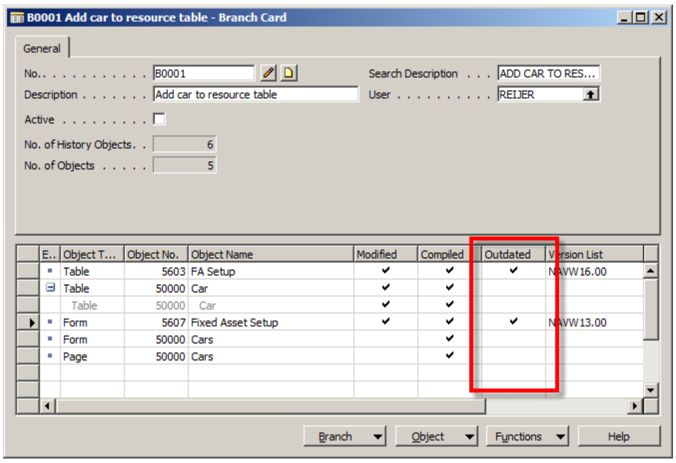
Object Explorer

Project No.	
Transport No.	
Branch No.	
Date	
Active User	☐
Active Role	
Outdated	☐
Locked	
Trace modifications	☐
Auto. Sync.	☐

T..	ID	Name	Date	Time
	9152	My Item	05-11-08	12:00:0
	9801	Property S	05-11-08	12:00:0
	50000	Car	23-02-11	11:25:06
	11102038	OM - Proje	23-01-11	12:00:0
	11102039	OM - Statu	06-02-11	10:00:0
	11102040	OM - Proje	06-02-11	10:00:0
	11102041	OM - Proje	06-02-11	10:00:0
	11102044	OM - Comr	06-02-11	10:00:0
	11102045	OM - Comr	23-01-11	12:00:0
	11102047	OM - Actio	23-01-11	12:00:0
	11102050	OM - Proje	23-01-11	12:00:0

10.5. Outdated Objects in a Branch

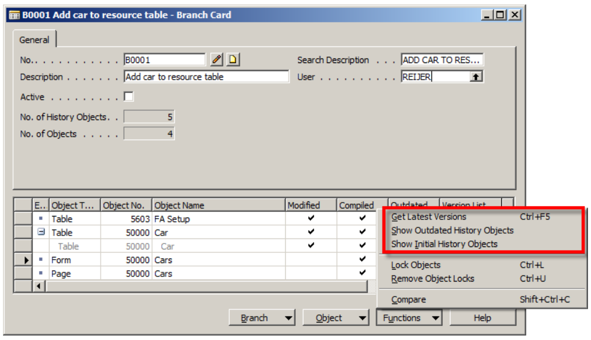
When an object in your branch in not the current version they have a Boolean in the outdated field.



No..	B0001	Search Description	ADD CAR TO RES...
Description	Add car to resource table	User	REIJER
Active	☐		
No. of History Objects	6		
No. of Objects	5		

E..	Object T...	Object No.	Object Name	Modified	Compiled	Outdated	Version List
▪	Table	5603	FA Setup	✓	✓	✓	NAVW16.00
⊟	Table	50000	Car	✓	✓		
	Table	50000	Car	✓	✓		
▪	Form	5607	Fixed Asset Setup	✓	✓	✓	NAVW13.00
▪	Form	50000	Cars		✓		
▪	Page	50000	Cars		✓		

There are three possibilities to cope with this conflicts.



B0001 Add car to resource table - Branch Card
General
Get Latest Versions Ctrl+F5
Show Outdated History Objects
Show Initial History Objects
Lock Objects Ctrl+L
Remove Object Locks Ctrl+U
Compare Shift+Ctrl+C
Branch
Object
Functions
Help

No..	B0001	Search Description	ADD CAR TO RES...
Description	Add car to resource table	User	REIJER
Active	☐		
No. of History Objects	5		
No. of Objects	4		

E..	Object T...	Object No.	Object Name	Modified	Compiled
▪	Table	5603	FA Setup	✓	✓
⊟	Table	50000	Car	✓	✓
	Table	50000	Car	✓	✓
▪	Form	50000	Cars		✓
▪	Page	50000	Cars		✓

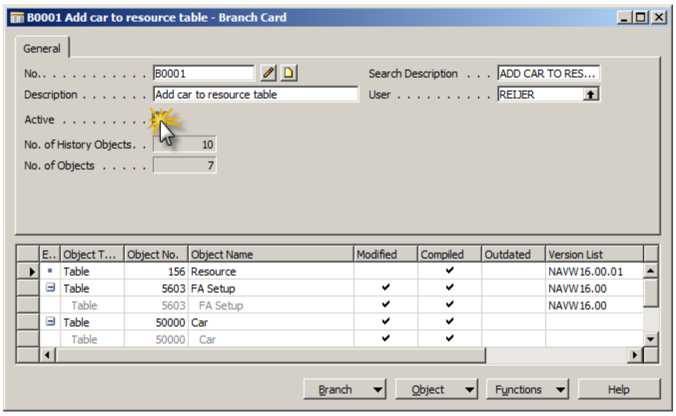
- **Get Latest Versions**
  If the version of an object in your branch is outdated a new line for the latest version will be added.

- **Show Outdated History Objects**
  Shows the latest version of the outdated objects in your branch. This can be used to roll back the objects in the database to the last known version of the branch.

- **Show Initial History Objects**
  Show the first version of the objects in the branch. This can be used to roll back the objects in the database to the initial state of the branch.

10.6. Activate a Branch

When you want to work on a branch and reserve the objects in the branch you can activate it. All objects in the branch will be locked by you.



B0001 Add car to resource table - Branch Card

General

No. B0001

Description Add car to resource table

Active

No. of History Objects 10

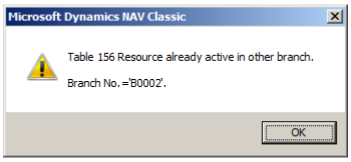
No. of Objects 7

Search Description ADD CAR TO RES...

User REIJER

E..	Object T...	Object No.	Object Name	Modified	Compiled	Outdated	Version List
▪	Table	156	Resource		✓		NAVW16.00.01
⊟	Table	5603	FA Setup	✓	✓		NAVW16.00
	Table	5603	FA Setup	✓	✓		NAVW16.00
⊟	Table	50000	Car	✓	✓		
	Table	50000	Car	✓	✓		

If an object is active in another branch you get an error.

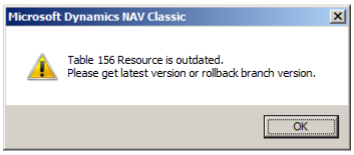


Microsoft Dynamics NAV Classic

Table 156 Resource already active in other branch.

Branch No.='B0002'.

If there is an object in your branch that is outdated you get the following error.



Microsoft Dynamics NAV Classic

Table 156 Resource is outdated.
Please get latest version or rollback branch version.

Because of this two checks it is possible to have an object in more than one branch with totally separated functionality.

11. Action Worksheet

You can perform actions in your own database, like filling fields, but you can also include them in a project so you can transport these actions to the customer database and execute them in there.

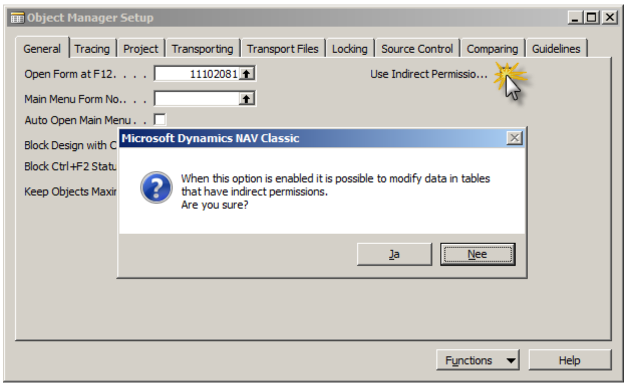
You can find the “Action Worksheet” in menu “Data Tools” or access it from the project- or transport card.

Action Types

- Copy Data: copy data between tables or companies
- Delete Data: delete records in a table or empty fields
- Transfer Data: transfer data between databases
- Fill Fields: fill fields with values
- Run Report: run a report
- Run Codeunit: run a codeunit
- Run Dataport: run a dataport
- Rename Data: Renames data
- Renumber Object: renames an object
- Renumber Field: renames a field
- Execute SQL Query: executes an SQL query
- Execute DOS Command: executes a DOS command

11.1. Setup

When you are using a customer license and you want to update data that is only allowed through indirect permissions you can enable the setting “Use Indirect Permissions” when executing actions.



Open Form at F12. . . . 11102081

When this option is enabled it is possible to modify data in tables that have indirect permissions.
Are you sure?

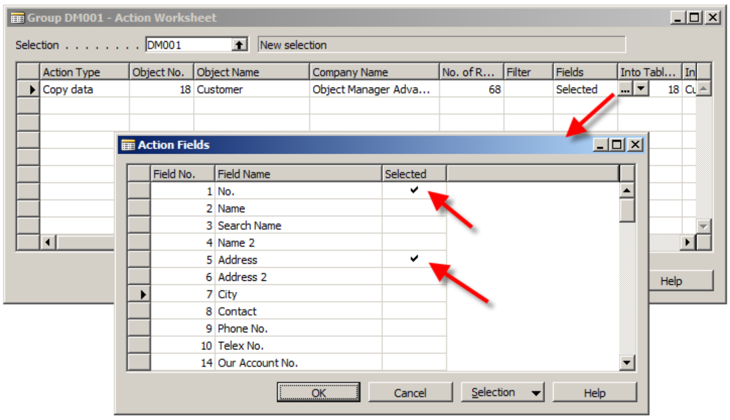
Now it possible to modify data in tables like “G/L Entry”.

11.2. Copy Data

You can copy data in the same or another table or between two companies.

Example
Copy the Customer Address to Address 2

1. Select the fields that you want to copy

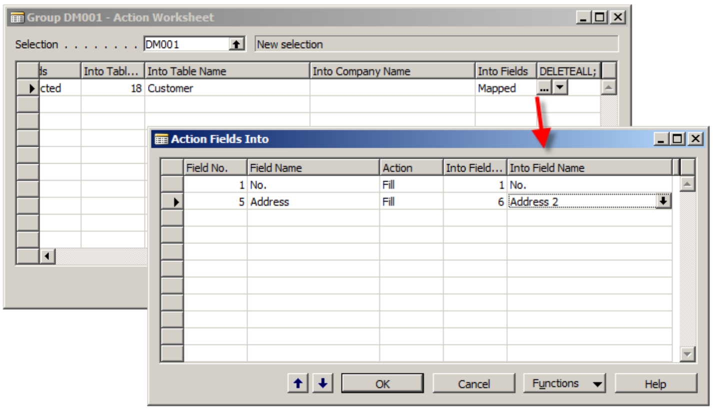


Selection . . . . DM001 New selection

Action Type	Object No.	Object Name	Company Name	No. of R...	Filter	Fields	Into Tabl...	In
Copy data	18	Customer	Object Manager Adva...	68		Selected	18	Cu

Field No.	Field Name	Selected
1	No.	✓
2	Name	
3	Search Name	
4	Name 2	
5	Address	✓
6	Address 2	
7	City	
8	Contact	
9	Phone No.	
10	Telex No.	
14	Our Account No.	

2. Select the table where you want to copy the data to. In our case the same Customer table.
3. Click the assist-edit button of “Into Fields” to map the fields where you want to put the data in.

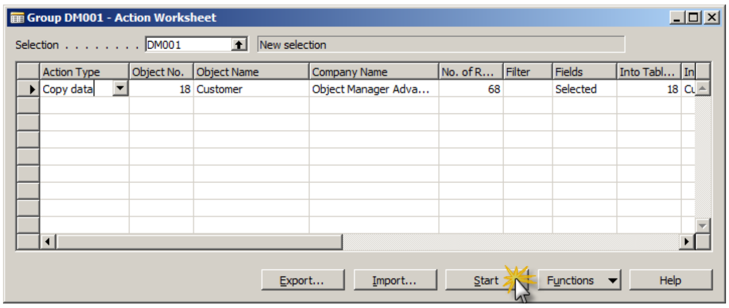


Selection . . . . DM001 New selection

ds	Into Tabl...	Into Table Name	Into Company Name	Into Fields	DELETEALL;
cted	18	Customer		Mapped	

Field No.	Field Name	Action	Into Field...	Into Field Name
1	No.	Fill	1	No.
5	Address	Fill	6	Address 2

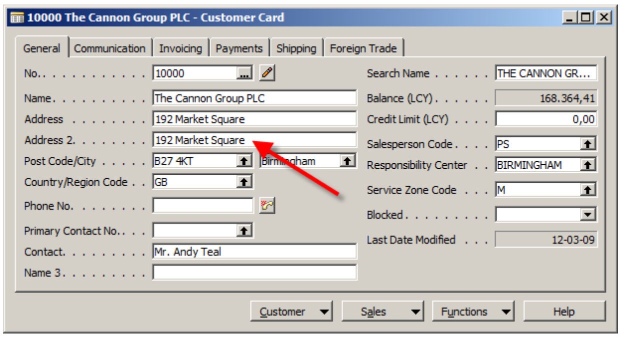
4. Press Start



Selection . . . . DM001 New selection

Action Type	Object No.	Object Name	Company Name	No. of R...	Filter	Fields	Into Tabl...	In
Copy data	18	Customer	Object Manager Adva...	68		Selected	18	Cu

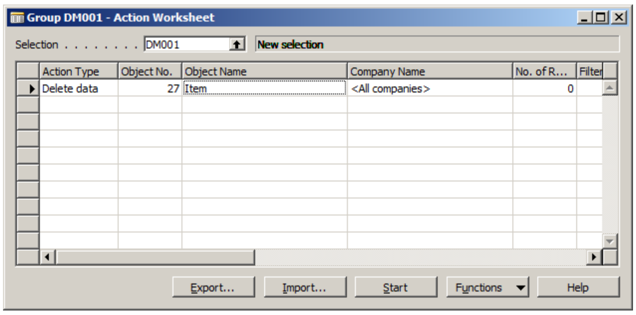
5. Result:



No. 10000
Name The Cannon Group PLC
Address 192 Market Square
Address 2 192 Market Square
Post Code/City B27 4KT Birmingham
Country/Region Code GB
Phone No.
Primary Contact No.
Contact Mr. Andy Teal
Name 3
Search Name THE CANNON GR...
Balance (LCY) 168.364,41
Credit Limit (LCY) 0,00
Salesperson Code PS
Responsibility Center BIRMINGHAM
Service Zone Code M
Blocked
Last Date Modified 12-03-09

11.3. Delete Data

With this action type you can delete data from all records, filtered records and from specific fields. So if you have to delete a field from a table you can first empty it with this action type.



Selection DM001 New selection

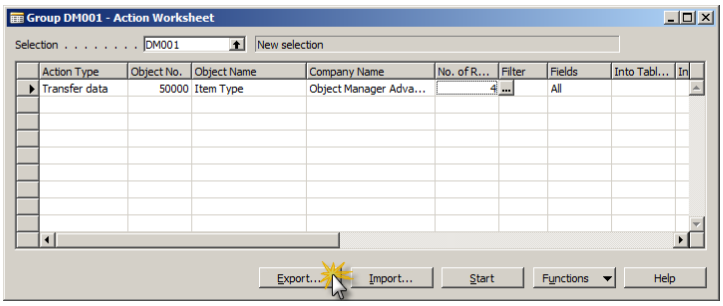
Action Type	Object No.	Object Name	Company Name	No. of R...	Filter
Delete data	27	Item	<All companies>	0	

11.4. Transfer Data

Transfer data from one table to another table in another database.

Example

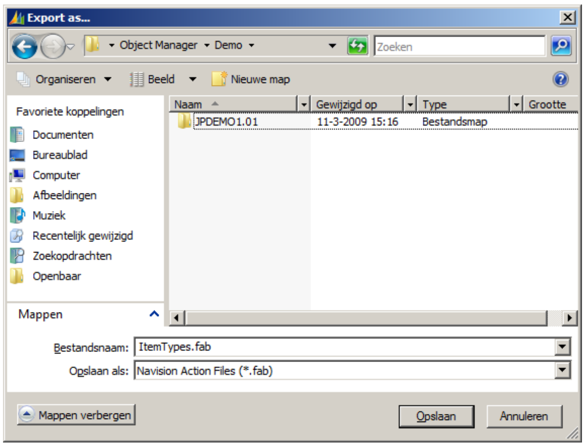
1. Action Type “Transfer Data”
2. Fill in the “Table No.” were you want to transfer data from
3. Press Export



Selection DM001 New selection

Action Type	Object No.	Object Name	Company Name	No. of R...	Filter	Fields	Into Tabl...	In
Transfer data	50000	Item Type	Object Manager Adva...	4	...	All		

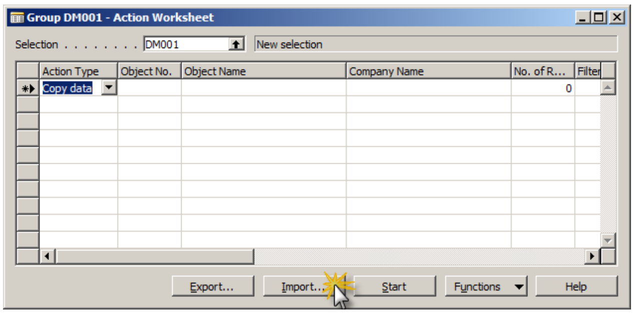
4. The Object Manager Exports the action + data as a FAB file
5. Save the file



Naam	Gewijzigd op	Type	Grootte
JPDEMO1.01	11-3-2009 15:16	Bestandsmap	

Bestandsnaam: ItemTypes.fab
Opslaan als: Navision Action Files (*.fab)

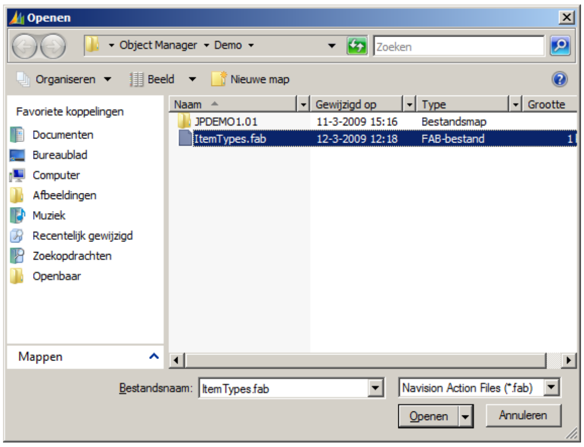
6. Open the Customer database
7. Open the “Action Worksheet”
8. Press Import



Selection DM001

Action Type	Object No.	Object Name	Company Name	No. of R...	Filter
Copy data				0	

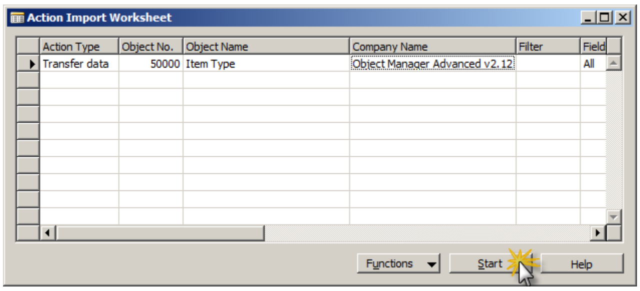
9. Open the FAB file



Naam	Gewijzigd op	Type	Grootte
JPDEMO1.01	11-3-2009 15:16	Bestandsmap	
ItemTypes.fab	12-3-2009 12:18	FAB-bestand	1

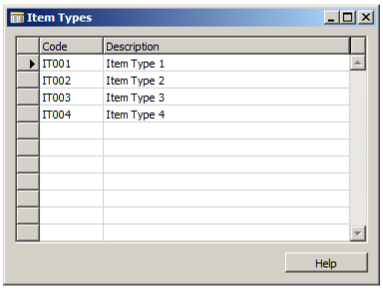
Bestandsnaam: ItemTypes.fab
Navision Action Files (*.fab)

10. The Action is imported in the “Action Import Worksheet”
11. Press Start



Action Type	Object No.	Object Name	Company Name	Filter	Field
Transfer data	50000	Item Type	Object Manager Advanced v2.12		All

12. Data from the development database is now in your Customer database

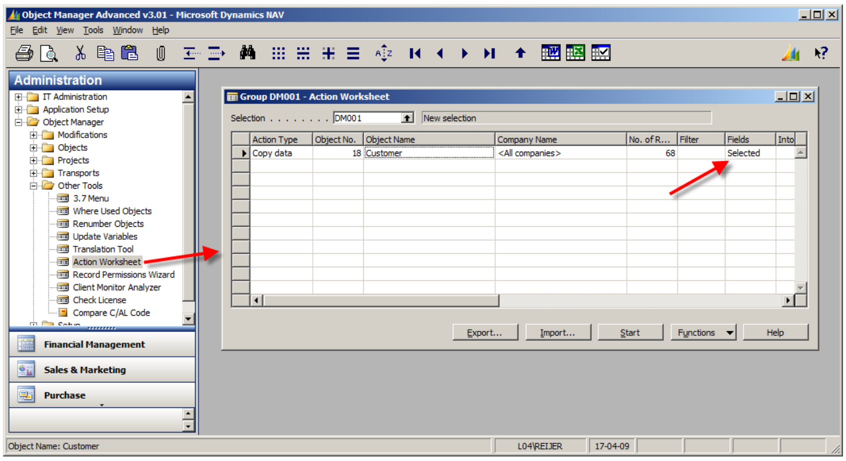


Code	Description
IT001	Item Type 1
IT002	Item Type 2
IT003	Item Type 3
IT004	Item Type 4

11.5. Fill Fields

Suppose you want to fill the “Customer Price Group” in the Customer table with value “PG001”.

1. Open Action Worksheet
2. Action Type Fill fields, “Object No.” 18 (Customer)
3. If you use multiple companies in NAV you can select All or a specific Company to perform the action on
4. Push the assist-edit button in field Fields

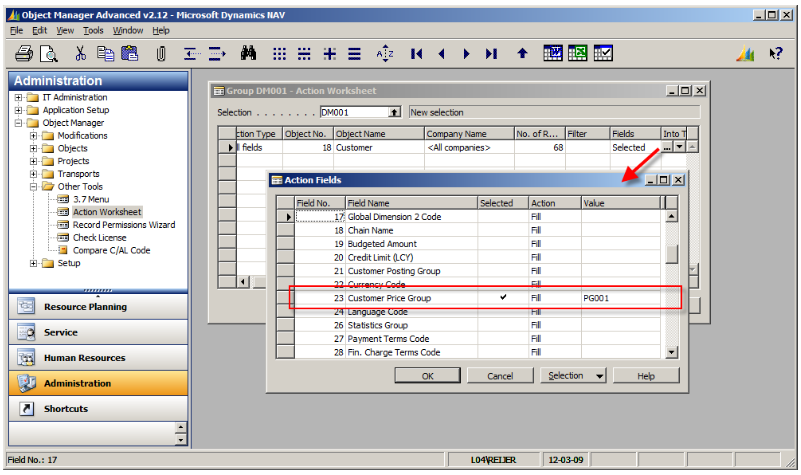


Selection . . . . DM001 New selection

Action Type	Object No.	Object Name	Company Name	No. of R...	Filter	Fields	Into
Copy data	18	Customer	<All companies>	68		Selected	

Object Name: Customer L04\REIJER 17-04-09

5. Select the “Customer Price Group” field by checking the Selected field
6. Fill in the field Value “PG001”, this is the value to fill



Group DM001 - Action Worksheet

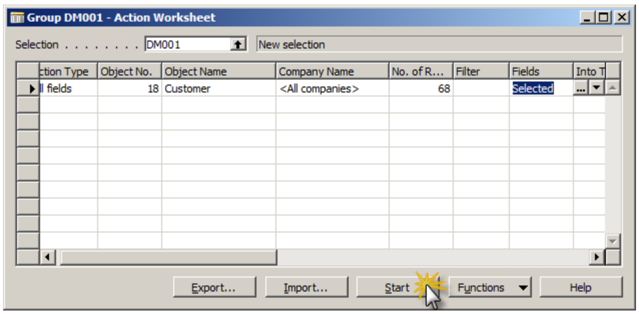
Selection . . . . DM001 New selection

ction Type	Object No.	Object Name	Company Name	No. of R...	Filter	Fields	Into T
l fields	18	Customer	<All companies>	68		Selected	

Action Fields

Field No.	Field Name	Selected	Action	Value
17	Global Dimension 2 Code		Fill	
18	Chain Name		Fill	
19	Budgeted Amount		Fill	
20	Credit Limit (LCY)		Fill	
21	Customer Posting Group		Fill	
22	Currency Code		Fill	
23	Customer Price Group	✓	Fill	PG001
24	Language Code		Fill	
26	Statistics Group		Fill	
27	Payment Terms Code		Fill	
28	Fin. Charge Terms Code		Fill	

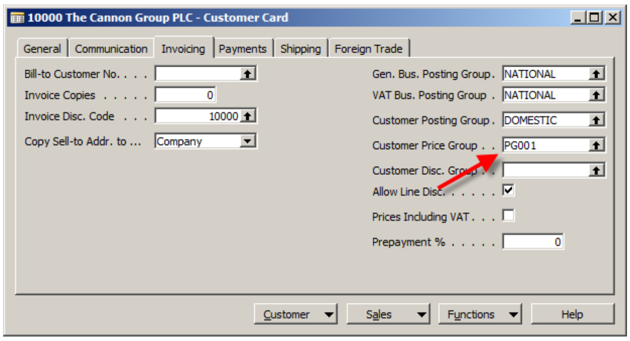
7. Return to the Worksheet and push Start. For all the Customers the “Customer Price Group” is filled with value “PG001”



Group DM001 - Action Worksheet

Selection . . . . DM001 New selection

ction Type	Object No.	Object Name	Company Name	No. of R...	Filter	Fields	Into T
l fields	18	Customer	<All companies>	68		Selected	

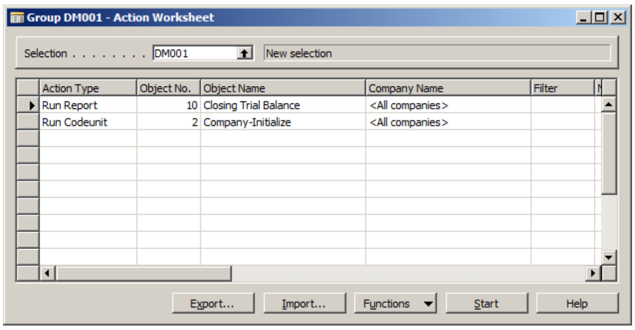


10000 The Cannon Group PLC - Customer Card

Bill-to Customer No. . . .
Invoice Copies 0
Invoice Disc. Code . . . 10000
Copy Sell-to Addr. to ... Company
Gen. Bus. Posting Group . NATIONAL
VAT Bus. Posting Group . NATIONAL
Customer Posting Group . DOMESTIC
Customer Price Group . . PG001
Customer Disc. Group . .
Allow Line Disc. . . . . . ☑
Prices Including VAT . . . ☐
Prepayment % 0

11.6. Run Report, Codeunit or Dataport

You can also execute reports, codeunits or dataports.



Selection . . . . DM001 New selection

Action Type	Object No.	Object Name	Company Name	Filter
Run Report	10	Closing Trial Balance	<All companies>	
Run Codeunit	2	Company-Initialize	<All companies>	

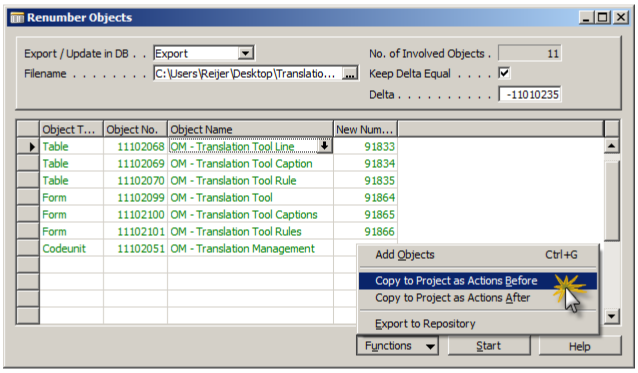
11.7. Rename Data

With the rename data option you can rename data.

Important note: In NAV3.6 and NAV4 the data is not renamed but the old record is deleted and the new record is inserted. This is because the function RENAME for RecordRef is only available in NAV5 and above.

11.8. Renumber Object

The renumber object function can be used to give an object another number. All references to this object will not be changed. So if you are doing a renumber action in your development database it is preferred to do this with the renumber objects function. For more information see chapter *21 - Renumber Objects*. If this renumber action is also needed in your customer database then it is possible to copy this renumber action to a specific project as “action before” with the function “Copy to Project as Action Before”.



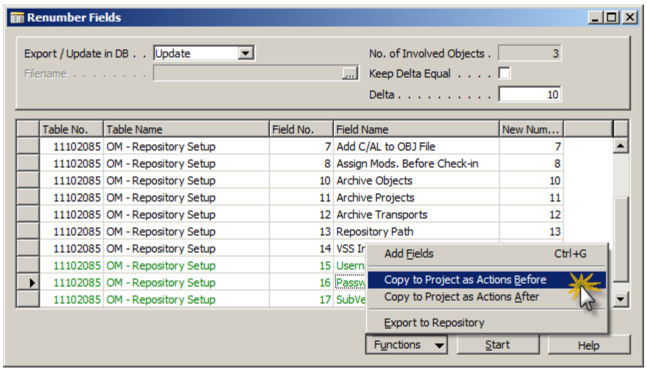
Export / Update in DB . . Export
Filename . . . . C:\Users\Reijer\Desktop\Translatio...
No. of Involved Objects . 11
Keep Delta Equal . . . . ☑
Delta . . . . -11010235

Object T...	Object No.	Object Name	New Num...
Table	11102068	OM - Translation Tool Line	91833
Table	11102069	OM - Translation Tool Caption	91834
Table	11102070	OM - Translation Tool Rule	91835
Form	11102099	OM - Translation Tool	91864
Form	11102100	OM - Translation Tool Captions	91865
Form	11102101	OM - Translation Tool Rules	91866
Codeunit	11102051	OM - Translation Management	

NOTE: Make the “Actions Before” before you start the renumbering.

11.9. Renumber Field

The renumber field function can be used to give a table field another number. All references to this object will not be changed. So if you are doing a renumber action in your development database it is preferred to do this with the renumber fields function. For more information see chapter _22 - Renumber Fields_. If this renumber action is also needed in your customer database then it is possible to copy this renumber action to a specific project as “actions before” with the function “Copy to Project as Action Before”.



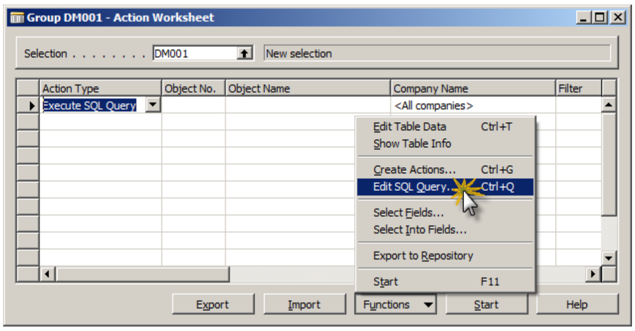
Export / Update in DB . . Update
Filename . . . .
No. of Involved Objects . 3
Keep Delta Equal . . . . ☐
Delta 10

Table No.	Table Name	Field No.	Field Name	New Num...
11102085	OM - Repository Setup	7	Add C/AL to OBJ File	7
11102085	OM - Repository Setup	8	Assign Mods. Before Check-in	8
11102085	OM - Repository Setup	10	Archive Objects	10
11102085	OM - Repository Setup	11	Archive Projects	11
11102085	OM - Repository Setup	12	Archive Transports	12
11102085	OM - Repository Setup	13	Repository Path	13
11102085	OM - Repository Setup	14	VSS Ir	
11102085	OM - Repository Setup	15	Usern	
11102085	OM - Repository Setup	16	Passw	
11102085	OM - Repository Setup	17	SubVe	

NOTE: Make the “Actions Before” before you start the renumbering.

11.10. Execute SQL Query

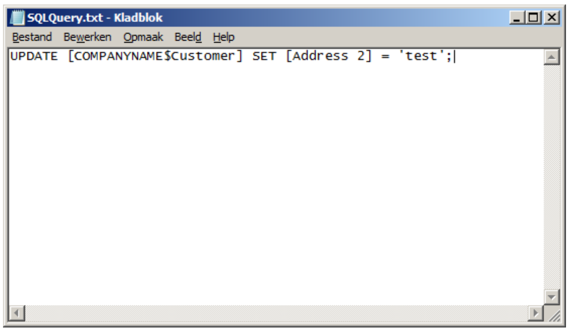
You can execute an SQL query.



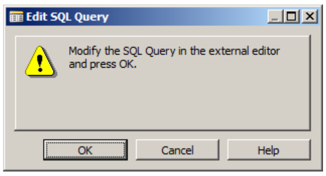
Selection . . . . DM001 New selection

Action Type	Object No.	Object Name	Company Name	Filter
Execute SQL Query			<All companies>	

Press Functions > Edit SQL Query and make your query in your text editor. When finished save the SQLQuery.txt file and press OK in the pending dialog.

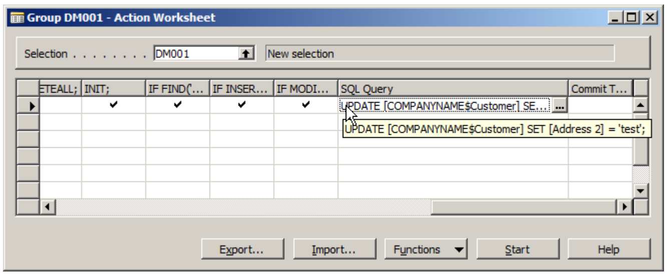


```
UPDATE [COMPANYNAME$Customer] SET [Address 2] = 'test';
```



Modify the SQL Query in the external editor and press OK.

NOTE: The string COMPANYNAME will be replaced by the companies you have selected in the action.



Selection DM001

ETEALL;	INIT;	IF FIND('...	IF INSER...	IF MODI...	SQL Query	Commit T...
	✓	✓	✓	✓	UPDATE [COMPANYNAME$Customer] SE...	

UPDATE [COMPANYNAME$Customer] SET [Address 2] = 'test';

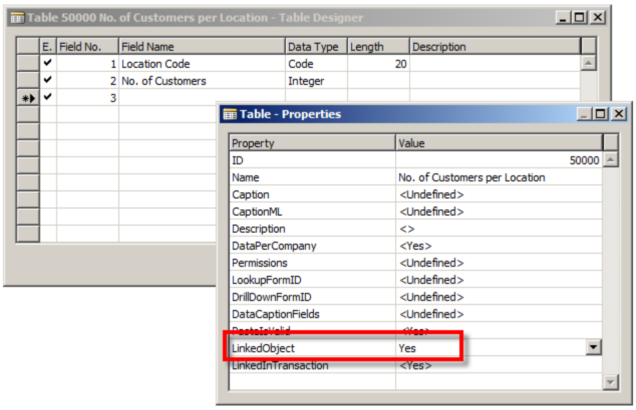
11.10.1. Example SQL Query to Change Data

If you e.g. want to move the content of the address field to the address 2 field in the customer table you can use the following query.

```
UPDATE [COMPANYNAME$Customer] SET [Address 2] = [Address] WHERE [Address] <> '';
UPDATE [COMPANYNAME$Customer] SET [Address] = '';
```

11.10.2. Example SQL Query to Add a View

If you want to send a LinkedObject to your customer database you also want the corresponding view or table to be created in your customer SQL database.



E.	Field No.	Field Name	Data Type	Length	Description
✓	1	Location Code	Code	20	
✓	2	No. of Customers	Integer		
✓	3				

Property	Value
ID	50000
Name	No. of Customers per Location
Caption	<Undefined>
CaptionML	<Undefined>
Description	<>
DataPerCompany	<Yes>
Permissions	<Undefined>
LookupFormID	<Undefined>
DrillDownFormID	<Undefined>
DataCaptionFields	<Undefined>
PasteIsValid	<Yes>
LinkedObject	Yes
LinkedInTransaction	<Yes>

You can do this by creating two actions of type “Execute SQL Query”.

The first one is to remove the existing view.

```
IF EXISTS (SELECT * FROM sys.views WHERE object_id = OBJECT_ID(N'[dbo].[COMPANYNAME$No_ of Customers per Location]'))
DROP VIEW [dbo].[COMPANYNAME$No_ of Customers per Location]
```

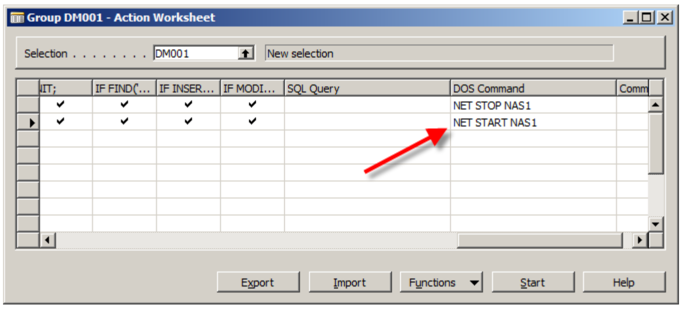
The second is to add the new view.

```
CREATE VIEW [dbo].[COMPANYNAME$No_ of Customers per Location]
AS
SELECT [Location Code], COUNT(No_) AS [No_ of Customers]
FROM dbo.[COMPANYNAME$Customer]
GROUP BY [Location Code]
```

11.11. Execute DOS Command

You can execute a DOS Command as action.

This can be useful if you need to start an application or script during or after an import. This can also be used to restart a NAS.

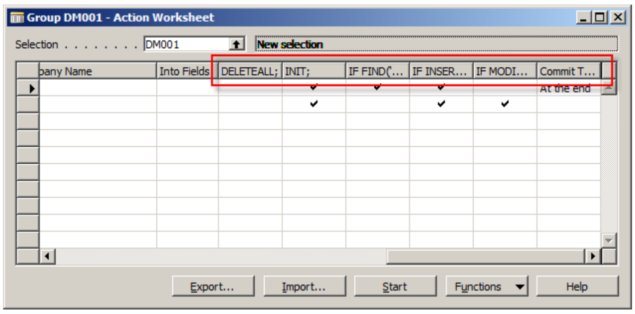


Group DM001 - Action Worksheet

Selection . . . . DM001 New selection

IT;	IF FIND('...	IF INSER...	IF MODI...	SQL Query	DOS Command	Comm
✓	✓	✓	✓		NET STOP NAS1	
✓	✓	✓	✓		NET START NAS1	

11.12. Save Options



Group DM001 - Action Worksheet

Selection . . . . DM001 New selection

pany Name	Into Fields	DELETEALL;	INIT;	IF FIND('...	IF INSER...	IF MODI...	Commit T...
			✓	✓	✓		At the end
			✓		✓	✓	

- **DELETEALL;**
  With this option the table will first be emptied before the action is executed.

- **INIT;**
  Every new record is first initialized before the action is executed.

- **IF FIND('=') THEN;**
  - True: if this option is enabled the action first reads the key fields into the new record and then tries to find the existing record. If found the existing record will be modified, otherwise a new record will be inserted (also depending on the next two options "IF INSERT THEN;" and "IF MODIFY THEN;").

    - False: if this option is disabled a new record will be used if used in combination with “INIT;”. The previous record will be used if the “INIT;” option is also disabled.

- **IF INSERT THEN;**
  If disabled, no new records will be created.

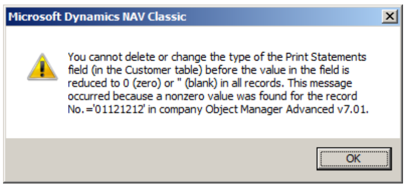
- **IF MODIFY THEN;**
  If disabled, no existing records will be modified.

- **Commit Type**
  Indicates how many times a commit is executed.
    - <EMPTY>: No committing is done. Only when all actions are executed
    - At the end: Commit is done when this action is executed.
    - After each record
    - After 100 records

11.13. Add Actions to a Project

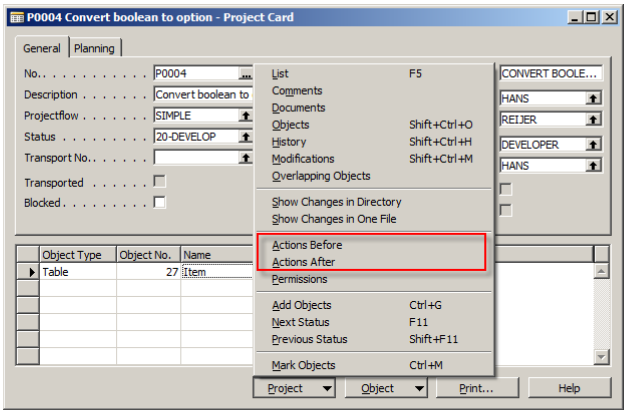
You can add actions to a project. These actions are executed in your customer database when you import the belonging transport. You can perform actions before reading the objects in a database and after.

Example
You have an existing table and you changed the property type of a field from Boolean to Option. If you read in a FOB file the conventional way, you will get an error if that field was filled in your customer database. You have to empty that field first for all records.



You cannot delete or change the type of the Print Statements field (in the Customer table) before the value in the field is reduced to 0 (zero) or '' (blank) in all records. This message occurred because a nonzero value was found for the record No.='01121212' in company Object Manager Advanced v7.01.

You can perform an “Action Before” that deletes the contents of that field before reading the new objects in the database.



P0004 Convert boolean to option - Project Card

No. P0004

Description Convert boolean to

Projectflow SIMPLE

Status 20-DEVELOP

Transport No.

Transported ☐

Blocked ☐

CONVERT BOOLE...

HANS

REIJER

DEVELOPER

HANS

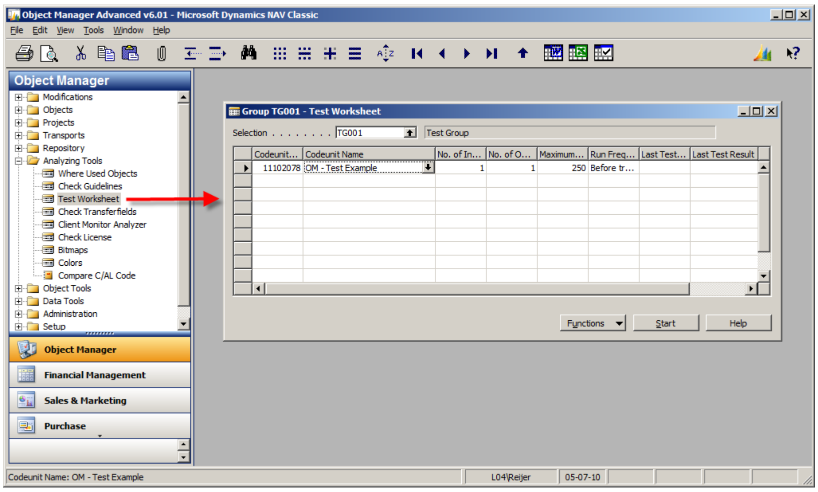
Object Type	Object No.	Name
Table	27	Item

With these functions you can define the actions and export them later with the project into the transport file.

12. Test Framework

The Test Framework is used to automate tests. If you have a certain process that has to be tested before you transport objects to your customer database you can write a codeunit with input and output parameters that will be tested before the transport is executed.

12.1. Create Test



Selection . . . . TG001 Test Group

Codeunit...	Codeunit Name	No. of In...	No. of O...	Maximum...	Run Freq...	Last Test...	Last Test Result
11102078	OM - Test Example	1	1	250	Before tr...		

Codeunit Name: OM - Test Example L04\Reijer 05-07-10

You can find the “Test Worksheet” in menu “Analyzing Tools” or access it from the project- or transport card. If you add a test to a project it will be transported to your customer database and it can be tested before you do the transport.

A test has the following options:

- **Codeunit No.**
  The codeunit that will execute the test. See section 12.2 - An example of a Test Codeunit for an example.
- **Codeunit Name**
  The name of the codeunit.
- **No. of Input Parameters**
  The number of input parameters that is used in the test codeunit.
- **No. of Output Parameters**
  The number of output parameters that is used in the test codeunit. If one of the output parameters has another value the test will fail.
- **Maximum Duration (ms)**
  If the duration of the test is longer than this value the test will fail.

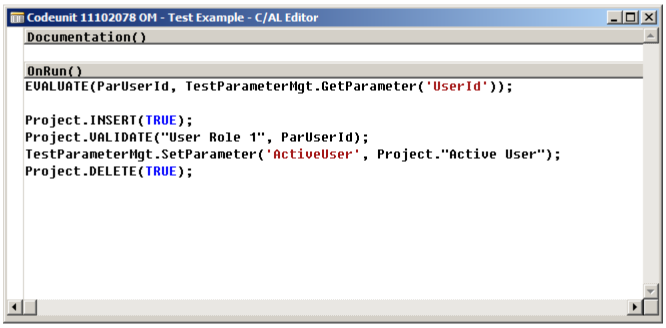
- **Run Frequency**
    - Only manual: The test will only be executed manually in the test worksheet.
    - Before transport: The test will be executed before it is transported. This option is only available if the test is added to a project
    - Before every transport: The test is executed every time a transport is done.

- **Last Test Succeeded**
  The result of the last executed test.

- **Last Test Result**
  The result of the last executed test. If an error has occurred the error message is shown in this field.

12.2. An example of a Test Codeunit

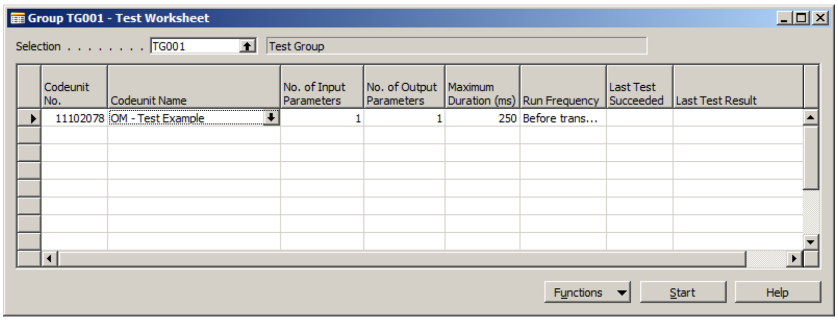
Codeunit 11102078 - OM - Test Example is a simplified example of how a test codeunit could look like.



```
Documentation()

OnRun()
EVALUATE(ParUserId, TestParameterMgt.GetParameter('UserId'));

Project.INSERT(TRUE);
Project.VALIDATE("User Role 1", ParUserId);
TestParameterMgt.SetParameter('ActiveUser', Project."Active User");
Project.DELETE(TRUE);
```

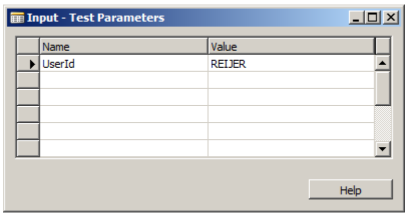
This codeunit will test if in a new project the user that is validated in the first user role will be the active user.

In this example you see that it has 1 input parameter and 1 output parameter.

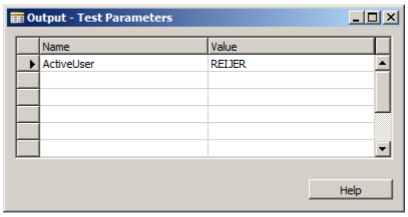


Selection . . . . TG001 Test Group

Codeunit No.	Codeunit Name	No. of Input Parameters	No. of Output Parameters	Maximum Duration (ms)	Run Frequency	Last Test Succeeded	Last Test Result
11102078	OM - Test Example	1	1	250	Before trans...		

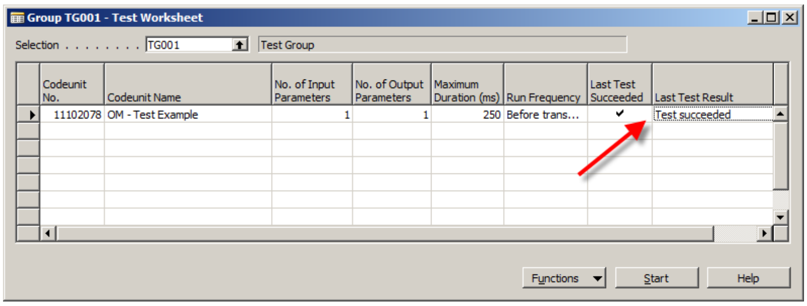

Input - Test Parameters

Name	Value
UserId	REIJER


Output - Test Parameters

Name	Value
ActiveUser	REIJER

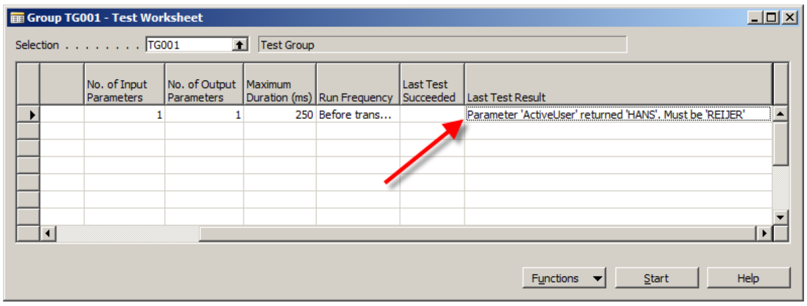
The result of this test will be “Test succeeded”.


Group TG001 - Test Worksheet

Selection . . . . TG001 Test Group

Codeunit No.	Codeunit Name	No. of Input Parameters	No. of Output Parameters	Maximum Duration (ms)	Run Frequency	Last Test Succeeded	Last Test Result
11102078	OM - Test Example	1	1	250	Before trans…	✓	Test succeeded

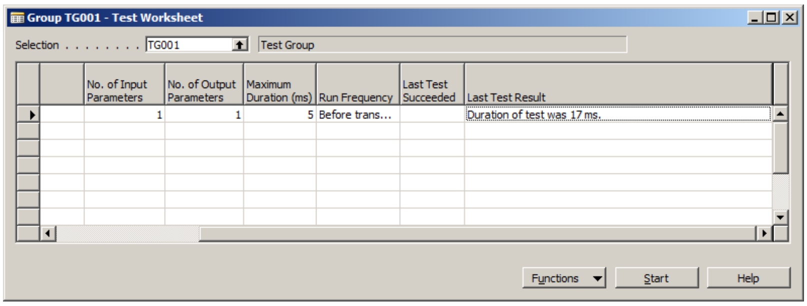
If you have done a modification to the project module which will result in another active user the test will fail and give you the result “Parameter 'ActiveUser' returned 'HANS'. Must be 'REIJER'”


Group TG001 - Test Worksheet

Selection . . . . TG001 Test Group

No. of Input Parameters	No. of Output Parameters	Maximum Duration (ms)	Run Frequency	Last Test Succeeded	Last Test Result
1	1	250	Before trans…		Parameter 'ActiveUser' returned 'HANS'. Must be 'REIJER'

If you lower the maximum duration to 5 ms. the test will also fail and the result will be “Duration of test was 17 ms.”.


Group TG001 - Test Worksheet

Selection . . . . TG001 Test Group

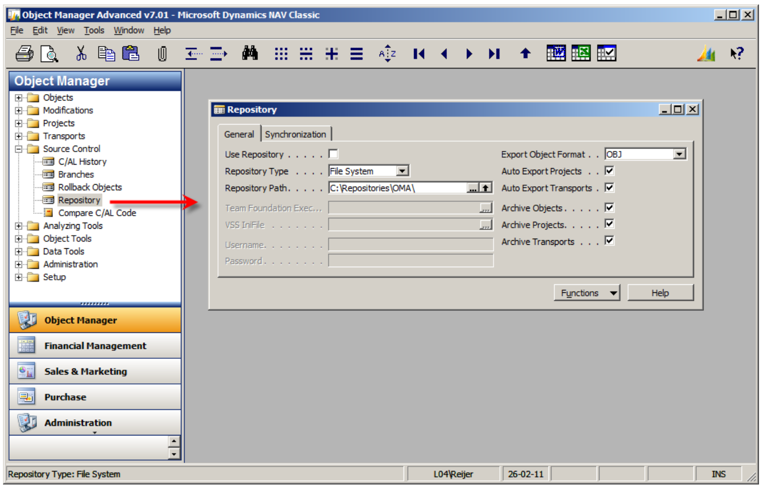
No. of Input Parameters	No. of Output Parameters	Maximum Duration (ms)	Run Frequency	Last Test Succeeded	Last Test Result
1	1	5	Before trans…		Duration of test was 17 ms.

13. Repository

Repository is used when you want to save your objects outside NAV. You can save your objects in a repository like VSS, TFS or SVN. Every time an object will be checked-in in the Object Explorer the object will be send to the repository.

You can use the repository with and without synchronization. The option with synchronization is used if you want to sync separated databases. Every change that is done in one database is also executed in all other databases that are connected to the same repository.

13.1. Repository without Synchronization



Use Repository . . . . . ☐
Repository Type . . . . File System
Repository Path. . . . . C:\Repositories\OMA\
Team Foundation Exec...
VSS IniFile . . . .
Username. . . . .
Password . . . .
Export Object Format . . OBJ
Auto Export Projects . . ☑
Auto Export Transports . ☑
Archive Objects . . . . . ☑
Archive Projects. . . . . ☑
Archive Transports . . . ☑

Repository Type: File System | L04\Reijer | 26-02-11 | INS

- **Use Repository**
  Set this flag if you want to use a repository

- **Repository Type**
  The type of the repository. One of the following options can be chosen:
  - File System
  - Visual Source Safe
  - Team Foundation Server
  - SubVersion

- **Repository Path**
  - File System: The path where all the files are saved that the repository is using.
  - Team Foundation Server: The local path that the repository is using to save the files.

- **Team Foundation Executable**
  The location of the executable that can be used to talk to Team Foundation Server. In most cases the executable is

located in the Visual Studio directory like: “C:\Program Files\Microsoft Visual Studio 10.0\Common7\IDE\TF.exe”

- **VSS IniFile**
  The path to the INI file that is used if you use repository type “Visual Source Safe”.

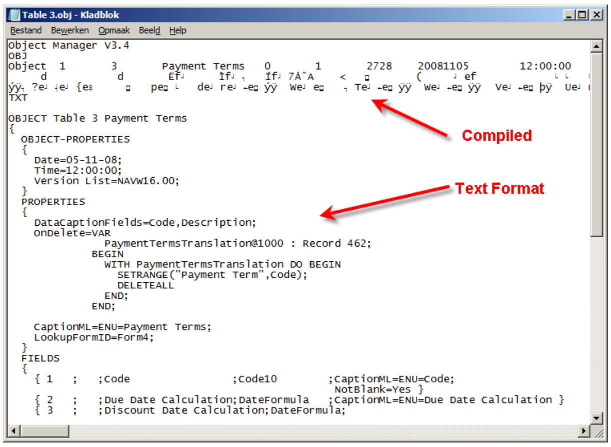
- **SubVersion Executable**
  The path to the SubVersion executable if you use repository type “Visual Source Safe”.

- **Username**
  The username that is used to login to the repository.

- **Password**
  The password that is used to login to the repository.

- **Export Object Format**
  OBJ: Objects will be exported in OBJ format. This is a file format with only the compiled version of the object and can be imported with the “Objects Import Worksheet”

  OBJ + TXT: Objects will be exported in OBJ format + TXT format.



```
Object Manager V3.4
OBJ
Object  1       3       Payment Terms   0       1       2728    20081105        12:00:00
     d           d          Ëf┘       Ìf┘ ┐    Íf┘ 7Å˜A    <   ◘        (      ┘ ef             └ └
ýÿ┐ ?e┘ ┤e┘ {e╪      ◘    pe◘ └    de┘ re┘ ┴e◘ ýÿ  We┘ e◘    ┐ Te┘ ┴e◘ ýÿ  We┘ ┴e◘ ýÿ  Ve┘ ┴e◘ þÿ  Ue┘
TXT

OBJECT Table 3 Payment Terms
{
  OBJECT-PROPERTIES
  {
    Date=05-11-08;
    Time=12:00:00;
    Version List=NAVW16.00;
  }
  PROPERTIES
  {
    DataCaptionFields=Code,Description;
    OnDelete=VAR
               PaymentTermsTranslation@1000 : Record 462;
             BEGIN
               WITH PaymentTermsTranslation DO BEGIN
                 SETRANGE("Payment Term",Code);
                 DELETEALL
               END;
             END;

    CaptionML=ENU=Payment Terms;
    LookupFormID=Form4;
  }
  FIELDS
  {
    { 1   ;   ;Code                ;Code10        ;CaptionML=ENU=Code;
                                                   NotBlank=Yes }
    { 2   ;   ;Due Date Calculation;DateFormula   ;CaptionML=ENU=Due Date Calculation }
    { 3   ;   ;Discount Date Calculation;DateFormula;
```

  TXT: The object will be exported in text format. The files will get TXT as extension.

- **Archive Objects**
  Each time an object changes a copy of this object will be saved to the archive directory in the repository.

- **Archive Projects**
  Each time a project changes a copy of this project will be saved to the archive directory in the repository.

- **Archive Transport**
  Each time a transport changes a copy of this transport will be saved to the archive directory in the repository.

13.1.1. Create Repository

When you create a repository all the directories are created in the repository and a token will be placed in the root. The directory structure looks like this:

- Archive
  - Objects
  - Projects
  - Transports
- Locks
- Log
  - 0-9999
    - 0-99
- Objects
- Projects
- Transports

13.1.2. Export

With this option all objects, projects and transports will be exported to the repository.

13.1.3. Import

This option imports all objects, projects, transports and locks from the repository.

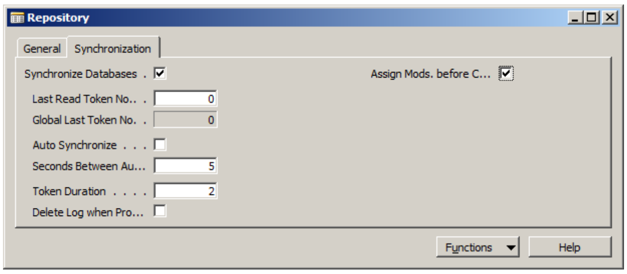
13.1.4. Lock Repository

Locks the repository. This places a file with the name LOCKED in the root of the repository. Nobody can read or write to the repository until somebody unlocks the repository

13.1.5. Unlock Repository

Unlocks the repository. The file with name LOCKED will be deleted from the root of the repository.

13.2. Repository with Synchronization



Synchronize Databases . ☑
Last Read Token No.. . 0
Global Last Token No. . 0
Auto Synchronize . . . ☐
Seconds Between Au... 5
Token Duration 2
Delete Log when Pro... ☐
Assign Mods. before C... ☑

- **Synchronize Databases**
  Use this option to enable synchronization between databases.

- **Last Read Token No.**
  The token number that is last read. This field is used to prevent that a specific token is read 2 times.

- **Global Last Token No.**
  The last used token number of the repository. This is directly read from the repository. If this field is empty then the repository cannot be reached.

- **Auto Synchronize**
  Use this option to poll if something changed in the repository. If something changed these modifications are automatically imported in your database and executed.

- **Seconds Between Auto Sync.**
  Seconds between each time the repository is checked for new changes.

- **Toke Duration**
  Seconds that a token is seen as valid. If one of the connected users sees that your token is busy for this number of seconds it is killed and you get a warning that the token duration has to be increased.

  If you use a repository that is on a WAN or internet then it can be necessary to increase this value.

- **Delete Log when Processed**
  Each modification will be saved to a log table. If you enabled this option these records will be deleted after they are processed

- **Assign Mods. Before Check-in**
  Before you check-in an object you have to assign the modifications on this objects to a project. This prevents that these modifications have to be assigned in each database individually.

13.2.1. Test Connection

The connection to the repository will be tested. If something is wrong you will get a detailed error message.

13.2.2. Connect

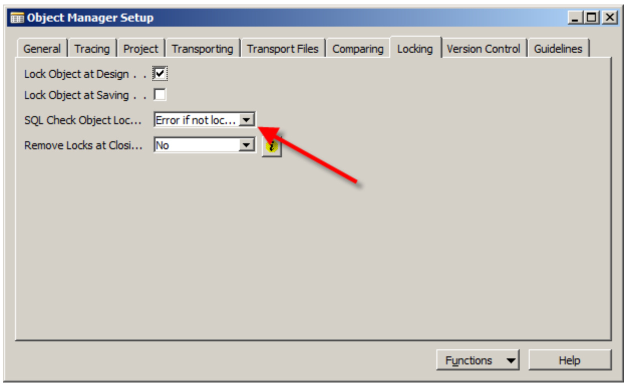
Sets the “Last Read Token No.” to the token number of the repository. Use this option if you want to connect to an existing repository but you do not want to process the log. It is highly recommended to import all objects, projects, transports and locks from the repository before you connect to an existing repository.

13.2.3. Synchronize

This option processes all pending log records.

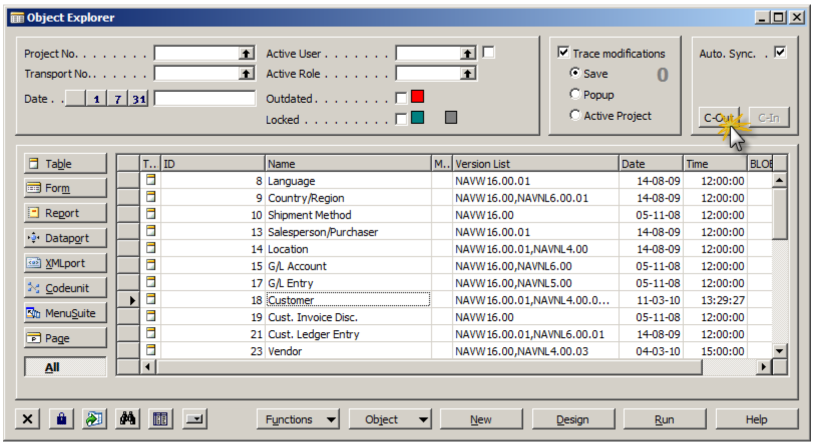
13.3. Check-out and Check-in Objects

If you use a repository it is highly recommended to enable the SQL check that an object can only be changed if it is locked. You can enable this setting in the setup by setting “SQL Check Lock Type” to “Error if not locked”. Now it is prevented that two developers are working on the same object.



Lock Object at Design . . ☑
Lock Object at Saving . . ☐
SQL Check Object Loc... Error if not loc...
Remove Locks at Closi... No

Before you can change an object you first have to check-out this object from the repository. You can do that in the Object Explorer by pressing the C-Out button or Ctrl + L:

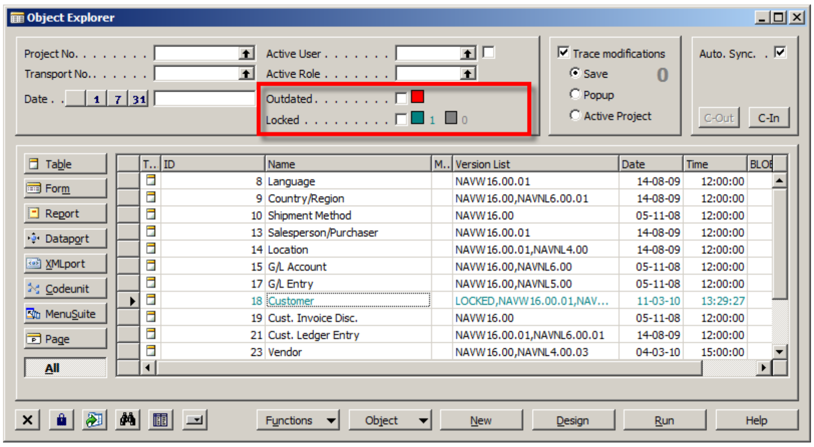


Project No. . . . .
Transport No. . . . . . .
Date . .
Active User . . . . ☐
Active Role . . . .
Outdated . . . . ☐ 🟥
Locked . . . . ☐ 🟩 ▨
☑ Trace modifications
● Save 0
○ Popup
○ Active Project
Auto. Sync. . ☑

T..	ID	Name	M..	Version List	Date	Time	BLOB
	8	Language		NAVW16.00.01	14-08-09	12:00:00	
	9	Country/Region		NAVW16.00,NAVNL6.00.01	14-08-09	12:00:00	
	10	Shipment Method		NAVW16.00	05-11-08	12:00:00	
	13	Salesperson/Purchaser		NAVW16.00.01	14-08-09	12:00:00	
	14	Location		NAVW16.00.01,NAVNL4.00	14-08-09	12:00:00	
	15	G/L Account		NAVW16.00,NAVNL6.00	05-11-08	12:00:00	
	17	G/L Entry		NAVW16.00,NAVNL5.00	05-11-08	12:00:00	
	18	Customer		NAVW16.00.01,NAVNL4.00.0...	11-03-10	13:29:27	
	19	Cust. Invoice Disc.		NAVW16.00	05-11-08	12:00:00	
	21	Cust. Ledger Entry		NAVW16.00.01,NAVNL6.00.01	14-08-09	12:00:00	
	23	Vendor		NAVW16.00,NAVNL4.00.03	04-03-10	15:00:00	

It is also possible to select more objects at the same time and lock them together.
Now you can see that the object gets a green color and that the green square gets a number 1 and the grey square gets the number 0.
The green square shows the number of objects that are locked by you. The grey square indicates the number of objects that are locked by others.

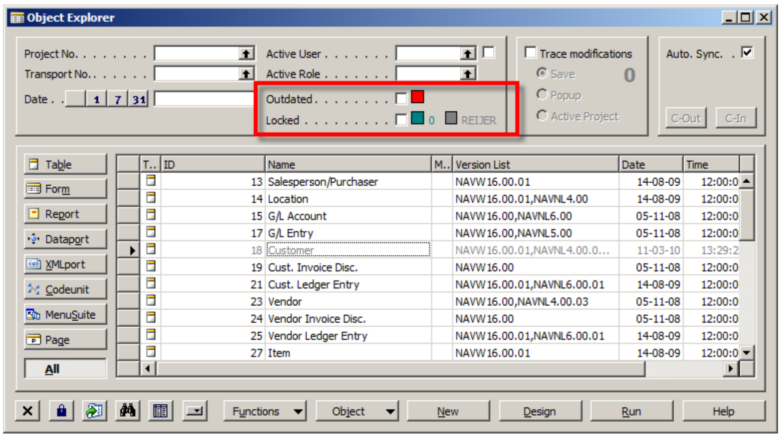
NOTE: There are hidden columns for color blind people which indicate if an object is locked and by who.



Project No. . . . .
Transport No.. . . . . .
Date . .
Active User . . . . ☐
Active Role . . . .
Outdated . . . . ☐ 🟥
Locked . . . . ☐ 🟩 1 ▨ 0
☑ Trace modifications
● Save 0
○ Popup
○ Active Project
Auto. Sync. . ☑

T..	ID	Name	M..	Version List	Date	Time
	8	Language		NAVW16.00.01	14-08-09	12:00:00
	9	Country/Region		NAVW16.00,NAVNL6.00.01	14-08-09	12:00:00
	10	Shipment Method		NAVW16.00	05-11-08	12:00:00
	13	Salesperson/Purchaser		NAVW16.00.01	14-08-09	12:00:00
	14	Location		NAVW16.00.01,NAVNL4.00	14-08-09	12:00:00
	15	G/L Account		NAVW16.00,NAVNL6.00	05-11-08	12:00:00
	17	G/L Entry		NAVW16.00,NAVNL5.00	05-11-08	12:00:00
	18	Customer		LOCKED,NAVW16.00.01,NAV...	11-03-10	13:29:27
	19	Cust. Invoice Disc.		NAVW16.00	05-11-08	12:00:00
	21	Cust. Ledger Entry		NAVW16.00.01,NAVNL6.00.01	14-08-09	12:00:00
	23	Vendor		NAVW16.00,NAVNL4.00.03	04-03-10	15:00:00

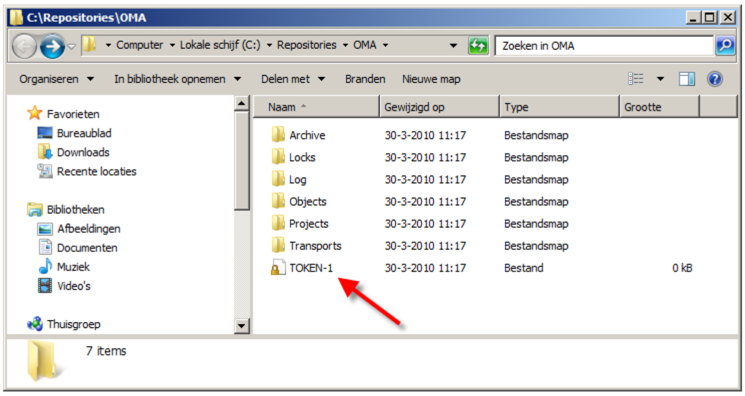
In other databases you can see that the object has a grey color and that the user that has locked the object is shown next to the grey square.



Project No. . . . .
Transport No.. . . . . .
Date . .
Active User . . . . ☐
Active Role . . . .
Outdated . . . . ☐ 🟥
Locked . . . . ☐ 🟩 0 ▨ REIJER
☐ Trace modifications
● Save 0
○ Popup
○ Active Project
Auto. Sync. . ☑

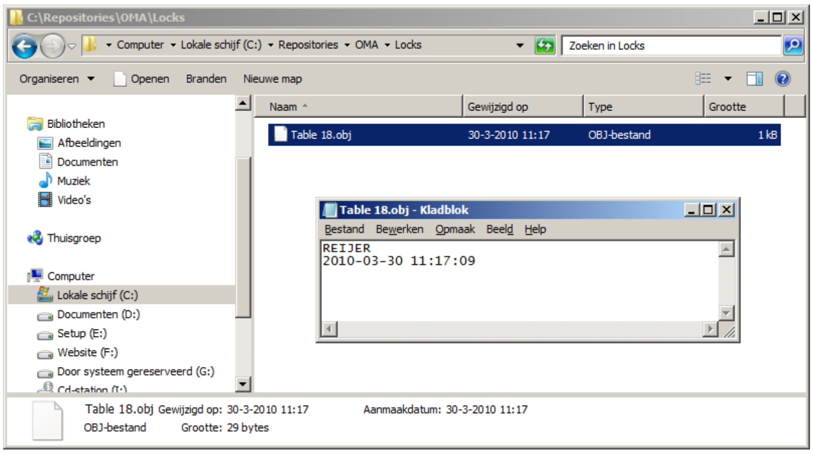
T..	ID	Name	M..	Version List	Date	Time
	13	Salesperson/Purchaser		NAVW16.00.01	14-08-09	12:00:0
	14	Location		NAVW16.00.01,NAVNL4.00	14-08-09	12:00:0
	15	G/L Account		NAVW16.00,NAVNL6.00	05-11-08	12:00:0
	17	G/L Entry		NAVW16.00,NAVNL5.00	05-11-08	12:00:0
	18	Customer		NAVW16.00.01,NAVNL4.00.0...	11-03-10	13:29:2
	19	Cust. Invoice Disc.		NAVW16.00	05-11-08	12:00:0
	21	Cust. Ledger Entry		NAVW16.00.01,NAVNL6.00.01	14-08-09	12:00:0
	23	Vendor		NAVW16.00,NAVNL4.00.03	05-11-08	12:00:0
	24	Vendor Invoice Disc.		NAVW16.00	05-11-08	12:00:0
	25	Vendor Ledger Entry		NAVW16.00.01,NAVNL6.00.01	14-08-09	12:00:0
	27	Item		NAVW16.00.01	14-08-09	12:00:0

What technically happens in the background is that the token has got a new number:



Naam	Gewijzigd op	Type	Grootte
Archive	30-3-2010 11:17	Bestandsmap	
Locks	30-3-2010 11:17	Bestandsmap	
Log	30-3-2010 11:17	Bestandsmap	
Objects	30-3-2010 11:17	Bestandsmap	
Projects	30-3-2010 11:17	Bestandsmap	
Transports	30-3-2010 11:17	Bestandsmap	
TOKEN-1	30-3-2010 11:17	Bestand	0 kB

A lock file is placed in the locked directory with the credentials of the user that has locked the object:

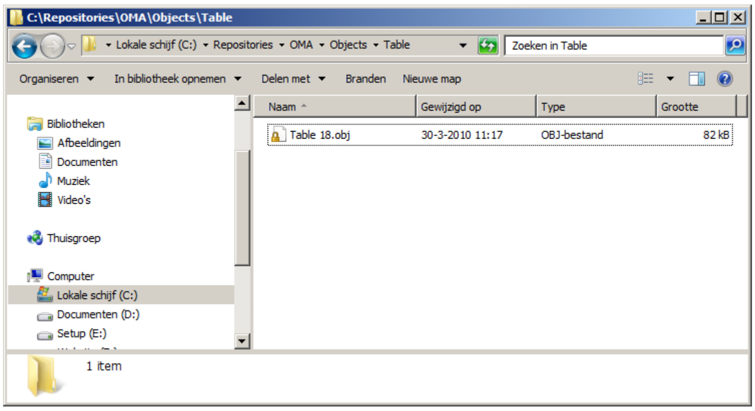


Naam	Gewijzigd op	Type	Grootte
Table 18.obj	30-3-2010 11:17	OBJ-bestand	1 kB

```
REIJER
2010-03-30 11:17:09
```

Table 18.obj Gewijzigd op: 30-3-2010 11:17 Aanmaakdatum: 30-3-2010 11:17
OBJ-bestand Grootte: 29 bytes

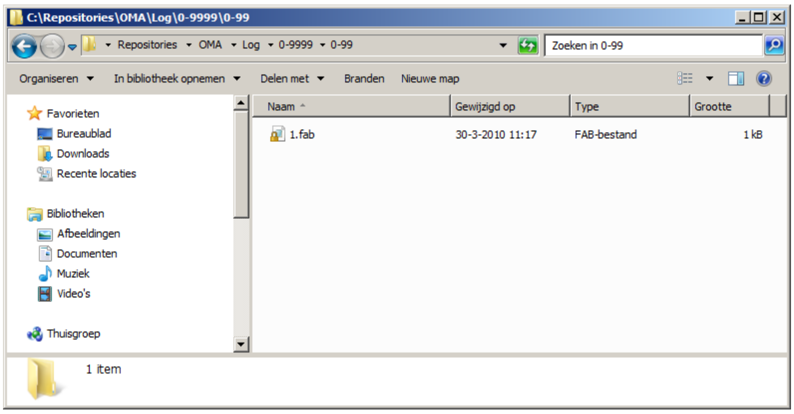
A copy of the locked object is placed in the object directory:



Naam	Gewijzigd op	Type	Grootte
Table 18.obj	30-3-2010 11:17	OBJ-bestand	82 kB

1 item

Every modification that involves the repository is logged in the log directory. This is done through FAB files. All the other databases check the token in the main directory and if there are new log files they are imported and processed.

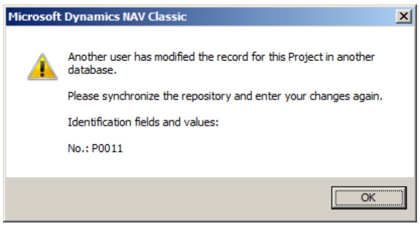


Naam	Gewijzigd op	Type	Grootte
1.fab	30-3-2010 11:17	FAB-bestand	1 kB

1 item

13.4. Modifying Project and Transports

If you change a project or transport this modification is immediately saved to the repository. If two developers are modifying the same project the following error appears and the modification is rolled back.



Another user has modified the record for this Project in another database.

Please synchronize the repository and enter your changes again.

Identification fields and values:

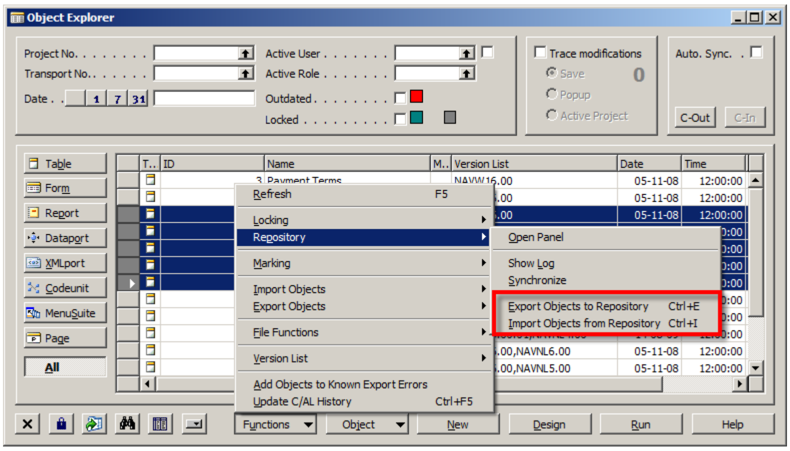
No.: P0011

13.5. Exporting and Importing

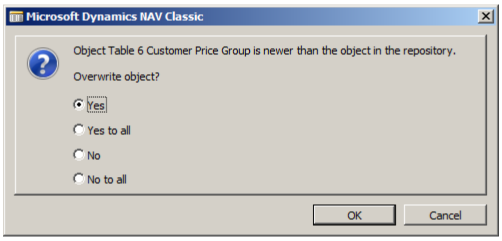
If you export data to the repository it is not placed in the log so it is not automatically read by the other databases.

13.5.1. Exporting and Importing Objects

If you want to ex- or import all objects you can use the functions on the Repository Panel. If you want to ex- or import one object or a couple of objects you can use the repository functions in the Object Explorer.



If you try to import an object that is newer in your database you will be warned.



Object Table 6 Customer Price Group is newer than the object in the repository.

Overwrite object?

● Yes
○ Yes to all
○ No
○ No to all

13.5.2. Exporting and Importing Projects

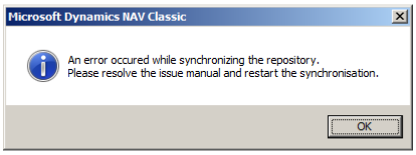
If you want to ex- or import all projects you can use the functions on the Repository Panel. If you want to ex- or import one project or a couple of projects you can use the repository functions on the Project Card or the Project List.

13.5.3. Exporting and Importing Transports

Similar to ex- and importing projects.

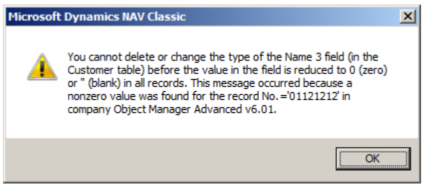
13.6. Place Action in Repository

All the data actions that you do with the Action Worksheet can be placed in the repository so that the action will also be executed in all other databases. If e.g. you want to delete a field from a table you first have to empty that field. If you only empty the field in your own database and remove the field all the users in other databases will get the following error.



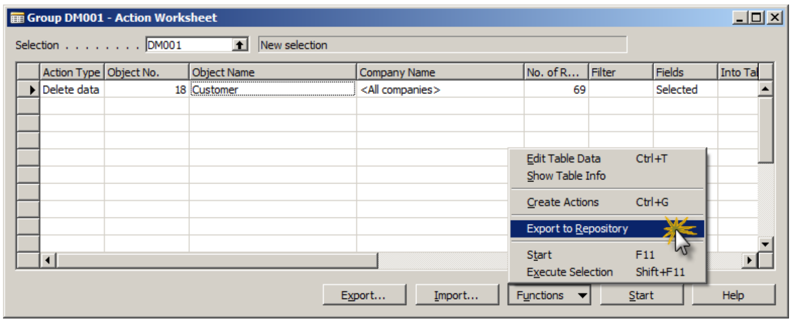
An error occured while synchronizing the repository.
Please resolve the issue manual and restart the synchronisation.

And if they process the log manual they will get the following error.



You cannot delete or change the type of the Name 3 field (in the Customer table) before the value in the field is reduced to 0 (zero) or '' (blank) in all records. This message occurred because a nonzero value was found for the record No.='01121212' in company Object Manager Advanced v6.01.

So it is important that you first export an action to empty the field to the repository before you save the object.

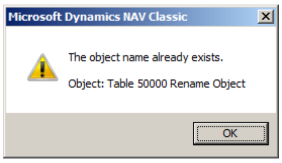


Selection . . . . DM001 New selection

Action Type	Object No.	Object Name	Company Name	No. of R...	Filter	Fields	Into Ta
Delete data	18	Customer	<All companies>	69		Selected	

13.7. Renumber an Object

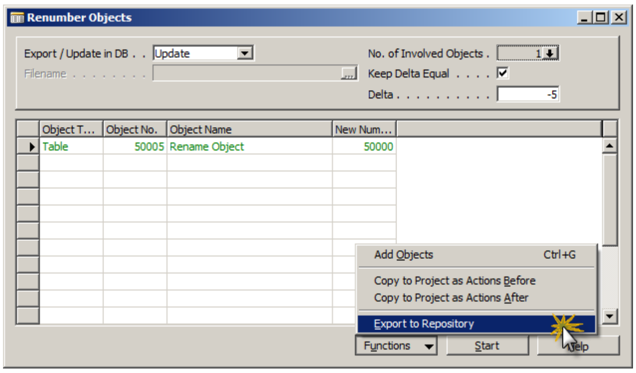
If you simply rename an object in your database this will not be noticed by the other connected databases and they will try to insert the renamed object which will result in two objects with the same name.



The object name already exists.

Object: Table 50000 Rename Object

To prevent this it is recommended that you renumber an object always with the Renumber Objects form and that you copy this renumber action to the repository before checking in your modifications.



Export / Update in DB . . Update

Filename . . . .

No. of Involved Objects . 1

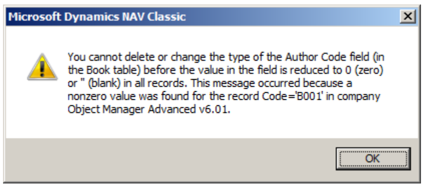
Keep Delta Equal . . . . ☑

Delta . . . . -5

Object T...	Object No.	Object Name	New Num...
Table	50005	Rename Object	50000

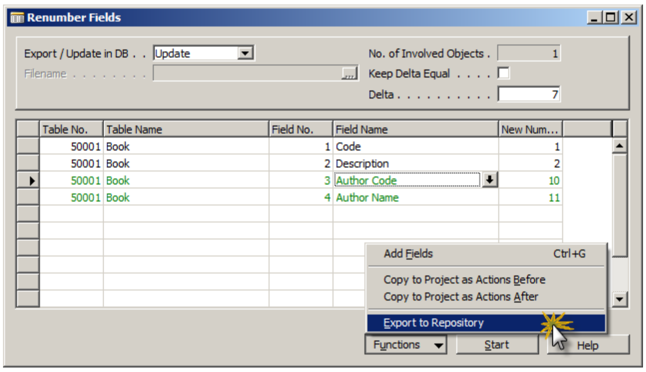
13.8. Renumber a Field

A field in a table cannot be renumbered if there is data present in this field. If you do so you will get an error like this.



You cannot delete or change the type of the Author Code field (in the Book table) before the value in the field is reduced to 0 (zero) or '' (blank) in all records. This message occurred because a nonzero value was found for the record Code='B001' in company Object Manager Advanced v6.01.

If you renumber a field in your own database all the other connected databases will also be confronted with this error. To prevent this you have to use the “Renumber Fields” functionality and copy your renumber actions to the repository.



Export / Update in DB . . Update

Filename . . . .

No. of Involved Objects . 1

Keep Delta Equal . . . . ☐

Delta 7

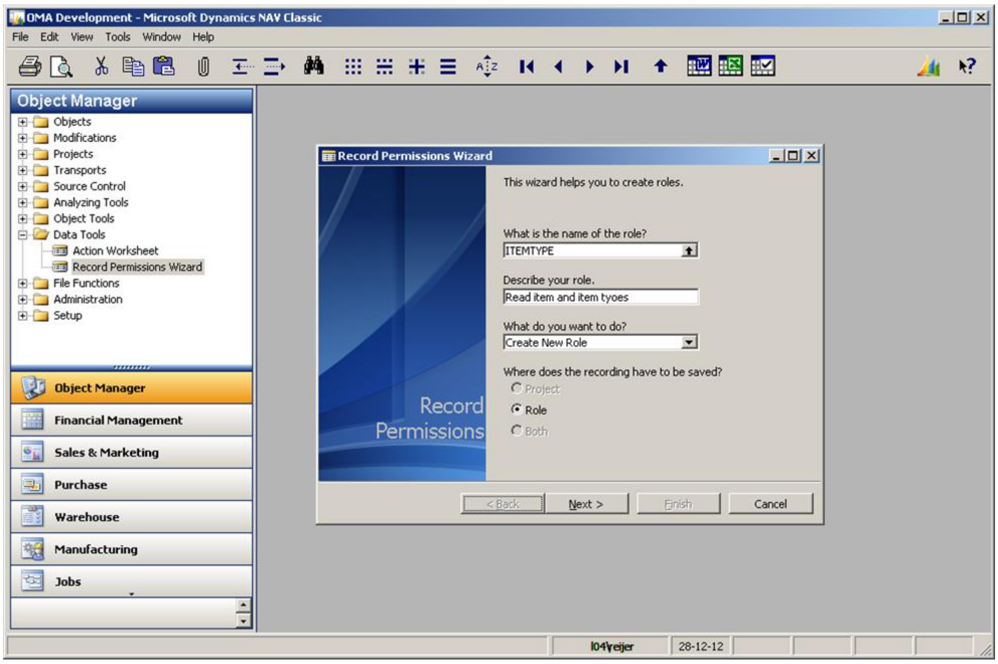
Table No.	Table Name	Field No.	Field Name	New Num...
50001	Book	1	Code	1
50001	Book	2	Description	2
50001	Book	3	Author Code	10
50001	Book	4	Author Name	11

All the other databases will now renumber the field without any problems.

14. Record Permission Wizard

The “Record Permission Wizard” creates or modifies roles. You can add these permissions to a project and or save them to your database.

1. Select Data Tools > Record Permission Wizard to start the wizard
   You can also start the wizard from the “Project Card” by pressing Project – Permissions, Functions – Record Permissions
2. Create the Role Name or select an existing role
3. Describe the role
4. What do you want to do? You have 3 options:
   - Create New Role
   - Modify Existing Role
   - Create New Role Based on Existing



This wizard helps you to create roles.

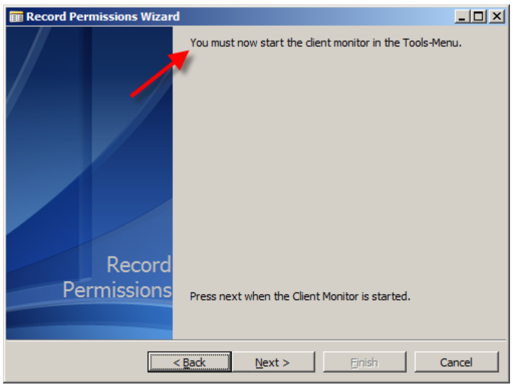
What is the name of the role? ITEMTYPE

Describe your role. Read item and item tyoes

What do you want to do? Create New Role

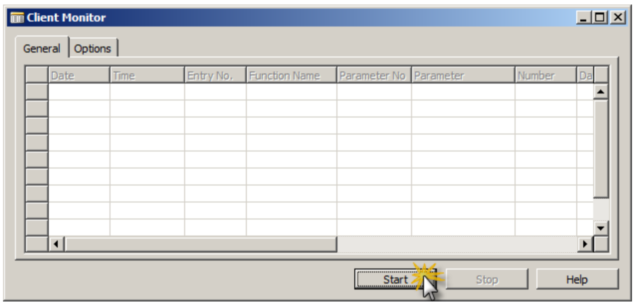
Where does the recording have to be saved?
○ Project
● Role
○ Both

5. Press Next.
6. Start the Client Monitor and press Next.

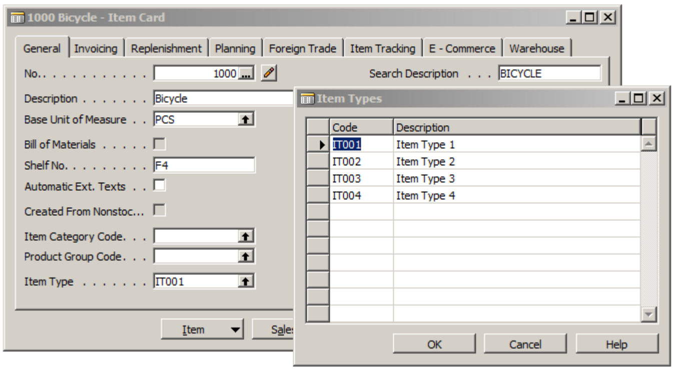


You must now start the client monitor in the Tools-Menu.

Press next when the Client Monitor is started.



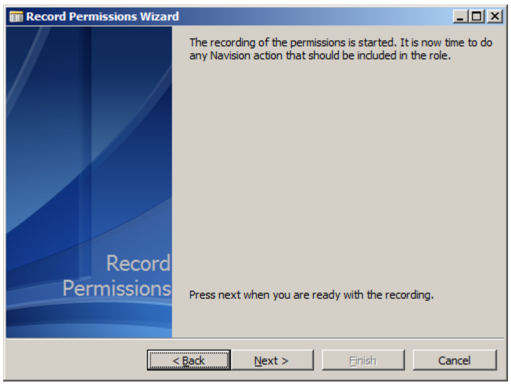
7. Now the recording starts; you perform now every action on the object(s) the user is allowed for the role you created



No.	1000
Description	Bicycle
Base Unit of Measure	PCS
Bill of Materials	☐
Shelf No.	F4
Automatic Ext. Texts	☐
Created From Nonstoc...	☐
Item Category Code.	
Product Group Code.	
Item Type	IT001
Search Description	BICYCLE

Code	Description
IT001	Item Type 1
IT002	Item Type 2
IT003	Item Type 3
IT004	Item Type 4

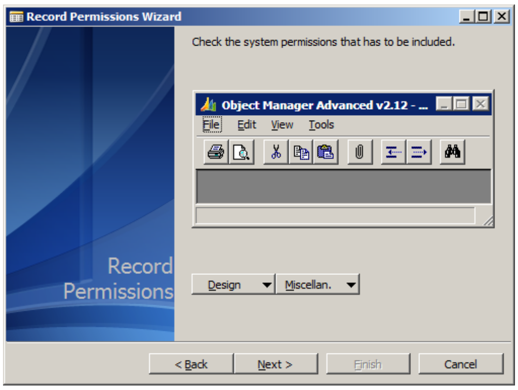
8. When ready recording press Next in the Wizard



The recording of the permissions is started. It is now time to do any Navision action that should be included in the role.

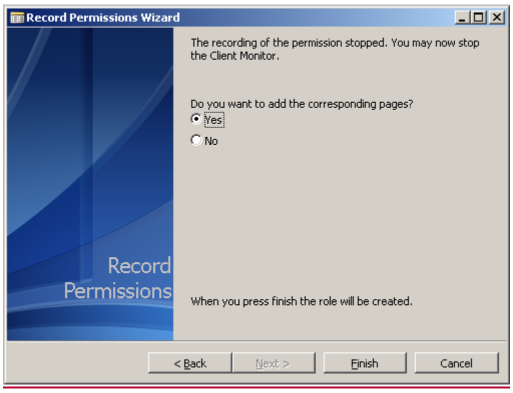
Press next when you are ready with the recording.

9. Check system permissions (if necessary) then Next



Check the system permissions that has to be included.

10. Stop the Client Monitor , indicate whether you want to include the pages corresponding to the forms you have recorded permissions for, and press Next to end the wizard and get the new role created or the existing role modified.



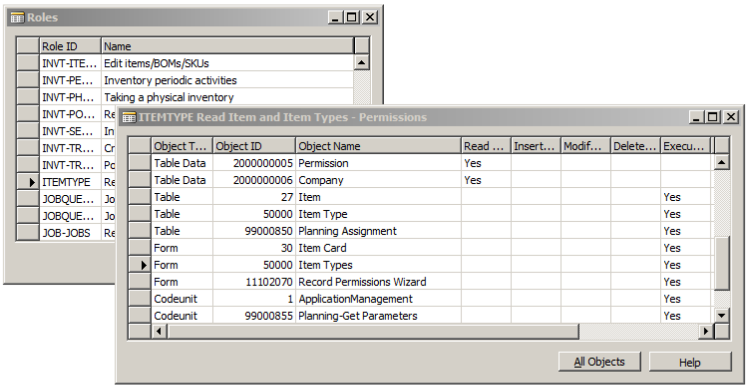
The recording of the permission stopped. You may now stop the Client Monitor.

Do you want to add the corresponding pages?

● Yes

○ No

When you press finish the role will be created.



Role ID	Name
INVT-ITE…	Edit items/BOMs/SKUs
INVT-PE…	Inventory periodic activities
INVT-PH…	Taking a physical inventory
INVT-PO…	Re
INVT-SE…	In
INVT-TR…	Cr
INVT-TR…	Pc
ITEMTYPE	Re
JOBQUE…	Jo
JOBQUE…	Jo
JOB-JOBS	Re

Object T…	Object ID	Object Name	Read …	Insert…	Modif…	Delete…	Execu…
Table Data	2000000005	Permission	Yes				
Table Data	2000000006	Company	Yes				
Table	27	Item					Yes
Table	50000	Item Type					Yes
Table	99000850	Planning Assignment					Yes
Form	30	Item Card					Yes
Form	50000	Item Types					Yes
Form	11102070	Record Permissions Wizard					Yes
Codeunit	1	ApplicationManagement					Yes
Codeunit	99000855	Planning-Get Parameters					Yes

15. Where Used Functionality

With this tool you can check where objects, fields, triggers, etc. are used. It is also possible to see if an object is unused and if it is called without validation.

You can search on the following type of objects:

- Object
- Trigger
- Key
- Sum Index Field
- Field
- Global Function
- Local Function
- Global Variable
- Local Variable
- Parameter
- Return Value

15.1. Setup

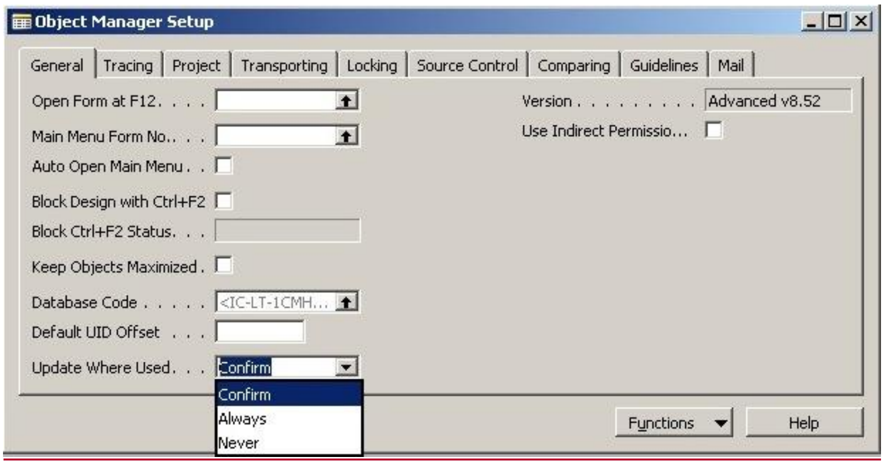
In order to use the Where Used functionality Object Manager needs to update the “Where Used Objects”. This is done in three steps:

1. Updating “C/AL History”
2. Updating “Where Used Objects”. The C/AL code is analyzed and every item that can be used somewhere is saved
3. Updating “Where Used In”. The C/AL code is analyzed and every entry where an item, found in step 2, is used in is saved

Use the “Update Where Used” field on Object Manager Setup window to

1. Let Object Manager ask you whether or not you want the “Where Used Objects” be updated or
2. Let this process be run automatically each time you open the “Where Used Object Card” or
3. Skip this process

Using the *Confirm*, *Always* or *Never* options respectively.



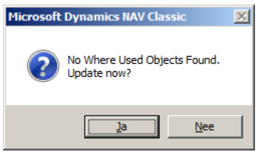
Object Manager Setup

Open Form at F12. . . .
Main Menu Form No.. . .
Auto Open Main Menu . . ☐
Block Design with Ctrl+F2 ☐
Block Ctrl+F2 Status. . .
Keep Objects Maximized . ☐
Database Code . . . . . <IC-LT-1CMH...
Default UID Offset . . .
Update Where Used. . . Confirm
Confirm
Always
Never
Version . . . . Advanced v8.52
Use Indirect Permissio... ☐

NOTE: For each object that changes these three steps will be executed again. So your “Where Used Objects” will always be up to date.

This is a time consuming process. If you cancel the updating process halfway, it will continue, the next time you open the “Where Used Card”, from that point on.

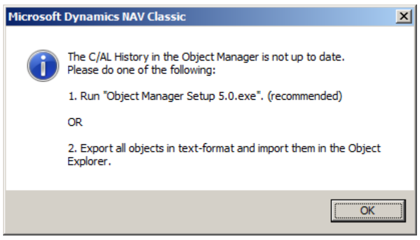
If you have not run the setup (see section 2.1 - Installing) you will have to update the “C/AL History” manually (see section 9.2 - Update C/AL History) otherwise it will be done automatically.



Microsoft Dynamics NAV Classic

No Where Used Objects Found.
Update now?

DLL installed



Microsoft Dynamics NAV Classic

The C/AL History in the Object Manager is not up to date.
Please do one of the following:

1. Run "Object Manager Setup 5.0.exe". (recommended)

OR

2. Export all objects in text-format and import them in the Object Explorer.

No DLL installed

15.2. Find out Where an Object is used

When everything is updated the “Where Used Object Card” opens.

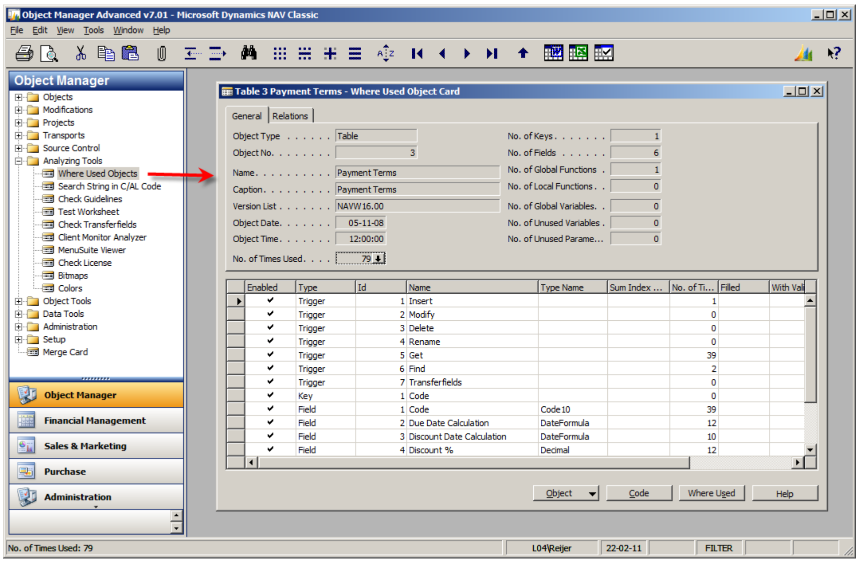


Table 3 Payment Terms - Where Used Object Card

Object Type . . . . . . Table
Object No. 3
Name. . . . . Payment Terms
Caption. . . . . Payment Terms
Version List NAVW16.00
Object Date. 05-11-08
Object Time. 12:00:00
No. of Times Used. . . . 79
No. of Keys. 1
No. of Fields 6
No. of Global Functions . 1
No. of Local Functions . . 0
No. of Global Variables. . 0
No. of Unused Variables . 0
No. of Unused Parame… 0

Enabled	Type	Id	Name	Type Name	Sum Index …	No. of Ti…	Filled	With Vali
✓	Trigger	1	Insert			1		
✓	Trigger	2	Modify			0		
✓	Trigger	3	Delete			0		
✓	Trigger	4	Rename			0		
✓	Trigger	5	Get			39		
✓	Trigger	6	Find			2		
✓	Trigger	7	Transferfields			0		
✓	Key	1	Code			0		
✓	Field	1	Code	Code10		39		
✓	Field	2	Due Date Calculation	DateFormula		12		
✓	Field	3	Discount Date Calculation	DateFormula		10		
✓	Field	4	Discount %	Decimal		12		

Some lines are colored:

- Red: Used but not active
- Green: Not used
- Grey: Not active
- Blue: Global function that is only used local

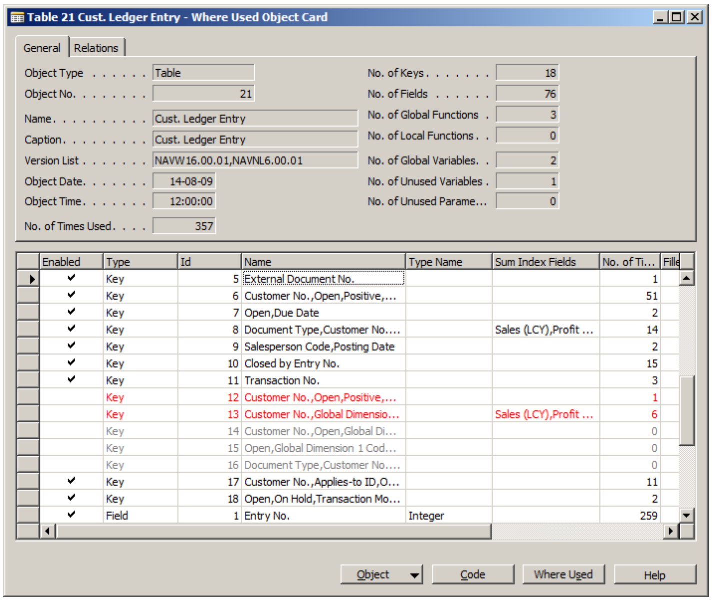


Table 21 Cust. Ledger Entry - Where Used Object Card

Object Type . . . . . . Table
Object No. 21
Name. . . . . Cust. Ledger Entry
Caption. . . . . Cust. Ledger Entry
Version List NAVW16.00.01,NAVNL6.00.01
Object Date. 14-08-09
Object Time. 12:00:00
No. of Times Used. . . . 357
No. of Keys. 18
No. of Fields 76
No. of Global Functions . 3
No. of Local Functions . . 0
No. of Global Variables. . 2
No. of Unused Variables . 1
No. of Unused Parame… 0

Enabled	Type	Id	Name	Type Name	Sum Index Fields	No. of Ti…	Fille
✓	Key	5	External Document No.			1	
✓	Key	6	Customer No.,Open,Positive,…			51	
✓	Key	7	Open,Due Date			2	
✓	Key	8	Document Type,Customer No.…		Sales (LCY),Profit …	14	
✓	Key	9	Salesperson Code,Posting Date			2	
✓	Key	10	Closed by Entry No.			15	
✓	Key	11	Transaction No.			3	
	Key	12	Customer No.,Open,Positive,…			1	
	Key	13	Customer No.,Global Dimensio…		Sales (LCY),Profit …	6	
	Key	14	Customer No.,Open,Global Di…			0	
	Key	15	Open,Global Dimension 1 Cod…			0	
	Key	16	Document Type,Customer No.…			0	
✓	Key	17	Customer No.,Applies-to ID,O…			11	
✓	Key	18	Open,On Hold,Transaction Mo…			2	
✓	Field	1	Entry No.	Integer		259	

To find out where an element is used you press the “Where Used” button. For example the Insert trigger of the “Cust. Ledger Entry” table.

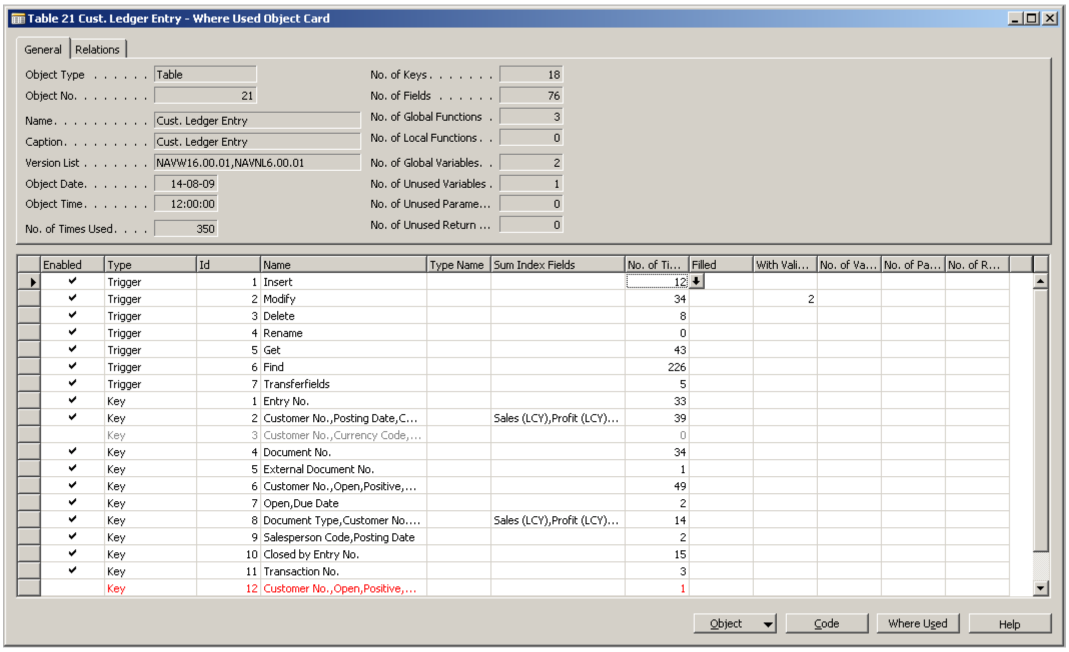
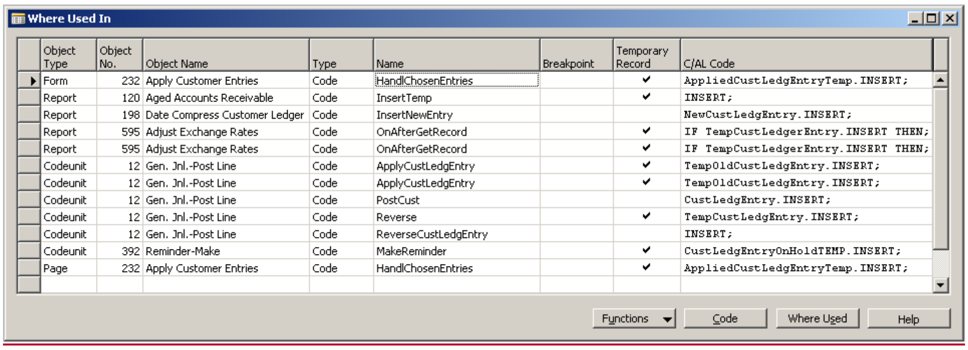


Table 21 Cust. Ledger Entry - Where Used Object Card

Object Type	Table	No. of Keys	18
Object No.	21	No. of Fields	76
Name	Cust. Ledger Entry	No. of Global Functions	3
Caption	Cust. Ledger Entry	No. of Local Functions	0
Version List	NAVW16.00.01,NAVNL6.00.01	No. of Global Variables.	2
Object Date	14-08-09	No. of Unused Variables	1
Object Time	12:00:00	No. of Unused Parame...	0
No. of Times Used	350	No. of Unused Return ...	0

Enabled	Type	Id	Name	Type Name	Sum Index Fields	No. of Ti...	Filled	With Vali...	No. of Va...	No. of Pa...	No. of R...
✓	Trigger	1	Insert			12					
✓	Trigger	2	Modify			34		2			
✓	Trigger	3	Delete			8					
✓	Trigger	4	Rename			0					
✓	Trigger	5	Get			43					
✓	Trigger	6	Find			226					
✓	Trigger	7	Transferfields			5					
✓	Key	1	Entry No.			33					
✓	Key	2	Customer No.,Posting Date,C...		Sales (LCY),Profit (LCY)...	39					
	Key	3	Customer No.,Currency Code,...			0					
✓	Key	4	Document No.			34					
✓	Key	5	External Document No.			1					
✓	Key	6	Customer No.,Open,Positive,...			49					
✓	Key	7	Open,Due Date			2					
✓	Key	8	Document Type,Customer No....		Sales (LCY),Profit (LCY)...	14					
✓	Key	9	Salesperson Code,Posting Date			2					
✓	Key	10	Closed by Entry No.			15					
✓	Key	11	Transaction No.			3					
	Key	12	Customer No.,Open,Positive,...			1					

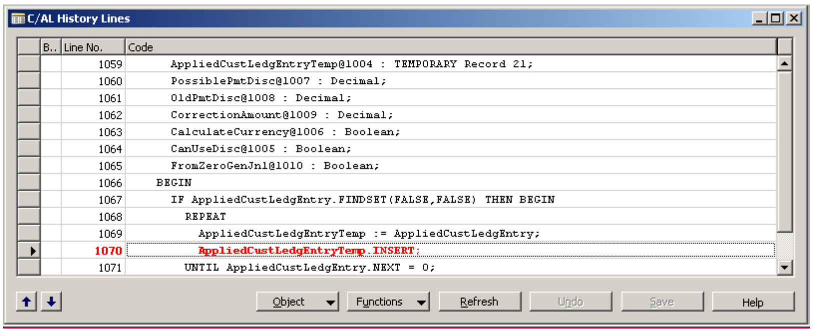
The “Where Used In” window opens. Here you see the different lines of code that are calling the Insert trigger.



Where Used In

Object Type	Object No.	Object Name	Type	Name	Breakpoint	Temporary Record	C/AL Code
Form	232	Apply Customer Entries	Code	HandlChosenEntries		✓	`AppliedCustLedgEntryTemp.INSERT;`
Report	120	Aged Accounts Receivable	Code	InsertTemp		✓	`INSERT;`
Report	198	Date Compress Customer Ledger	Code	InsertNewEntry			`NewCustLedgEntry.INSERT;`
Report	595	Adjust Exchange Rates	Code	OnAfterGetRecord		✓	`IF TempCustLedgerEntry.INSERT THEN;`
Report	595	Adjust Exchange Rates	Code	OnAfterGetRecord		✓	`IF TempCustLedgerEntry.INSERT THEN;`
Codeunit	12	Gen. Jnl.-Post Line	Code	ApplyCustLedgEntry		✓	`TempOldCustLedgEntry.INSERT;`
Codeunit	12	Gen. Jnl.-Post Line	Code	ApplyCustLedgEntry		✓	`TempOldCustLedgEntry.INSERT;`
Codeunit	12	Gen. Jnl.-Post Line	Code	PostCust			`CustLedgEntry.INSERT;`
Codeunit	12	Gen. Jnl.-Post Line	Code	Reverse		✓	`TempCustLedgEntry.INSERT;`
Codeunit	12	Gen. Jnl.-Post Line	Code	ReverseCustLedgEntry			`INSERT;`
Codeunit	392	Reminder-Make	Code	MakeReminder		✓	`CustLedgEntryOnHoldTEMP.INSERT;`
Page	232	Apply Customer Entries	Code	HandlChosenEntries		✓	`AppliedCustLedgEntryTemp.INSERT;`

Once you have selected a line press Code to get a view on the full C/Al code context with the specific line where the trigger is used highlighted.

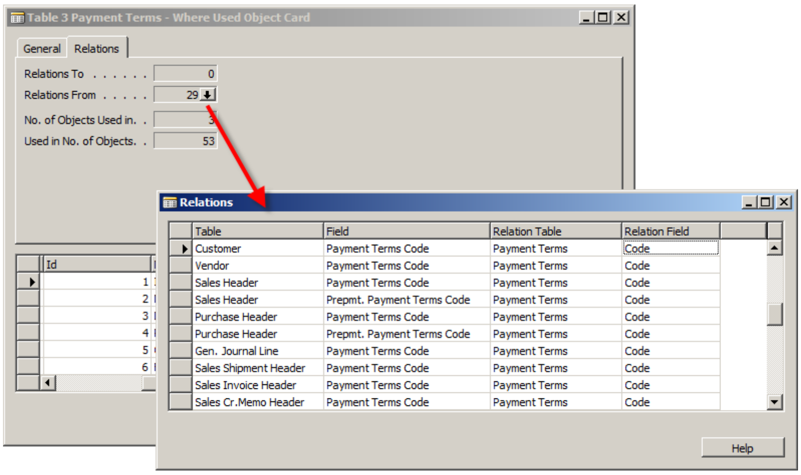


```
      AppliedCustLedgEntryTemp@1004 : TEMPORARY Record 21;
      PossiblePmtDisc@1007 : Decimal;
      OldPmtDisc@1008 : Decimal;
      CorrectionAmount@1009 : Decimal;
      CalculateCurrency@1006 : Boolean;
      CanUseDisc@1005 : Boolean;
      FromZeroGenJnl@1010 : Boolean;
    BEGIN
      IF AppliedCustLedgEntry.FINDSET(FALSE,FALSE) THEN BEGIN
        REPEAT
          AppliedCustLedgEntryTemp := AppliedCustLedgEntry;
          AppliedCustLedgEntryTemp.INSERT;
        UNTIL AppliedCustLedgEntry.NEXT = 0;
```

The “Temporary Record” column in the “Where Used In” window indicates that the C/AL statement uses a temporary record. In our example the call to the Insert trigger is done through the temporary record AppliedCustLedgEntryTemp.

15.3. Relations

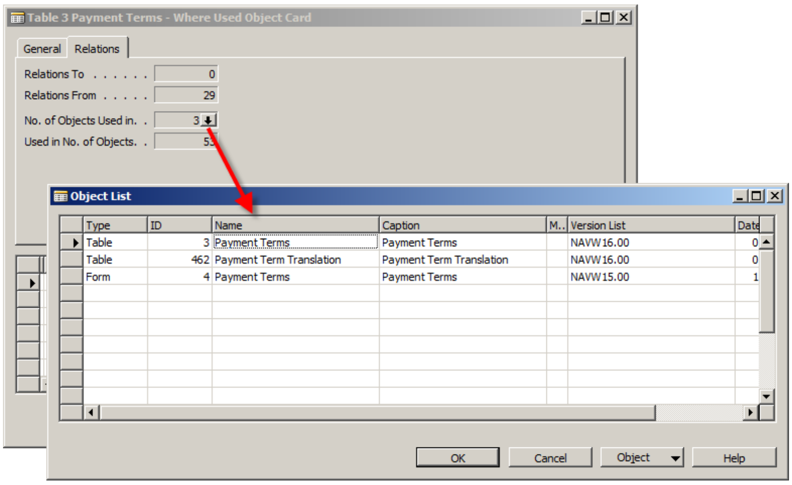
On tab relations you can see the relations between tables.



Relations To 0
Relations From 29
No. of Objects Used in. . 3
Used in No. of Objects. . 53

Table	Field	Relation Table	Relation Field
Customer	Payment Terms Code	Payment Terms	Code
Vendor	Payment Terms Code	Payment Terms	Code
Sales Header	Payment Terms Code	Payment Terms	Code
Sales Header	Prepmt. Payment Terms Code	Payment Terms	Code
Purchase Header	Payment Terms Code	Payment Terms	Code
Purchase Header	Prepmt. Payment Terms Code	Payment Terms	Code
Gen. Journal Line	Payment Terms Code	Payment Terms	Code
Sales Shipment Header	Payment Terms Code	Payment Terms	Code
Sales Invoice Header	Payment Terms Code	Payment Terms	Code
Sales Cr.Memo Header	Payment Terms Code	Payment Terms	Code

It is also possible to see which objects are used in the object and in which objects the object is used.

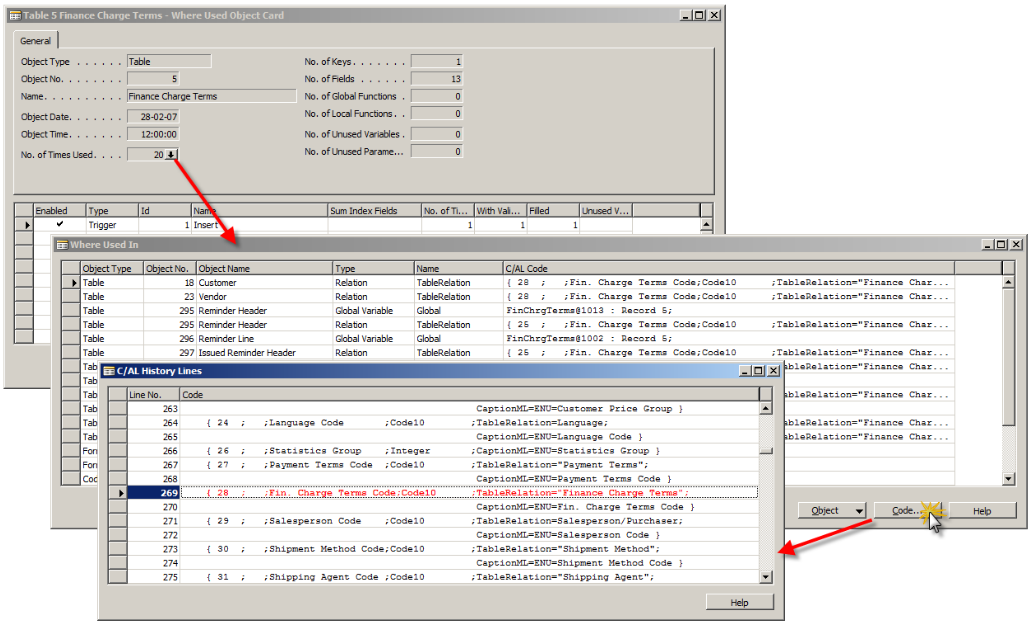


Relations To 0
Relations From 29
No. of Objects Used in. . 3
Used in No. of Objects. . 53

Type	ID	Name	Caption	M..	Version List	Date
Table	3	Payment Terms	Payment Terms		NAVW16.00	0
Table	462	Payment Term Translation	Payment Term Translation		NAVW16.00	0
Form	4	Payment Terms	Payment Terms		NAVW15.00	1

15.4. Examples

Find out where the Table “Finance Charge Terms” is used.



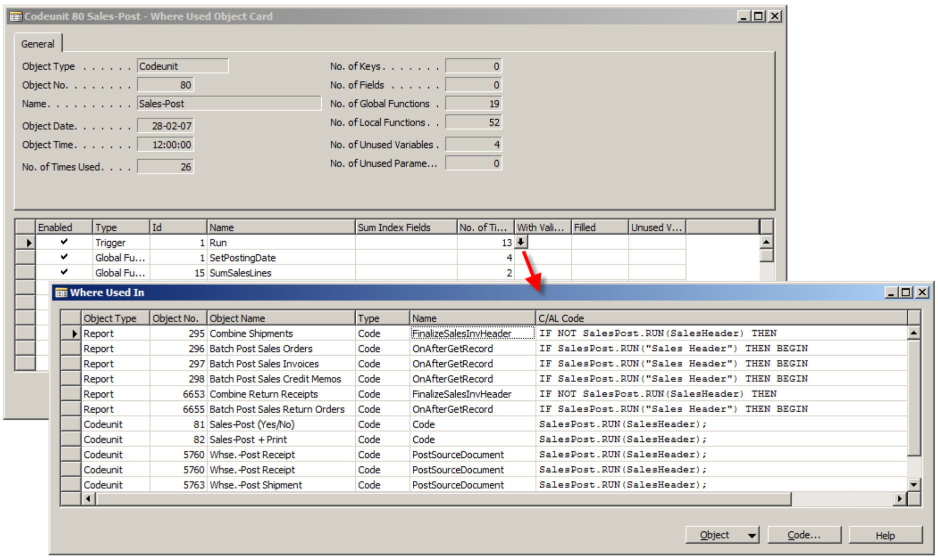
Object Type . . . . . . Table
Object No. 5
Name. . . . . Finance Charge Terms
Object Date. 28-02-07
Object Time. 12:00:00
No. of Times Used. . . . 20
No. of Keys. 1
No. of Fields 13
No. of Global Functions . 0
No. of Local Functions. . 0
No. of Unused Variables . 0
No. of Unused Parame... 0

Enabled	Type	Id	Name	Sum Index Fields	No. of Ti...	With Vali...	Filled	Unused V...
✓	Trigger	1	Insert		1	1	1	

Object Type	Object No.	Object Name	Type	Name	C/AL Code
Table	18	Customer	Relation	TableRelation	`{ 28   ;   ;Fin. Charge Terms Code;Code10       ;TableRelation="Finance Char...`
Table	23	Vendor	Relation	TableRelation	`{ 28   ;   ;Fin. Charge Terms Code;Code10       ;TableRelation="Finance Char...`
Table	295	Reminder Header	Global Variable	Global	`FinChrgTerms@1013 : Record 5;`
Table	295	Reminder Header	Relation	TableRelation	`{ 25   ;   ;Fin. Charge Terms Code;Code10       ;TableRelation="Finance Char...`
Table	296	Reminder Line	Global Variable	Global	`FinChrgTerms@1002 : Record 5;`
Table	297	Issued Reminder Header	Relation	TableRelation	`{ 25   ;   ;Fin. Charge Terms Code;Code10       ;TableRelation="Finance Char...`

```
Line No. Code
263                                                  CaptionML=ENU=Customer Price Group }
264        { 24   ;   ;Language Code       ;Code10        ;TableRelation=Language;
265                                                  CaptionML=ENU=Language Code }
266        { 26   ;   ;Statistics Group    ;Integer       ;CaptionML=ENU=Statistics Group }
267        { 27   ;   ;Payment Terms Code  ;Code10        ;TableRelation="Payment Terms";
268                                                  CaptionML=ENU=Payment Terms Code }
269        { 28   ;   ;Fin. Charge Terms Code;Code10      ;TableRelation="Finance Charge Terms";
270                                                  CaptionML=ENU=Fin. Charge Terms Code }
271        { 29   ;   ;Salesperson Code    ;Code10        ;TableRelation=Salesperson/Purchaser;
272                                                  CaptionML=ENU=Salesperson Code }
273        { 30   ;   ;Shipment Method Code;Code10        ;TableRelation="Shipment Method";
274                                                  CaptionML=ENU=Shipment Method Code }
275        { 31   ;   ;Shipping Agent Code ;Code10        ;TableRelation="Shipping Agent";
```

Find out where codeunit “Sales Post” is called.



Codeunit 80 Sales-Post - Where Used Object Card

Object Type	Codeunit	No. of Keys	0
Object No.	80	No. of Fields	0
Name	Sales-Post	No. of Global Functions	19
Object Date	28-02-07	No. of Local Functions	52
Object Time	12:00:00	No. of Unused Variables	4
No. of Times Used	26	No. of Unused Parame…	0

Enabled	Type	Id	Name	Sum Index Fields	No. of Ti…	With Vali…	Filled	Unused V…
✓	Trigger	1	Run		13			
✓	Global Fu…	1	SetPostingDate		4			
✓	Global Fu…	15	SumSalesLines		2			

Where Used In

Object Type	Object No.	Object Name	Type	Name	C/AL Code
Report	295	Combine Shipments	Code	FinalizeSalesInvHeader	`IF NOT SalesPost.RUN(SalesHeader) THEN`
Report	296	Batch Post Sales Orders	Code	OnAfterGetRecord	`IF SalesPost.RUN("Sales Header") THEN BEGIN`
Report	297	Batch Post Sales Invoices	Code	OnAfterGetRecord	`IF SalesPost.RUN("Sales Header") THEN BEGIN`
Report	298	Batch Post Sales Credit Memos	Code	OnAfterGetRecord	`IF SalesPost.RUN("Sales Header") THEN BEGIN`
Report	6653	Combine Return Receipts	Code	FinalizeSalesInvHeader	`IF NOT SalesPost.RUN(SalesHeader) THEN`
Report	6655	Batch Post Sales Return Orders	Code	OnAfterGetRecord	`IF SalesPost.RUN("Sales Header") THEN BEGIN`
Codeunit	81	Sales-Post (Yes/No)	Code	Code	`SalesPost.RUN(SalesHeader);`
Codeunit	82	Sales-Post + Print	Code	Code	`SalesPost.RUN(SalesHeader);`
Codeunit	5760	Whse.-Post Receipt	Code	PostSourceDocument	`SalesPost.RUN(SalesHeader);`
Codeunit	5760	Whse.-Post Receipt	Code	PostSourceDocument	`SalesPost.RUN(SalesHeader);`
Codeunit	5763	Whse.-Post Shipment	Code	PostSourceDocument	`SalesPost.RUN(SalesHeader);`

Find out where the table “Sales Header” is deleted.

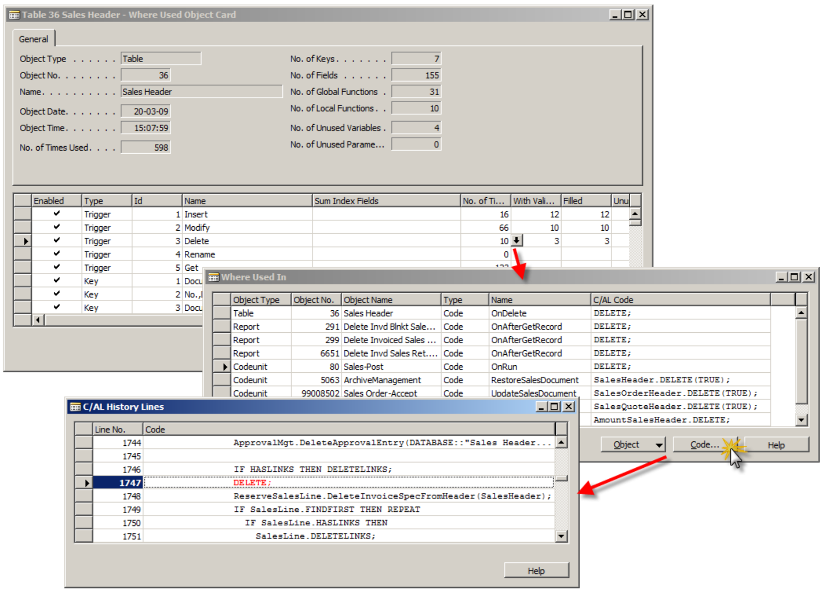


Table 36 Sales Header - Where Used Object Card

Object Type	Table	No. of Keys	7
Object No.	36	No. of Fields	155
Name	Sales Header	No. of Global Functions	31
Object Date	20-03-09	No. of Local Functions	10
Object Time	15:07:59	No. of Unused Variables	4
No. of Times Used	598	No. of Unused Parame…	0

Enabled	Type	Id	Name	Sum Index Fields	No. of Ti…	With Vali…	Filled	Unu
✓	Trigger	1	Insert		16	12	12	
✓	Trigger	2	Modify		66	10	10	
✓	Trigger	3	Delete		10	3	3	
✓	Trigger	4	Rename		0			
✓	Trigger	5	Get					
✓	Key	1	Docu					
✓	Key	2	No.,I					
✓	Key	3	Docu					

Where Used In

Object Type	Object No.	Object Name	Type	Name	C/AL Code
Table	36	Sales Header	Code	OnDelete	`DELETE;`
Report	291	Delete Invd Blnkt Sale…	Code	OnAfterGetRecord	`DELETE;`
Report	299	Delete Invoiced Sales …	Code	OnAfterGetRecord	`DELETE;`
Report	6651	Delete Invd Sales Ret.…	Code	OnAfterGetRecord	`DELETE;`
Codeunit	80	Sales-Post	Code	OnRun	`DELETE;`
Codeunit	5063	ArchiveManagement	Code	RestoreSalesDocument	`SalesHeader.DELETE(TRUE);`
Codeunit	99008502	Sales Order-Accept	Code	UpdateSalesDocument	`SalesOrderHeader.DELETE(TRUE);`
					`SalesQuoteHeader.DELETE(TRUE);`
					`AmountSalesHeader.DELETE;`

C/AL History Lines

Line No.	Code
1744	`ApprovalMgt.DeleteApprovalEntry(DATABASE::"Sales Header...`
1745	
1746	`IF HASLINKS THEN DELETELINKS;`
1747	`DELETE;`
1748	`ReserveSalesLine.DeleteInvoiceSpecFromHeader(SalesHeader);`
1749	`IF SalesLine.FINDFIRST THEN REPEAT`
1750	`  IF SalesLine.HASLINKS THEN`
1751	`    SalesLine.DELETELINKS;`

Find out where the field “Ship-to Code” of table “Sales Header” is filled without validation.

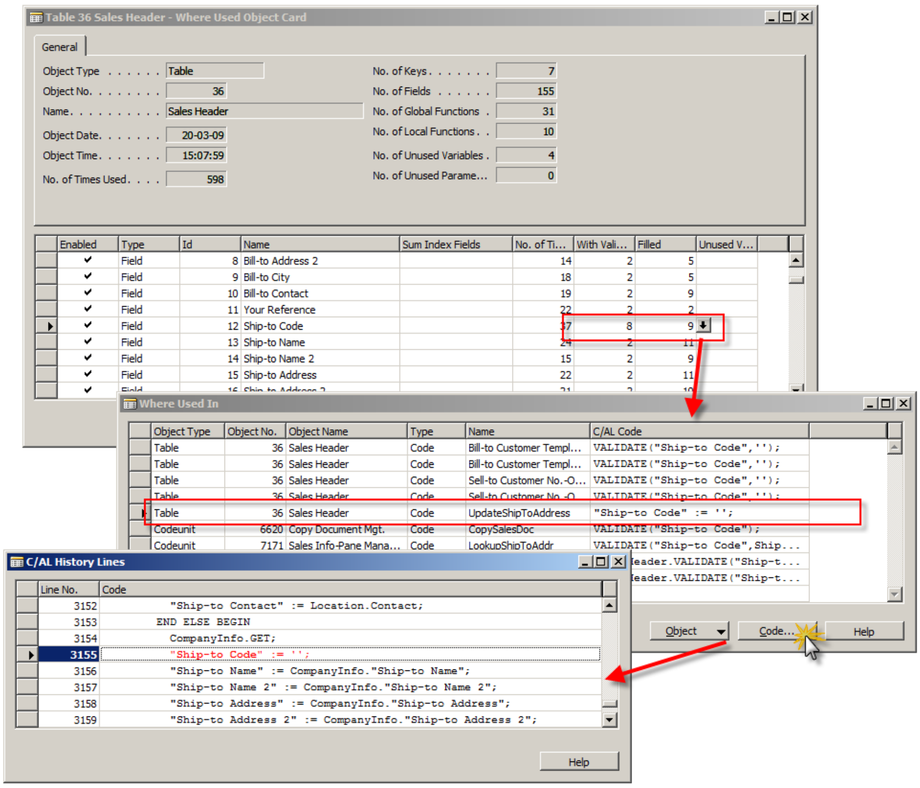


Table 36 Sales Header - Where Used Object Card

General

Object Type: Table
Object No.: 36
Name: Sales Header
Object Date: 20-03-09
Object Time: 15:07:59
No. of Times Used: 598
No. of Keys: 7
No. of Fields: 155
No. of Global Functions: 31
No. of Local Functions: 10
No. of Unused Variables: 4
No. of Unused Parame...: 0

Enabled	Type	Id	Name	Sum Index Fields	No. of Ti...	With Vali...	Filled	Unused V...
✓	Field	8	Bill-to Address 2		14	2	5	
✓	Field	9	Bill-to City		18	2	5	
✓	Field	10	Bill-to Contact		19	2	9	
✓	Field	11	Your Reference		22	2	2	
✓	Field	12	Ship-to Code		37	8	9	
✓	Field	13	Ship-to Name		24	2	11	
✓	Field	14	Ship-to Name 2		15	2	9	
✓	Field	15	Ship-to Address		22	2	11	

Where Used In

Object Type	Object No.	Object Name	Type	Name	C/AL Code
Table	36	Sales Header	Code	Bill-to Customer Templ...	`VALIDATE("Ship-to Code",'');`
Table	36	Sales Header	Code	Bill-to Customer Templ...	`VALIDATE("Ship-to Code",'');`
Table	36	Sales Header	Code	Sell-to Customer No.-O...	`VALIDATE("Ship-to Code",'');`
Table	36	Sales Header	Code	Sell-to Customer No.-O...	`VALIDATE("Ship-to Code",'');`
Table	36	Sales Header	Code	UpdateShipToAddress	`"Ship-to Code" := '';`
Codeunit	6620	Copy Document Mgt.	Code	CopySalesDoc	`VALIDATE("Ship-to Code");`
Codeunit	7171	Sales Info-Pane Mana...	Code	LookupShipToAddr	`VALIDATE("Ship-to Code",Ship...`
					`eader.VALIDATE("Ship-t...`
					`eader.VALIDATE("Ship-t...`

C/AL History Lines

Line No.	Code
3152	`"Ship-to Contact" := Location.Contact;`
3153	`END ELSE BEGIN`
3154	`CompanyInfo.GET;`
3155	`"Ship-to Code" := '';`
3156	`"Ship-to Name" := CompanyInfo."Ship-to Name";`
3157	`"Ship-to Name 2" := CompanyInfo."Ship-to Name 2";`
3158	`"Ship-to Address" := CompanyInfo."Ship-to Address";`
3159	`"Ship-to Address 2" := CompanyInfo."Ship-to Address 2";`

Find out where the key “Sell-to Customer No.”, “External Document No.” of the table “Sales Header” is used.

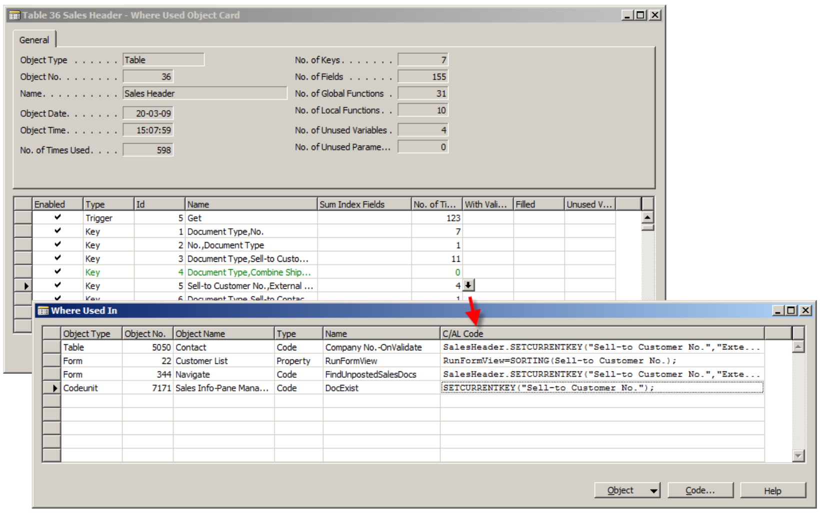


Table 36 Sales Header - Where Used Object Card

General

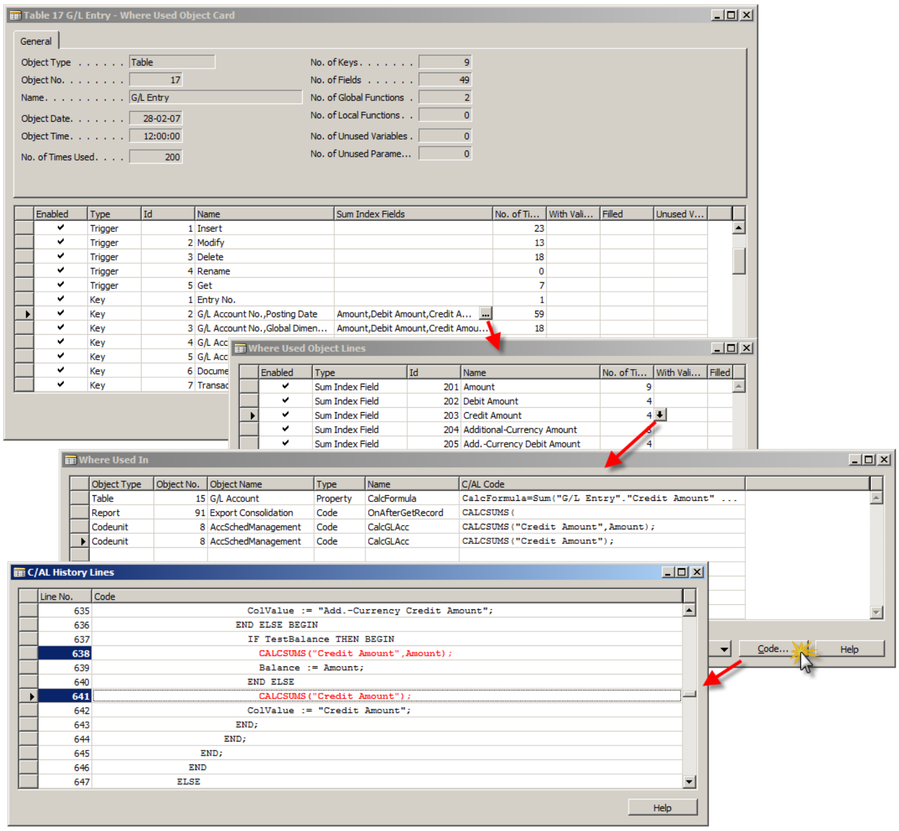
Object Type: Table
Object No.: 36
Name: Sales Header
Object Date: 20-03-09
Object Time: 15:07:59
No. of Times Used: 598
No. of Keys: 7
No. of Fields: 155
No. of Global Functions: 31
No. of Local Functions: 10
No. of Unused Variables: 4
No. of Unused Parame...: 0

Enabled	Type	Id	Name	Sum Index Fields	No. of Ti...	With Vali...	Filled	Unused V...
✓	Trigger	5	Get		123			
✓	Key	1	Document Type,No.		7			
✓	Key	2	No.,Document Type		1			
✓	Key	3	Document Type,Sell-to Custo...		11			
✓	Key	4	Document Type,Combine Ship...		0			
✓	Key	5	Sell-to Customer No.,External ...		4			

Where Used In

Object Type	Object No.	Object Name	Type	Name	C/AL Code
Table	5050	Contact	Code	Company No.-OnValidate	`SalesHeader.SETCURRENTKEY("Sell-to Customer No.","Exte...`
Form	22	Customer List	Property	RunFormView	`RunFormView=SORTING(Sell-to Customer No.);`
Form	344	Navigate	Code	FindUnpostedSalesDocs	`SalesHeader.SETCURRENTKEY("Sell-to Customer No.","Exte...`
Codeunit	7171	Sales Info-Pane Mana...	Code	DocExist	`SETCURRENTKEY("Sell-to Customer No.");`

Find out where the SIF Field “Credit Amount” of the table “G/L Entry” is used.



Object Type . . . . . . Table
Object No. 17
Name. . . . . G/L Entry
Object Date. 28-02-07
Object Time. 12:00:00
No. of Times Used. . . . 200
No. of Keys. 9
No. of Fields 49
No. of Global Functions . 2
No. of Local Functions . . 0
No. of Unused Variables . 0
No. of Unused Parame... 0

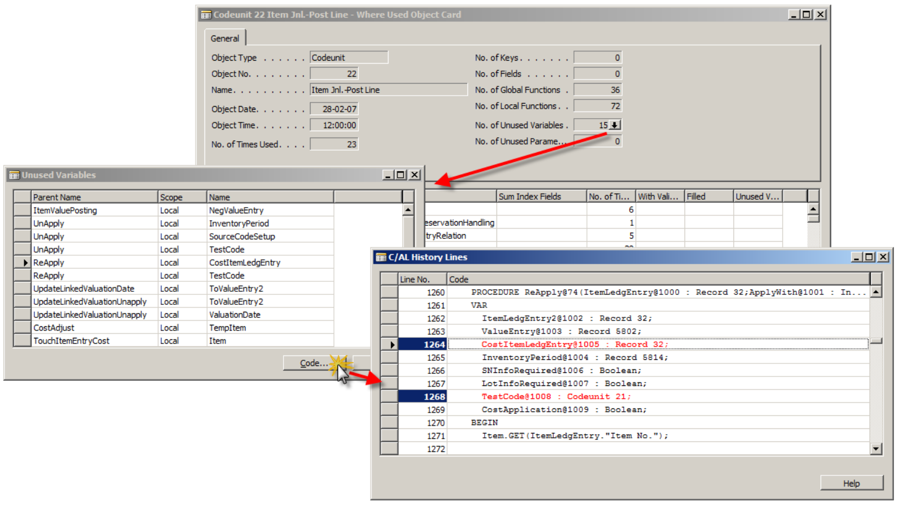
Enabled	Type	Id	Name	Sum Index Fields	No. of Ti...	With Vali...	Filled	Unused V...
✓	Trigger	1	Insert		23			
✓	Trigger	2	Modify		13			
✓	Trigger	3	Delete		18			
✓	Trigger	4	Rename		0			
✓	Trigger	5	Get		7			
✓	Key	1	Entry No.		1			
✓	Key	2	G/L Account No.,Posting Date	Amount,Debit Amount,Credit A...	59			
✓	Key	3	G/L Account No.,Global Dimen...	Amount,Debit Amount,Credit Amou...	18			
✓	Key	4	G/L Acc					
✓	Key	5	G/L Acc					
✓	Key	6	Docume					
✓	Key	7	Transac					

Enabled	Type	Id	Name	No. of Ti...	With Vali...	Filled
✓	Sum Index Field	201	Amount	9		
✓	Sum Index Field	202	Debit Amount	4		
✓	Sum Index Field	203	Credit Amount	4		
✓	Sum Index Field	204	Additional-Currency Amount			
✓	Sum Index Field	205	Add.-Currency Debit Amount	4		

Object Type	Object No.	Object Name	Type	Name	C/AL Code
Table	15	G/L Account	Property	CalcFormula	`CalcFormula=Sum("G/L Entry"."Credit Amount" ...`
Report	91	Export Consolidation	Code	OnAfterGetRecord	`CALCSUMS (`
Codeunit	8	AccSchedManagement	Code	CalcGLAcc	`CALCSUMS("Credit Amount",Amount);`
Codeunit	8	AccSchedManagement	Code	CalcGLAcc	`CALCSUMS("Credit Amount");`

```
            ColValue := "Add.-Currency Credit Amount";
          END ELSE BEGIN
            IF TestBalance THEN BEGIN
              CALCSUMS("Credit Amount",Amount);
              Balance := Amount;
            END ELSE
              CALCSUMS("Credit Amount");
            ColValue := "Credit Amount";
          END;
        END;
      END;
    END
  ELSE
```

Find out how much unused variables codeunit “Item Jnl.-Post Line” has.



Object Type . . . . . . Codeunit
Object No. 22
Name. . . . . Item Jnl.-Post Line
Object Date. 28-02-07
Object Time. 12:00:00
No. of Times Used. . . . 23
No. of Keys. 0
No. of Fields 0
No. of Global Functions . 36
No. of Local Functions . . 72
No. of Unused Variables . 15
No. of Unused Parame... 0

	Sum Index Fields	No. of Ti...	With Vali...	Filled	Unused V...
		6			
eservationHandling		1			
tryRelation		5			

Parent Name	Scope	Name
ItemValuePosting	Local	NegValueEntry
UnApply	Local	InventoryPeriod
UnApply	Local	SourceCodeSetup
UnApply	Local	TestCode
ReApply	Local	CostItemLedgEntry
ReApply	Local	TestCode
UpdateLinkedValuationDate	Local	ToValueEntry2
UpdateLinkedValuationUnapply	Local	ToValueEntry2
UpdateLinkedValuationUnapply	Local	ValuationDate
CostAdjust	Local	TempItem
TouchItemEntryCost	Local	Item

```
    PROCEDURE ReApply@74(ItemLedgEntry@1000 : Record 32;ApplyWith@1001 : In...
    VAR
      ItemLedgEntry2@1002 : Record 32;
      ValueEntry@1003 : Record 5802;
      CostItemLedgEntry@1005 : Record 32;
      InventoryPeriod@1004 : Record 5814;
      SNInfoRequired@1006 : Boolean;
      LotInfoRequired@1007 : Boolean;
      TestCode@1008 : Codeunit 21;
      CostApplication@1009 : Boolean;
    BEGIN
      Item.GET(ItemLedgEntry."Item No.");

```

Find out which keys of Table “Cust. Ledger Entry” are disabled but used.

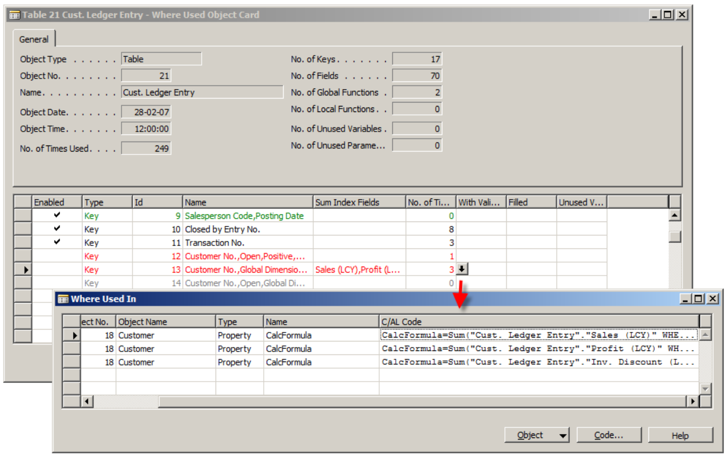


Table 21 Cust. Ledger Entry - Where Used Object Card

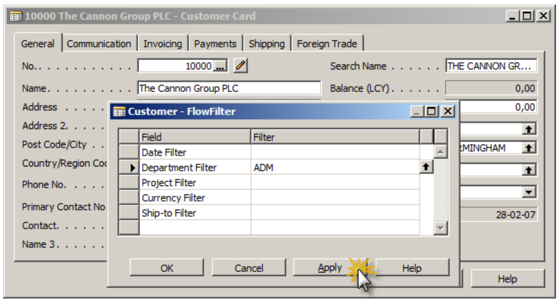
Object Type	Table
Object No.	21
Name	Cust. Ledger Entry
Object Date	28-02-07
Object Time	12:00:00
No. of Times Used	249
No. of Keys	17
No. of Fields	70
No. of Global Functions	2
No. of Local Functions	0
No. of Unused Variables	0
No. of Unused Parame...	0

Enabled	Type	Id	Name	Sum Index Fields	No. of Ti...	With Vali...	Filled	Unused V...
✓	Key	9	Salesperson Code,Posting Date		0			
✓	Key	10	Closed by Entry No.		8			
✓	Key	11	Transaction No.		3			
	Key	12	Customer No.,Open,Positive,...		1			
	Key	13	Customer No.,Global Dimensio...	Sales (LCY),Profit (L...	3			
	Key	14	Customer No.,Open,Global Di...		0			

Where Used In

ect No.	Object Name	Type	Name	C/AL Code
18	Customer	Property	CalcFormula	`CalcFormula=Sum("Cust. Ledger Entry"."Sales (LCY)" WHE...`
18	Customer	Property	CalcFormula	`CalcFormula=Sum("Cust. Ledger Entry"."Profit (LCY)" WH...`
18	Customer	Property	CalcFormula	`CalcFormula=Sum("Cust. Ledger Entry"."Inv. Discount (L...`

To check if this is really the case you can set a filter on “Department Code” in the Customer table and you get an error.

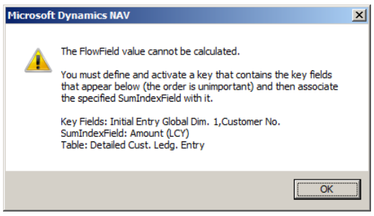


10000 The Cannon Group PLC - Customer Card

No.	10000
Name	The Cannon Group PLC
Search Name	THE CANNON GR...
Balance (LCY)	0,00

Customer - FlowFilter

Field	Filter
Date Filter	
Department Filter	ADM
Project Filter	
Currency Filter	
Ship-to Filter	



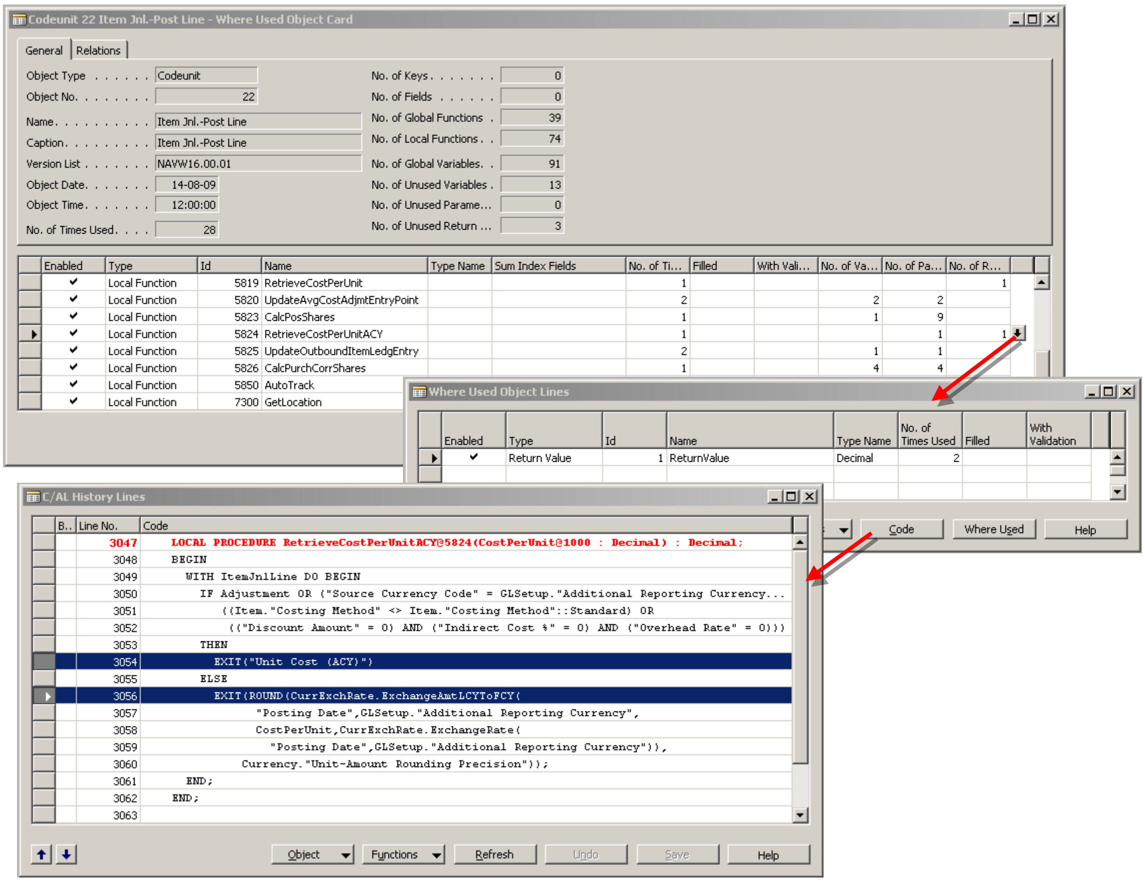
Microsoft Dynamics NAV

The FlowField value cannot be calculated.

You must define and activate a key that contains the key fields that appear below (the order is unimportant) and then associate the specified SumIndexField with it.

Key Fields: Initial Entry Global Dim. 1,Customer No.
SumIndexField: Amount (LCY)
Table: Detailed Cust. Ledg. Entry

Determine the different return values of a function.



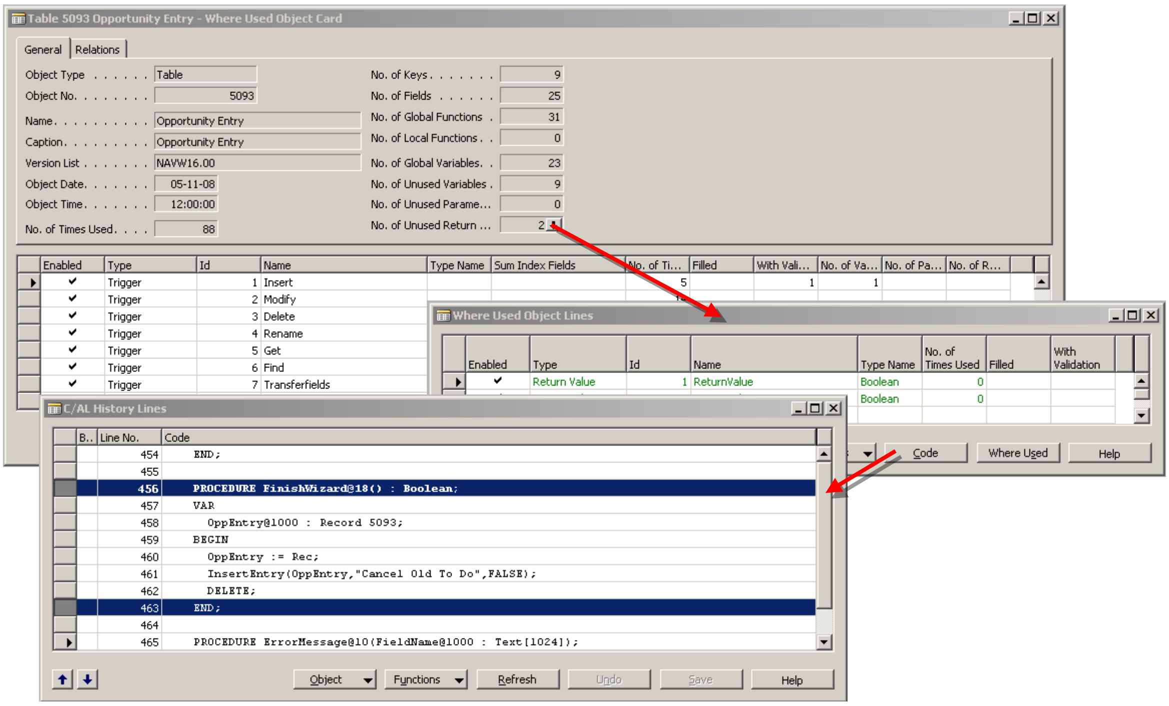
Object Type . . . . . . Codeunit
Object No. 22
Name. . . . . Item Jnl.-Post Line
Caption. . . . . Item Jnl.-Post Line
Version List NAVW16.00.01
Object Date. 14-08-09
Object Time. 12:00:00
No. of Times Used. . . . 28
No. of Keys. 0
No. of Fields 0
No. of Global Functions . 39
No. of Local Functions. . 74
No. of Global Variables. . 91
No. of Unused Variables . 13
No. of Unused Parame... 0
No. of Unused Return ... 3

Enabled	Type	Id	Name	Type Name	Sum Index Fields	No. of Ti...	Filled	With Vali...	No. of Va...	No. of Pa...	No. of R...
✓	Local Function	5819	RetrieveCostPerUnit			1					1
✓	Local Function	5820	UpdateAvgCostAdjmtEntryPoint			2			2	2	
✓	Local Function	5823	CalcPosShares			1			1	9	
✓	Local Function	5824	RetrieveCostPerUnitACY			1				1	1
✓	Local Function	5825	UpdateOutboundItemLedgEntry			2			1	1	
✓	Local Function	5826	CalcPurchCorrShares			1			4	4	
✓	Local Function	5850	AutoTrack								
✓	Local Function	7300	GetLocation								

Enabled	Type	Id	Name	Type Name	No. of Times Used	Filled	With Validation
✓	Return Value	1	ReturnValue	Decimal	2		

```
LOCAL PROCEDURE RetrieveCostPerUnitACY@5824(CostPerUnit@1000 : Decimal) : Decimal;
BEGIN
  WITH ItemJnlLine DO BEGIN
    IF Adjustment OR ("Source Currency Code" = GLSetup."Additional Reporting Currency...
       ((Item."Costing Method" <> Item."Costing Method"::Standard) OR
        (("Discount Amount" = 0) AND ("Indirect Cost %" = 0) AND ("Overhead Rate" = 0)))
    THEN
      EXIT("Unit Cost (ACY)")
    ELSE
      EXIT(ROUND(CurrExchRate.ExchangeAmtLCYToFCY(
            "Posting Date",GLSetup."Additional Reporting Currency",
            CostPerUnit,CurrExchRate.ExchangeRate(
              "Posting Date",GLSetup."Additional Reporting Currency")),
          Currency."Unit-Amount Rounding Precision"));
  END;
END;
```

Or see what return values are not being used.



Object Type . . . . . . Table
Object No. 5093
Name. . . . . Opportunity Entry
Caption. . . . . Opportunity Entry
Version List NAVW16.00
Object Date. 05-11-08
Object Time. 12:00:00
No. of Times Used. . . . 88
No. of Keys. 9
No. of Fields 25
No. of Global Functions . 31
No. of Local Functions. . 0
No. of Global Variables. . 23
No. of Unused Variables . 9
No. of Unused Parame... 0
No. of Unused Return ... 2

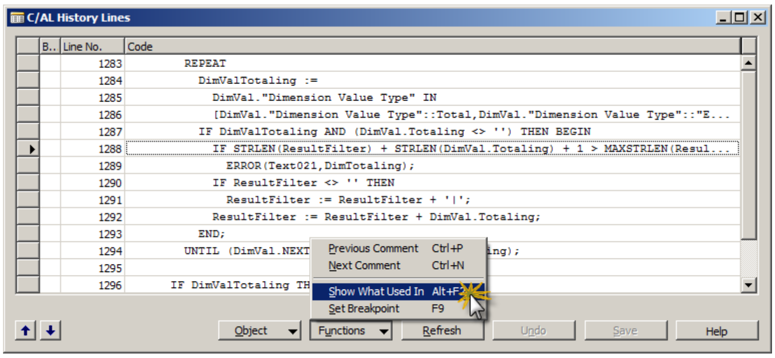
Enabled	Type	Id	Name	Type Name	Sum Index Fields	No. of Ti...	Filled	With Vali...	No. of Va...	No. of Pa...	No. of R...
✓	Trigger	1	Insert			5		1	1		
✓	Trigger	2	Modify								
✓	Trigger	3	Delete								
✓	Trigger	4	Rename								
✓	Trigger	5	Get								
✓	Trigger	6	Find								
✓	Trigger	7	Transferfields								

Enabled	Type	Id	Name	Type Name	No. of Times Used	Filled	With Validation
✓	Return Value	1	ReturnValue	Boolean	0		
				Boolean	0		

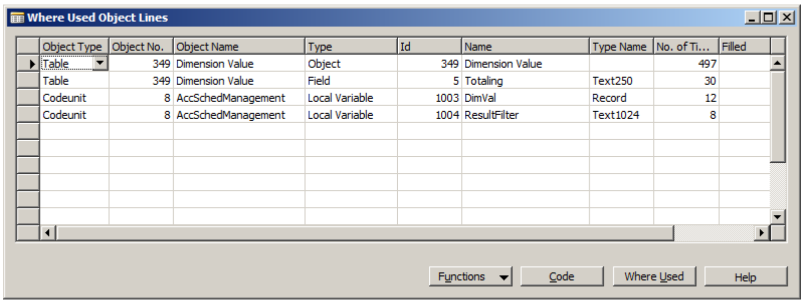
```
END;

PROCEDURE FinishWizard@18() : Boolean;
VAR
  OppEntry@1000 : Record 5093;
BEGIN
  OppEntry := Rec;
  InsertEntry(OppEntry,"Cancel Old To Do",FALSE);
  DELETE;
END;

PROCEDURE ErrorMessage@10(FieldName@1000 : Text[1024]);
```

15.5. What Used In

If you want to see what is used in a line of code you can use the function “What Used In”.


C/AL History Lines

```
        REPEAT
          DimValTotaling :=
            DimVal."Dimension Value Type" IN
            [DimVal."Dimension Value Type"::Total,DimVal."Dimension Value Type"::"E...
          IF DimValTotaling AND (DimVal.Totaling <> '') THEN BEGIN
            IF STRLEN(ResultFilter) + STRLEN(DimVal.Totaling) + 1 > MAXSTRLEN(Resul...
              ERROR(Text021,DimTotaling);
            IF ResultFilter <> '' THEN
              ResultFilter := ResultFilter + '|';
            ResultFilter := ResultFilter + DimVal.Totaling;
          END;
        UNTIL (DimVal.NEXT                   ing);

      IF DimValTotaling TH
```


Where Used Object Lines

Object Type	Object No.	Object Name	Type	Id	Name	Type Name	No. of Ti...	Filled
Table	349	Dimension Value	Object	349	Dimension Value		497	
Table	349	Dimension Value	Field	5	Totaling	Text250	30	
Codeunit	8	AccSchedManagement	Local Variable	1003	DimVal	Record	12	
Codeunit	8	AccSchedManagement	Local Variable	1004	ResultFilter	Text1024	8	

15.6. Delete Where Used Object Lines

It is possible to delete unused “Where Used Object Lines”. It will also delete the entity from the object itself.

For example we have made a field 50.000 in the Customer table but is not used anywhere. We can delete that “Where Used Object Line”. The content of the field will be deleted and the field will be deleted from the table.

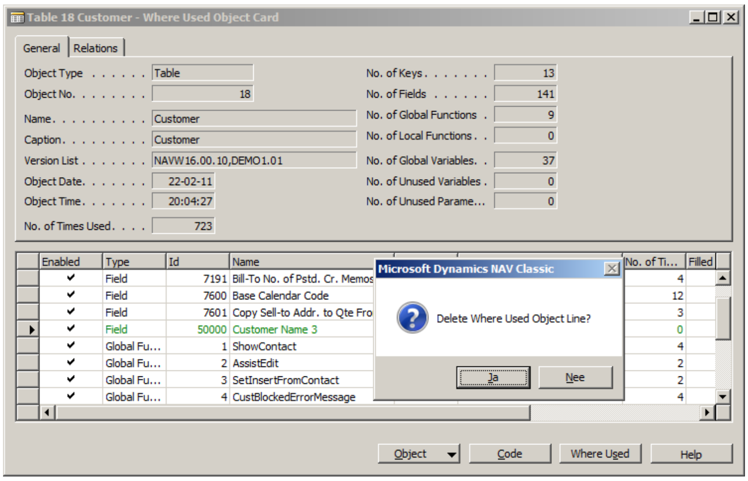


Table 18 Customer - Where Used Object Card

Object Type . . . . . . Table
Object No. 18
Name. . . . . Customer
Caption. . . . . Customer
Version List NAVW16.00.10,DEMO1.01
Object Date. 22-02-11
Object Time. 20:04:27
No. of Times Used. . . . 723
No. of Keys 13
No. of Fields 141
No. of Global Functions . 9
No. of Local Functions . . 0
No. of Global Variables. . 37
No. of Unused Variables . 0
No. of Unused Parame... 0

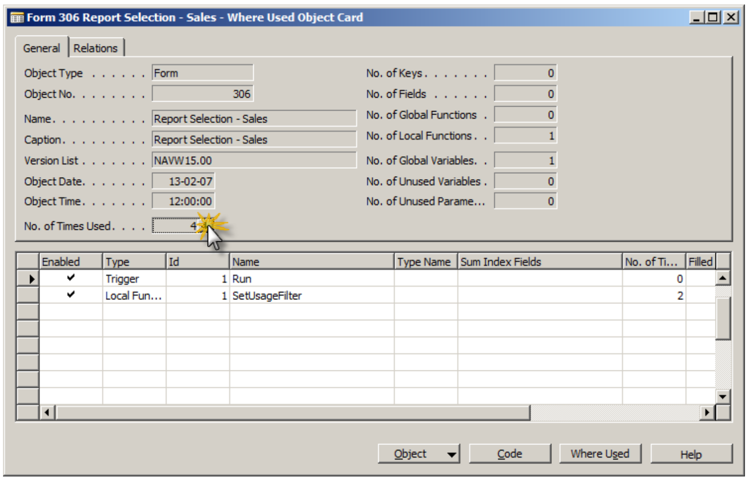
Enabled	Type	Id	Name	No. of Ti...	Filled
✓	Field	7191	Bill-To No. of Pstd. Cr. Memos	4	
✓	Field	7600	Base Calendar Code	12	
✓	Field	7601	Copy Sell-to Addr. to Qte Fro	3	
✓	Field	50000	Customer Name 3	0	
✓	Global Fu...	1	ShowContact	4	
✓	Global Fu...	2	AssistEdit	2	
✓	Global Fu...	3	SetInsertFromContact	2	
✓	Global Fu...	4	CustBlockedErrorMessage	4	

Microsoft Dynamics NAV Classic

Delete Where Used Object Line?

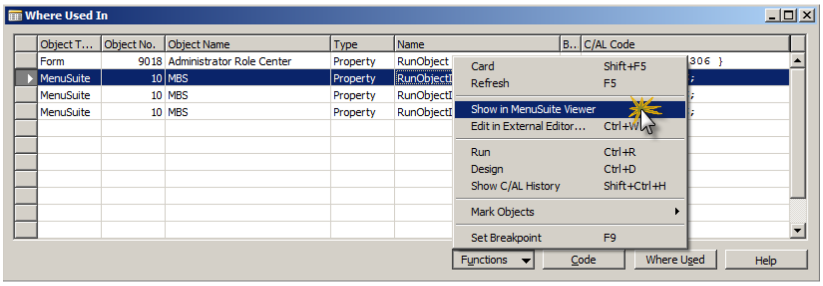
15.7. MenuSuite Viewer from Where Used

When objects are used in MenuSuites you can see with the function “Show in MenuSuite Viewer” where it is used.



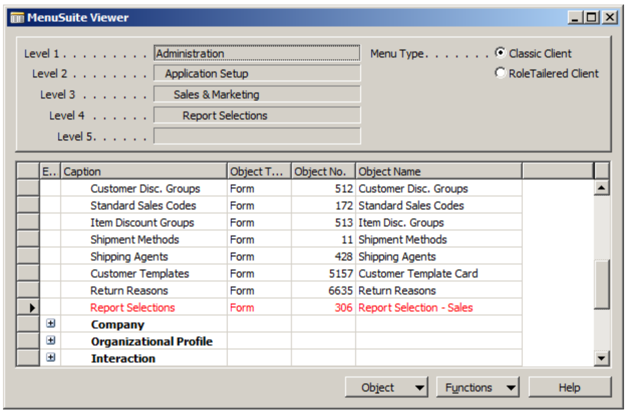
Object Type . . . . . . Form
Object No. 306
Name. . . . . Report Selection - Sales
Caption. . . . . Report Selection - Sales
Version List NAVW15.00
Object Date. 13-02-07
Object Time. 12:00:00
No. of Times Used. . . . 4
No. of Keys. 0
No. of Fields 0
No. of Global Functions . 0
No. of Local Functions . . 1
No. of Global Variables. . 1
No. of Unused Variables . 0
No. of Unused Parame... 0

Enabled	Type	Id	Name	Type Name	Sum Index Fields	No. of Ti...	Filled
✓	Trigger	1	Run			0	
✓	Local Fun...	1	SetUsageFilter			2	



Object T...	Object No.	Object Name	Type	Name	B..	C/AL Code
Form	9018	Administrator Role Center	Property	RunObject		306 }
MenuSuite	10	MBS	Property	RunObjectI		;
MenuSuite	10	MBS	Property	RunObjectI		;
MenuSuite	10	MBS	Property	RunObjectI		;

The Object will be shown as a red line.



Level 1 . . . . Administration
Level 2 . . . . Application Setup
Level 3 . . . . Sales & Marketing
Level 4 . . . . Report Selections
Level 5. . . . . . .
Menu Type. . . . . . . ● Classic Client ○ RoleTailered Client

E..	Caption	Object T...	Object No.	Object Name
	Customer Disc. Groups	Form	512	Customer Disc. Groups
	Standard Sales Codes	Form	172	Standard Sales Codes
	Item Discount Groups	Form	513	Item Disc. Groups
	Shipment Methods	Form	11	Shipment Methods
	Shipping Agents	Form	428	Shipping Agents
	Customer Templates	Form	5157	Customer Template Card
	Return Reasons	Form	6635	Return Reasons
	Report Selections	Form	306	Report Selection - Sales
⊞	**Company**			
⊞	**Organizational Profile**			
⊞	**Interaction**			

15.8. Enable or Disable Keys and Fields

Using the “Enabled” field on the lines of the “Where Used Object Card”, you can enable of disable keys and fields.

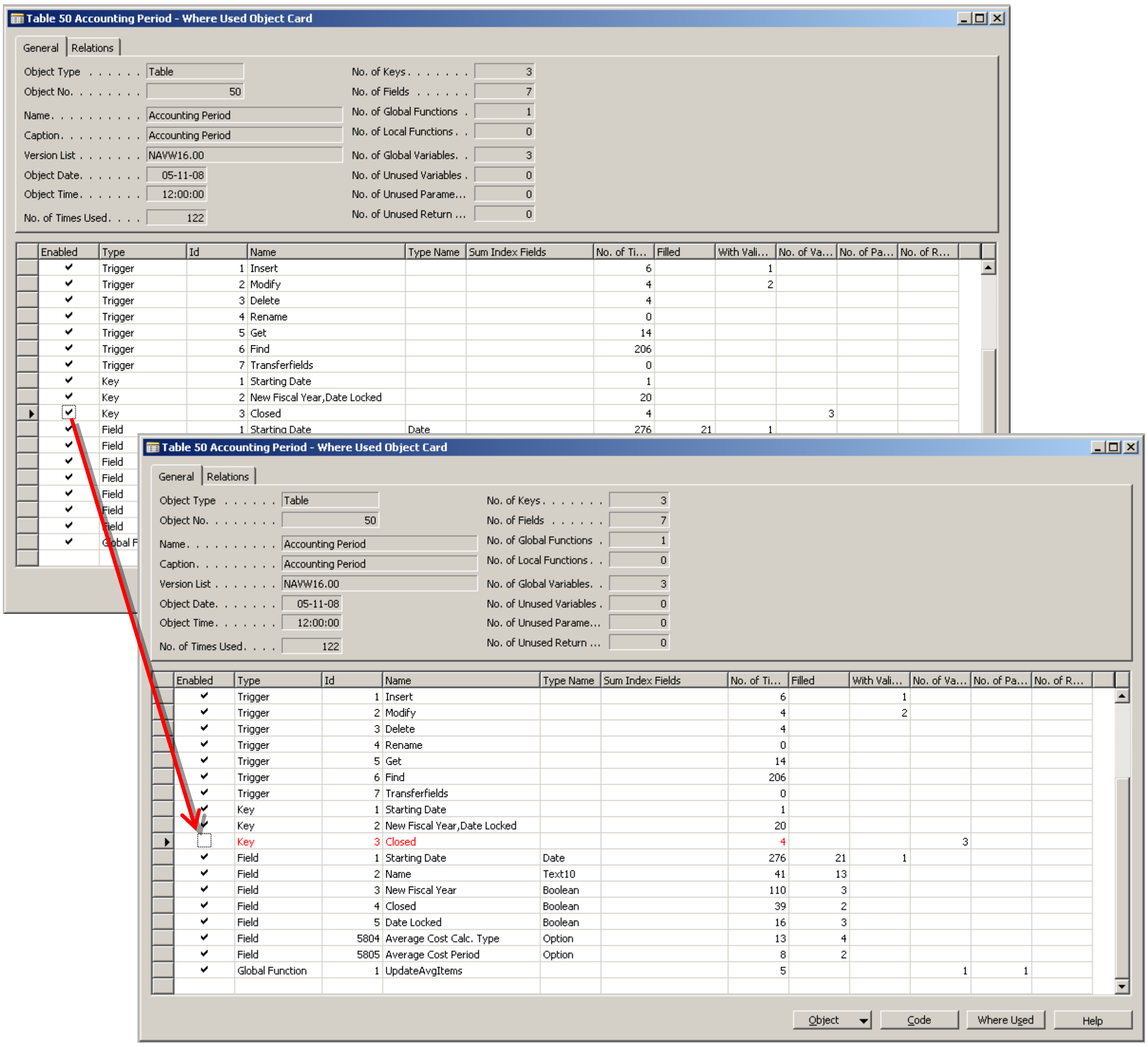


Table 50 Accounting Period - Where Used Object Card

Object Type	Table	No. of Keys	3
Object No.	50	No. of Fields	7
Name	Accounting Period	No. of Global Functions	1
Caption	Accounting Period	No. of Local Functions	0
Version List	NAVW16.00	No. of Global Variables	3
Object Date	05-11-08	No. of Unused Variables	0
Object Time	12:00:00	No. of Unused Parame…	0
No. of Times Used	122	No. of Unused Return …	0

Enabled	Type	Id	Name	Type Name	Sum Index Fields	No. of Ti…	Filled	With Vali…	No. of Va…	No. of Pa…	No. of R…
✓	Trigger	1	Insert			6		1			
✓	Trigger	2	Modify			4		2			
✓	Trigger	3	Delete			4					
✓	Trigger	4	Rename			0					
✓	Trigger	5	Get			14					
✓	Trigger	6	Find			206					
✓	Trigger	7	Transferfields			0					
✓	Key	1	Starting Date			1					
✓	Key	2	New Fiscal Year,Date Locked			20					
☑	Key	3	Closed			4			3		
✓	Field	1	Starting Date	Date		276	21	1			
✓	Field										
✓	Field										
✓	Field										
✓	Field										
✓	Field										
✓	Field										
✓	Global F										

Table 50 Accounting Period - Where Used Object Card

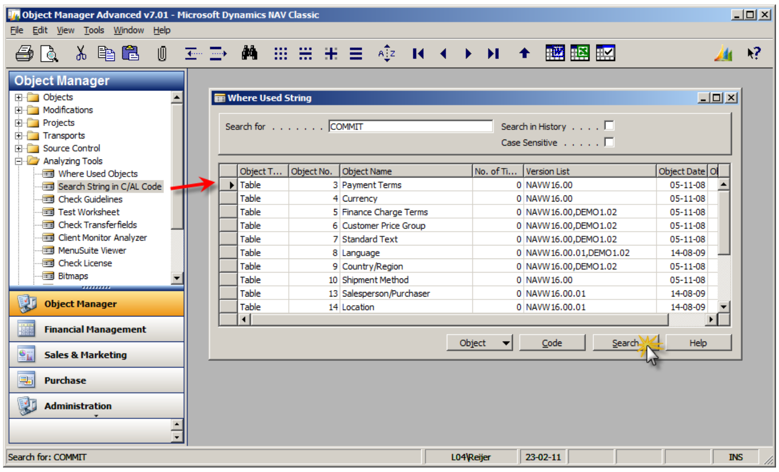
Object Type	Table	No. of Keys	3
Object No.	50	No. of Fields	7
Name	Accounting Period	No. of Global Functions	1
Caption	Accounting Period	No. of Local Functions	0
Version List	NAVW16.00	No. of Global Variables	3
Object Date	05-11-08	No. of Unused Variables	0
Object Time	12:00:00	No. of Unused Parame…	0
No. of Times Used	122	No. of Unused Return …	0

Enabled	Type	Id	Name	Type Name	Sum Index Fields	No. of Ti…	Filled	With Vali…	No. of Va…	No. of Pa…	No. of R…
✓	Trigger	1	Insert			6		1			
✓	Trigger	2	Modify			4		2			
✓	Trigger	3	Delete			4					
✓	Trigger	4	Rename			0					
✓	Trigger	5	Get			14					
✓	Trigger	6	Find			206					
✓	Trigger	7	Transferfields			0					
✓	Key	1	Starting Date			1					
✓	Key	2	New Fiscal Year,Date Locked			20					
☐	Key	3	Closed			4			3		
✓	Field	1	Starting Date	Date		276	21	1			
✓	Field	2	Name	Text10		41	13				
✓	Field	3	New Fiscal Year	Boolean		110	3				
✓	Field	4	Closed	Boolean		39	2				
✓	Field	5	Date Locked	Boolean		16	3				
✓	Field	5804	Average Cost Calc. Type	Option		13	4				
✓	Field	5805	Average Cost Period	Option		8	2				
✓	Global Function	1	UpdateAvgItems			5			1	1	

Note that this will not work for primary keys, fields containing data, fields being part of an active key, and any other type of lines, like Trigger, Global Function, etc.

16. Search String in C/AL Code

With this tool you can search for a string in objects.



Search for . . . . COMMIT

Search in History . . . . ☐

Case Sensitive . . . . . ☐

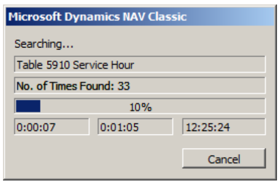
Object T...	Object No.	Object Name	No. of Ti...	Version List	Object Date
Table	3	Payment Terms	0	NAVW16.00	05-11-08
Table	4	Currency	0	NAVW16.00	05-11-08
Table	5	Finance Charge Terms	0	NAVW16.00,DEMO1.02	05-11-08
Table	6	Customer Price Group	0	NAVW16.00,DEMO1.02	05-11-08
Table	7	Standard Text	0	NAVW16.00,DEMO1.02	05-11-08
Table	8	Language	0	NAVW16.00.01,DEMO1.02	14-08-09
Table	9	Country/Region	0	NAVW16.00,DEMO1.02	05-11-08
Table	10	Shipment Method	0	NAVW16.00	05-11-08
Table	13	Salesperson/Purchaser	0	NAVW16.00.01	14-08-09
Table	14	Location	0	NAVW16.00.01	14-08-09

Search for: COMMIT | L04\Reijer | 23-02-11 | INS

In the “Search for” you can enter the statement you want to search for.

- **Search in History**
  The function will not only search for the string in the actual objects but also in the former versions.

Press Search.



Searching...

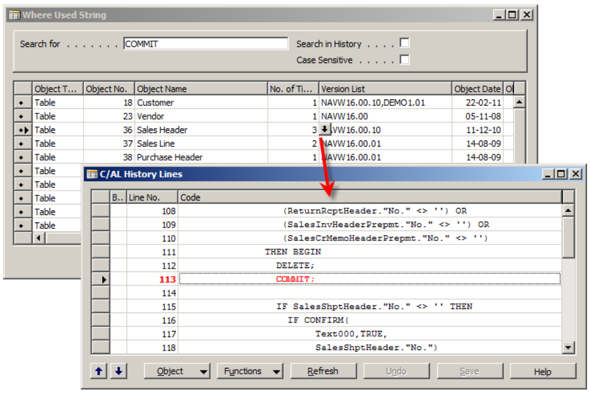
Table 5910 Service Hour

No. of Times Found: 33

10%

0:00:07 | 0:01:05 | 12:25:24

The “No. of Times” column is filled and it is possible to zoom in to the C/AL where the statement is used.



Search for . . . . COMMIT

Search in History . . . . ☐

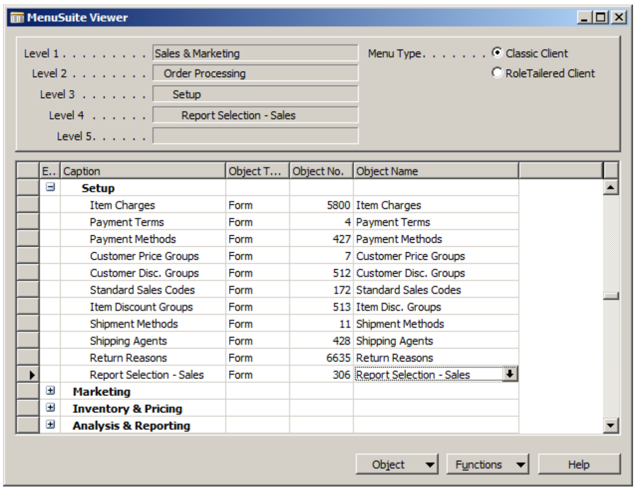
Case Sensitive . . . . . ☐

Object T...	Object No.	Object Name	No. of Ti...	Version List	Object Date
Table	18	Customer	1	NAVW16.00.10,DEMO1.01	22-02-11
Table	23	Vendor	1	NAVW16.00	05-11-08
Table	36	Sales Header	3	VW16.00.10	11-12-10
Table	37	Sales Line	2	NAVW16.00.01	14-08-09
Table	38	Purchase Header	1	NAVW16.00.01	14-08-09
Table					
Table					
Table					
Table					
Table					

```
108                (ReturnRcptHeader."No." <> '') OR
109                (SalesInvHeaderPrepmt."No." <> '') OR
110                (SalesCrMemoHeaderPrepmt."No." <> '')
111            THEN BEGIN
112              DELETE;
113              COMMIT;
114
115              IF SalesShptHeader."No." <> '' THEN
116                IF CONFIRM(
117                     Text000,TRUE,
118                     SalesShptHeader."No.")
```

17. MenuSuite Viewer

With this tool you can check where an object is situated in a MenuSuite of the Classic Client and the Role Tailored Client.



Level 1 . . . . Sales & Marketing
Level 2 . . . . Order Processing
Level 3 . . . . Setup
Level 4 . . . . Report Selection - Sales
Level 5. . . . . .

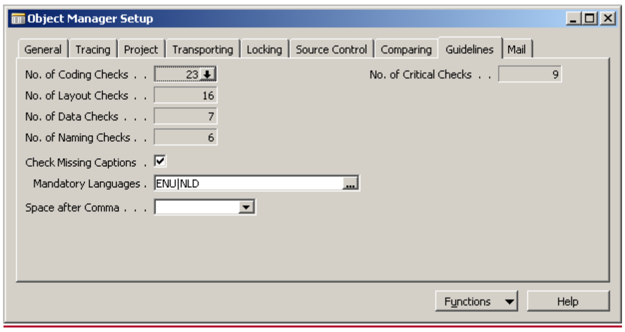
Menu Type. . . . . . . ● Classic Client ○ RoleTailered Client

E..	Caption	Object T...	Object No.	Object Name
⊟	**Setup**			
	Item Charges	Form	5800	Item Charges
	Payment Terms	Form	4	Payment Terms
	Payment Methods	Form	427	Payment Methods
	Customer Price Groups	Form	7	Customer Price Groups
	Customer Disc. Groups	Form	512	Customer Disc. Groups
	Standard Sales Codes	Form	172	Standard Sales Codes
	Item Discount Groups	Form	513	Item Disc. Groups
	Shipment Methods	Form	11	Shipment Methods
	Shipping Agents	Form	428	Shipping Agents
	Return Reasons	Form	6635	Return Reasons
	Report Selection - Sales	Form	306	Report Selection - Sales
⊞	**Marketing**			
⊞	**Inventory & Pricing**			
⊞	**Analysis & Reporting**			

18. Check Guidelines

This feature will check and/or correct C/AL code of selected objects in NAV. The C/AL code and layout of forms is checked if they meet the Microsoft Guidelines requirements.

18.1. Setup

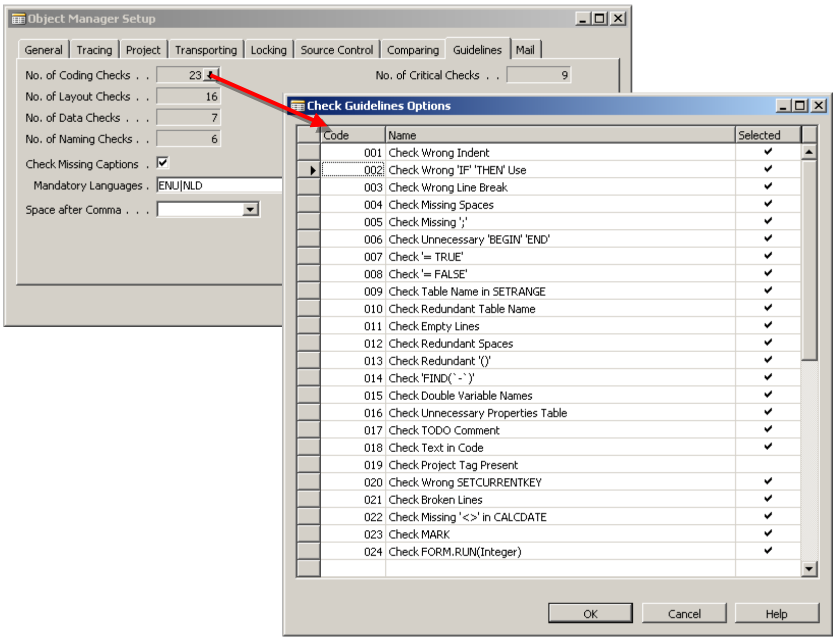


No. of Coding Checks . . 23
No. of Layout Checks . . 16
No. of Data Checks . . . 7
No. of Naming Checks . . 6
Check Missing Captions . ☑
Mandatory Languages . ENU|NLD
Space after Comma . . .
No. of Critical Checks . . 9

There are 6 types of checks:

- Coding Checks
- Layout Checks
- Data Checks
- Naming Checks
- Missing Captions
- Space after Comma

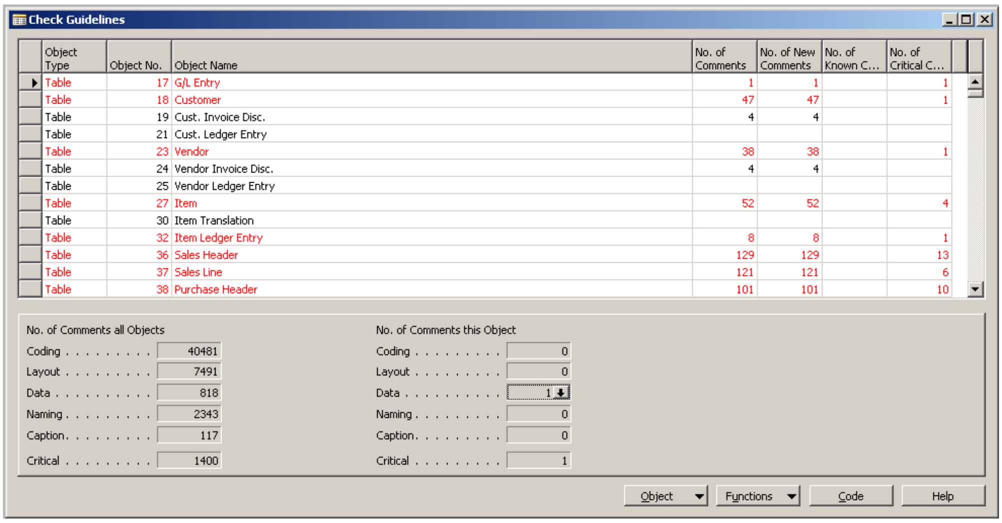
Zooming in to the Check Guidelines Options window for each check, you will be able to select or deselect any of the guidelines checks.



No. of Coding Checks . . 23
No. of Layout Checks . . 16
No. of Data Checks . . . 7
No. of Naming Checks . . 6
Check Missing Captions . ☑
Mandatory Languages . ENU|NLD
Space after Comma . . .
No. of Critical Checks . . 9

Code	Name	Selected
001	Check Wrong Indent	✓
002	Check Wrong 'IF' 'THEN' Use	✓
003	Check Wrong Line Break	✓
004	Check Missing Spaces	✓
005	Check Missing ';'	✓
006	Check Unnecessary 'BEGIN' 'END'	✓
007	Check '= TRUE'	✓
008	Check '= FALSE'	✓
009	Check Table Name in SETRANGE	✓
010	Check Redundant Table Name	✓
011	Check Empty Lines	✓
012	Check Redundant Spaces	✓
013	Check Redundant '()'	✓
014	Check 'FIND(`-`)'	✓
015	Check Double Variable Names	✓
016	Check Unnecessary Properties Table	✓
017	Check TODO Comment	✓
018	Check Text in Code	✓
019	Check Project Tag Present	
020	Check Wrong SETCURRENTKEY	✓
021	Check Broken Lines	✓
022	Check Missing '<>' in CALCDATE	✓
023	Check MARK	✓
024	Check FORM.RUN(Integer)	✓

If you would like to define a check as critical zoom into the “No. of Critical Check” field a select the guideline check. Any object non complying to one of the critical checks will be displayed in red in the “Check Guidelines” window.



Object Type	Object No.	Object Name	No. of Comments	No. of New Comments	No. of Known C...	No. of Critical C...
Table	17	G/L Entry	1	1		1
Table	18	Customer	47	47		1
Table	19	Cust. Invoice Disc.	4	4		
Table	21	Cust. Ledger Entry				
Table	23	Vendor	38	38		1
Table	24	Vendor Invoice Disc.	4	4		
Table	25	Vendor Ledger Entry				
Table	27	Item	52	52		4
Table	30	Item Translation				
Table	32	Item Ledger Entry	8	8		1
Table	36	Sales Header	129	129		13
Table	37	Sales Line	121	121		6
Table	38	Purchase Header	101	101		10

No. of Comments all Objects

Coding 40481

Layout 7491

Data 818

Naming 2343

Caption. 117

Critical 1400

No. of Comments this Object

Coding 0

Layout 0

Data 1

Naming 0

Caption. 0

Critical 1

18.2. Coding Checks

This consists of requirements the coding has to meet. Most of these requirements can be autocorrected. See section 18.10 - Auto Apply Guidelines for more information .

Code	Name	Autocorrect
001	Check Wrong Indent	X
002	Check Wrong 'IF' 'THEN' Use	X
003	Check Wrong Line Break	X
004	Check Missing Spaces	X
005	Check Missing ';'	X
006	Check Unnecessary 'BEGIN' 'END'	X
007	Check '= TRUE'	X
008	Check '= FALSE'	
009	Check Table Name in SETRANGE	X
010	Check Redundant Table Name	X
011	Check Empty Lines	X
012	Check Redundant Spaces	X
013	Check Redundant '()'	X
014	Check 'FIND(`-`)'	
015	Check Double Variable Names	
016	Check Unnecessary Properties Table	
017	Check TODO Comment	
018	Check Text in Code	

019	Check Project Tag Present	
020	Check Wrong SETCURRENTKEY	
021	Check Broken Lines	
022	Check Missing '<>' in CALCDATE	
023	Check MARK	
024	Check FORM.RUN(Integer)	

18.2.1. Check Wrong Indent

Checks if indentation is correct in code.

Comment: Wrong indent: Remove 1 space(s)

```
IF SalesLine."Document Type" = SalesLine."Document Type"::Order THEN BEGIN
   SalesLine.VALIDATE(Type, SalesLine.Type::Item);
   SalesLine.VALIDATE("No.", Item."No.");
END;
```

Should be:

```
IF SalesLine."Document Type" = SalesLine."Document Type"::Order THEN BEGIN
  SalesLine.VALIDATE(Type, SalesLine.Type::Item);
  SalesLine.VALIDATE("No.", Item."No.");
END;
```

18.2.2. Check Wrong 'IF' 'THEN' Use

Check on use of IF THEN (BEGIN).

Comment: 'IF' must start at new line

```
IF "Document Type" = "Document Type"::Order THEN
  SalesLine.SETFILTER("Quantity Shipped",'<>0')
ELSE IF "Document Type" = "Document Type"::Invoice THEN BEGIN
    SalesLine.SETRANGE("Sell-to Customer No.",xRec."Sell-to Customer No.");
    SalesLine.SETFILTER("Shipment No.",'<>%1','');
  END;
```

Should be:

```
IF "Document Type" = "Document Type"::Order THEN
  SalesLine.SETFILTER("Quantity Shipped",'<>0')
ELSE
  IF "Document Type" = "Document Type"::Invoice THEN BEGIN
    SalesLine.SETRANGE("Sell-to Customer No.",xRec."Sell-to Customer No.");
    SalesLine.SETFILTER("Shipment No.",'<>%1','');
  END;
```

Comment: Keep 'THEN' and 'BEGIN' together

```
IF xRec."Ship-to Code" <> '' THEN
  BEGIN
  GetCust("Sell-to Customer No.");
  "Tax Area Code" := Cust."Tax Area Code";
END;
```

Should be:

```
IF xRec."Ship-to Code" <> '' THEN BEGIN
  GetCust("Sell-to Customer No.");
  "Tax Area Code" := Cust."Tax Area Code";
END;
```

Comment: 'REPEAT' must start at new line

```
IF ServLedgerEntry.FINDSET THEN REPEAT
  DiscountAmount := DiscountAmount + -ServLedgerEntry."Discount Amount";
  ContractDiscAmount := ContractDiscAmount + -ServLedgerEntry."Contract Disc. Amount";
UNTIL ServLedgerEntry.NEXT = 0;
```

Should be:

```
IF ServLedgerEntry.FINDSET THEN
  REPEAT
    DiscountAmount := DiscountAmount + -ServLedgerEntry."Discount Amount";
    ContractDiscAmount := ContractDiscAmount + -ServLedgerEntry."Contract Disc. Amount";
  UNTIL ServLedgerEntry.NEXT = 0;
```

Comment: 'THEN' must start at new line

```
IF ("Account Type" = "Account Type"::"IC Partner") AND
  ("Bal. Account Type" = "Bal. Account Type"::"G/L Account") THEN BEGIN
  GLAcc.SETRANGE("No.","Bal. Account No.");
  IF GLAcc.FIND('-') THEN
    "IC Partner G/L Acc. No." := GLAcc."Default IC Partner G/L Acc. No";
END;
```

Should be:

```
IF ("Account Type" = "Account Type"::"IC Partner") AND
   ("Bal. Account Type" = "Bal. Account Type"::"G/L Account")
THEN BEGIN
  GLAcc.SETRANGE("No.","Bal. Account No.");
  IF GLAcc.FIND('-') THEN
    "IC Partner G/L Acc. No." := GLAcc."Default IC Partner G/L Acc. No";
END;
```

Comment: Start new line after ELSE

```
IF MapPoint.FIND('-') THEN
  MapMgt.MakeSelection(DATABASE::Contact,GETPOSITION)
ELSE MESSAGE(Text033);
```

Should be:

```
IF MapPoint.FIND('-') THEN
  MapMgt.MakeSelection(DATABASE::Contact,GETPOSITION)
ELSE
  MESSAGE(Text033);
```

18.2.3. Check Wrong Line Break

If a line is broken in two lines you should break it at the first possible break position.

Comment: Break the line at position 14

```
IF AskQuestion THEN BEGIN
  Question := STRSUBSTNO(
      Text031 +
      Text032,ChangedFieldName);
```

Should be:

```
IF AskQuestion THEN BEGIN
  Question :=
    STRSUBSTNO(
      Text031 +
      Text032,ChangedFieldName);
```

If you use a BEGIN in a CASE construction the BEGIN statement must always start at a new line.

Comment: 'BEGIN' must start at new line

```
CASE "Table ID" OF
  DATABASE::"G/L Account": BEGIN
    IF GLAcc.GET("No.") THEN BEGIN
      GLAcc."Global Dimension 1 Code" := NewDimValue;
      GLAcc.MODIFY(TRUE);
    END;
  END;
END;
```

Should be:

```
CASE "Table ID" OF
  DATABASE::"G/L Account":
    BEGIN
      IF GLAcc.GET("No.") THEN BEGIN
        GLAcc."Global Dimension 1 Code" := NewDimValue;
        GLAcc.MODIFY(TRUE);
      END;
    END;
END;
```

18.2.4. Check Missing Spaces

Checks if statements are glued together where this is not allowed.

Comment: Add space at position 27

```
IF "Applies-to Doc. No." <>'' THEN
  CustLedgEntry.SETRANGE("Document No.","Applies-to Doc. No.");
```

Should be:

```
IF "Applies-to Doc. No." <> '' THEN
  CustLedgEntry.SETRANGE("Document No.","Applies-to Doc. No.");
```

18.2.5. Check Missing ';'

Checks for open ends before an END;

Comment: Add a ';'

```
WITH PaymentTermsTranslation DO BEGIN
  SETRANGE("Payment Term",Code);
  DELETEALL
END;
```

Should be:

```
WITH PaymentTermsTranslation DO BEGIN
  SETRANGE("Payment Term",Code);
  DELETEALL;
END;
```

18.2.6. Check Unnecessary 'BEGIN' 'END'

Checks where a ‘BEGIN’ and ‘END’ is used where this is not necessary.

Comments: Remove 'BEGIN' and Remove ‘END'

```
IF "Price Includes VAT" THEN BEGIN
  IF NOT VATPostingSetup.GET("VAT Bus. Posting Gr. (Price)","VAT Prod. Posting Group") THEN
    FIELDERROR("VAT Bus. Posting Gr. (Price)");
END;
```

Should be:

```
IF "Price Includes VAT" THEN
  IF NOT VATPostingSetup.GET("VAT Bus. Posting Gr. (Price)","VAT Prod. Posting Group") THEN
    FIELDERROR("VAT Bus. Posting Gr. (Price)");
```

18.2.7. Check '= TRUE'

Checks where ‘= TRUE’ is added in an equation. This is not necessary and should be removed.

Comment: Remove '= TRUE'

```
IF Complete = TRUE THEN
  MESSAGE(Text003,ReportCaption1,ReportCaption2)
```

Should be:

```
IF Complete THEN
  MESSAGE(Text003,ReportCaption1,ReportCaption2)
```

18.2.8. Check '= FALSE'

Comment: Use NOT instead of '= FALSE'

```
IF "Related to Base Unit of Meas." = FALSE THEN
  "Qty. per Unit of Measure" := 1;
```

Should be:

```
IF NOT "Related to Base Unit of Meas." THEN
  "Qty. per Unit of Measure" := 1;
```

18.2.9. Check Table Name in SETRANGE

Checks if the variablename of the record is stated inside of a setrange function. All other statements (like SETFILTER, SETCURRENTKEY, etc.) are also checked.

Comment: Remove tablename

```
SalesPrice.SETRANGE("Sales Type",SalesPrice."Sales Type"::Campaign);
SalesPrice.SETRANGE(SalesPrice."Sales Code","No.");
SalesPrice.LOCKTABLE;
```

Should be:

```
SalesPrice.SETRANGE("Sales Type",SalesPrice."Sales Type"::Campaign);
SalesPrice.SETRANGE("Sales Code","No.");
SalesPrice.LOCKTABLE;
```

18.2.10. Check Redundant Table Name

Checks for unnecessary recordreferences (Tablenames). This happens the most in reports where the name of the DataItem is not necessary in the code.

Comment: Remove tablename

```
VAT Entry - OnPreDataItem()
"VAT Entry".COPYFILTERS(VATEntry);
```

Should be:

```
VAT Entry - OnPreDataItem()
COPYFILTERS(VATEntry);
```

If you use the WITH statement in your code then the tablename is not necessary in the following code:

Comment: Remove tablename

```
WITH Period DO BEGIN
  Period.SETRANGE("Period Type","Period Type"::Date);
  SETFILTER("Period Start",DateFilter);
  IF FIND('-') THEN
    EXIT("Period Start")
END;
```

Should be:

```
WITH Period DO BEGIN
```

```
  SETRANGE("Period Type","Period Type"::Date);
  SETFILTER("Period Start",DateFilter);
  IF FIND('-') THEN
    EXIT("Period Start")
END;
```

18.2.11. Check Empty Lines

Checks if there are two or more empty lines. And if a function starts with an empty line.

Comment: Remove empty line

```
IF Campaign.GET(GETFILTER("Campaign No.")) THEN
  "Campaign Description" := Campaign.Description;


IF SegHeader.GET(GETFILTER("Segment No.")) THEN
  "Segment Description" := SegHeader.Description;
```

Should be:

```
IF Campaign.GET(GETFILTER("Campaign No.")) THEN
  "Campaign Description" := Campaign.Description;

IF SegHeader.GET(GETFILTER("Segment No.")) THEN
  "Segment Description" := SegHeader.Description;
```

18.2.12. Check Redundant Spaces

Checks unnecessary spaces.

Comment: Remove space at position 44

```
IF CustLedgEntry."Amount to Apply" = 0 THEN  BEGIN
  CustLedgEntry.CALCFIELDS("Remaining Amount");
  CustLedgEntry."Amount to Apply" := CustLedgEntry."Remaining Amount";
END;
```

Should be:

```
IF CustLedgEntry."Amount to Apply" = 0 THEN BEGIN
  CustLedgEntry.CALCFIELDS("Remaining Amount");
  CustLedgEntry."Amount to Apply" := CustLedgEntry."Remaining Amount";
END;
```

18.2.13. Check Redundant '()'

Checks for unnecessary brackets.

Comment: Remove '()'

```
ReservEntry2 := ReservEntry;
ReservEntry2.ClearItemTrackingFields;
ReservEntry2.MODIFY();
```

Should be:

```
ReservEntry2 := ReservEntry;
ReservEntry2.ClearItemTrackingFields;
ReservEntry2.MODIFY;
```

Comment: Remove '(' at position 4, Remove ')' at position 26

```
IF (STRLEN(Parameter) = 2) THEN
  BizTalkNASStartup.RUN;
```

Should be:

```
IF STRLEN(Parameter) = 2 THEN
```

```
  BizTalkNASStartup.RUN;
```

18.2.14. Check 'FIND(`-`)'

Checks for FIND(‘-‘) instructions which should be replaced by FINDSET, FINDFIRST or ISEMPTY. Also spots FIND(‘+’) which should be changed to FINDLAST. The FIND(‘-‘) instructions may cause performance issues on SQL based Navision.

Comment: Replace FIND('-') with 'FINDFIRST', 'FINDSET' or 'ISEMPTY'

```
Job.SETRANGE("Bill-to Customer No.","No.");
IF Job.FIND('-') THEN
  ERROR(Text015,TABLECAPTION,"No.",Job.TABLECAPTION);
```

Should be:

```
Job.SETRANGE("Bill-to Customer No.","No.");
IF Job.FINDFIRST THEN
  ERROR(Text015,TABLECAPTION,"No.",Job.TABLECAPTION);
```

Comment: Replace FIND('+') with 'FINDLAST'

```
CustLedgEntry.SETRANGE(Open,TRUE);
IF CustLedgEntry.FIND('+') THEN
  ERROR(Text012,FIELDCAPTION("IC Partner Code"), CustLedgEntry."IC Partner Code");
```

Should be:

```
CustLedgEntry.SETRANGE(Open,TRUE);
IF CustLedgEntry.FINDLAST THEN
  ERROR(Text012,FIELDCAPTION("IC Partner Code"), CustLedgEntry."IC Partner Code");
```

18.2.15. Check Double Variable Names

Checks for variable names which are used local as well as global variable.

Comment: Variablename 'Bin' already in use as global

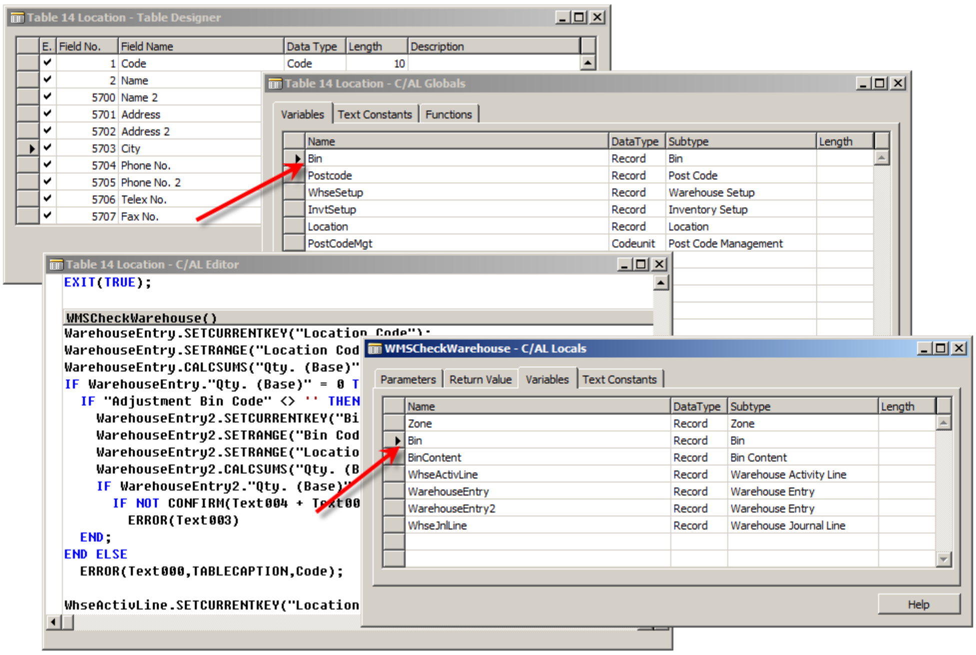


Table 14 Location - Table Designer
Table 14 Location - C/AL Globals
Variables
Text Constants
Functions
Table 14 Location - C/AL Editor
WMSCheckWarehouse - C/AL Locals
Parameters
Return Value
Variables
Text Constants
Help

E.	Field No.	Field Name	Data Type	Length	Description
✓	1	Code	Code	10	
✓	2	Name			
✓	5700	Name 2			
✓	5701	Address			
✓	5702	Address 2			
✓	5703	City			
✓	5704	Phone No.			
✓	5705	Phone No. 2			
✓	5706	Telex No.			
✓	5707	Fax No.			

Name	DataType	Subtype	Length
Bin	Record	Bin	
Postcode	Record	Post Code	
WhseSetup	Record	Warehouse Setup	
InvtSetup	Record	Inventory Setup	
Location	Record	Location	
PostCodeMgt	Codeunit	Post Code Management	

```
EXIT(TRUE);

WMSCheckWarehouse()
WarehouseEntry.SETCURRENTKEY("Location Code");
WarehouseEntry.SETRANGE("Location Cod
WarehouseEntry.CALCSUMS("Qty. (Base)"
IF WarehouseEntry."Qty. (Base)" = 0 T
  IF "Adjustment Bin Code" <> '' THEN
    WarehouseEntry2.SETCURRENTKEY("Bi
    WarehouseEntry2.SETRANGE("Bin Cod
    WarehouseEntry2.SETRANGE("Locatio
    WarehouseEntry2.CALCSUMS("Qty. (B
    IF WarehouseEntry2."Qty. (Base)"
      IF NOT CONFIRM(Text004 + Text00
        ERROR(Text003)
  END;
END ELSE
  ERROR(Text000,TABLECAPTION,Code);

WhseActivLine.SETCURRENTKEY("Location
```

Name	DataType	Subtype	Length
Zone	Record	Zone	
Bin	Record	Bin	
BinContent	Record	Bin Content	
WhseActivLine	Record	Warehouse Activity Line	
WarehouseEntry	Record	Warehouse Entry	
WarehouseEntry2	Record	Warehouse Entry	
WhseJnlLine	Record	Warehouse Journal Line	

18.2.16. Check Unnecessary Properties Table

Checks for properties on fields and tables which are not needed because they are the same as the default.

Comment: Remove property: 'Yes' is default for ‘TestTableRelation’

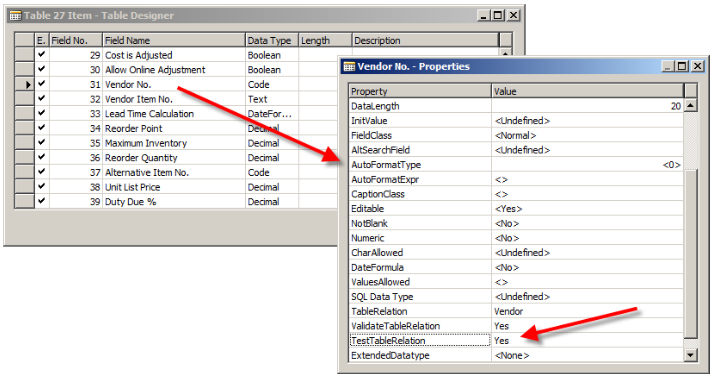


Table 27 Item - Table Designer

E.	Field No.	Field Name	Data Type	Length	Description
✓	29	Cost is Adjusted	Boolean		
✓	30	Allow Online Adjustment	Boolean		
✓	31	Vendor No.	Code		
✓	32	Vendor Item No.	Text		
✓	33	Lead Time Calculation	DateFor...		
✓	34	Reorder Point	Decimal		
✓	35	Maximum Inventory	Decimal		
✓	36	Reorder Quantity	Decimal		
✓	37	Alternative Item No.	Code		
✓	38	Unit List Price	Decimal		
✓	39	Duty Due %	Decimal		

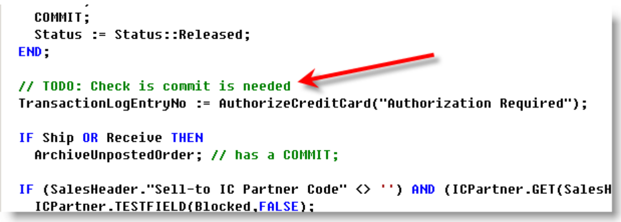
Vendor No. - Properties

Property	Value
DataLength	20
InitValue	<Undefined>
FieldClass	<Normal>
AltSearchField	<Undefined>
AutoFormatType	<0>
AutoFormatExpr	<>
CaptionClass	<>
Editable	<Yes>
NotBlank	<No>
Numeric	<No>
CharAllowed	<Undefined>
DateFormula	<No>
ValuesAllowed	<>
SQL Data Type	<Undefined>
TableRelation	Vendor
ValidateTableRelation	Yes
TestTableRelation	Yes
ExtendedDatatype	<None>

18.2.17. Check TODO Comment

Checks for “// TODO:” in C/AL code. You can use this as a reminder.

Comment: // TODO found in code



```
  COMMIT;
  Status := Status::Released;
END;

// TODO: Check is commit is needed
TransactionLogEntryNo := AuthorizeCreditCard("Authorization Required");

IF Ship OR Receive THEN
  ArchiveUnpostedOrder; // has a COMMIT;

IF (SalesHeader."Sell-to IC Partner Code" <> '') AND (ICPartner.GET(SalesH
  ICPartner.TESTFIELD(Blocked,FALSE);
```

18.2.18. Check Text in Code

Checks if there are hardcoded text messages instead of using textconstants.

Comment: Replace text by text constant

```
CASE globalVariable OF
  'CurrentRow': value := CurrentRow;
  'CurrentCol': value := CurrentCol;
  'RangeStartXlRow': value := RangeStartXlRow;
  'RangeStartXlCol': value := RangeStartXlCol;
  'RangeEndXlRow': value := RangeEndXlRow;
  'RangeEndXlCol': value := RangeEndXlCol;
  'XlWrkSht': value := XlWrkSht;
  ELSE
    ERROR('Global variable %1 is not included for test.',globalVariable);
END;
```

Should be:

```
CASE globalVariable OF
  'CurrentRow': value := CurrentRow;
  'CurrentCol': value := CurrentCol;
  'RangeStartXlRow': value := RangeStartXlRow;
  'RangeStartXlCol': value := RangeStartXlCol;
  'RangeEndXlRow': value := RangeEndXlRow;
  'RangeEndXlCol': value := RangeEndXlCol;
  'XlWrkSht': value := XlWrkSht;
  ELSE
```

```
    ERROR(Text001,globalVariable);
```

18.2.19. Check Project Tag Present

Checks if the no. of the project is found in the documentation trigger. This check will only be done if the check is done in a project or a transport.

Comment: Project tag 'P0001' not present

18.2.20. Check Wrong SETCURRENTKEY

Checks if a not existing key is used in the code.

Comment: Not Existing Key

```
WITH ProdOrderLine do begin
  RESET;
  SETCURRENTKEY(Status,"Completely Invoiced");
  SETRANGE(Status,Status::Finished);
  SETRANGE("Completely Invoiced",FALSE);
  EXIT(FIND('-'));
END;
```

18.2.21. Check Broken Lines

Checks if statement is split over multiple lines where this is not necessary.

Comment: Line does not have to be broken

```
IF WhseJnlLine.FINDFIRST THEN
  ERROR(
    Text007,
    FIELDCAPTION("Adjustment Bin Code"));
```

Should be:

```
IF WhseJnlLine.FINDFIRST THEN
  ERROR(Text007,FIELDCAPTION("Adjustment Bin Code"));
```

18.2.22. Check Missing '<>' in CALCDATE

Checks if the DateExpression in a CALCDATE call is entered with < > delimiters surrounding it.

Comment: Add '<>' to CALCDATE

```
AllowedPostingDate := CALCDATE('+1D',AllowedPostingDate);
```

Should be:

```
AllowedPostingDate := CALCDATE('<+1D>',AllowedPostingDate);
```

18.2.23. Check MARK

Identifies any call to MARK and suggests to use a temporary record instead.

Comment: Use temporary record instead of marking

18.2.24. Check FORM.RUN(Integer)

Checks if an integer is used in commands like FORM.RUN(…), REPORT.RUN(...), etc.

Comment: Replace integer with OBJECT::<object name>"

```
FORM.RUN(1);
```

Should be:

```
FORM.RUN(FORM::"Company Information");
```

18.3. Layout checks

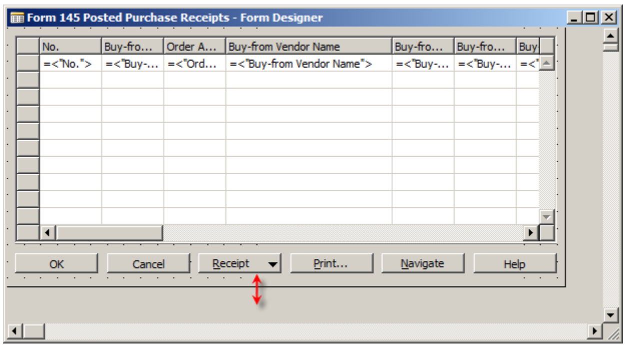
Code	Name
001	Check Form Margins
002	Check Margins Between Controls
003	Check Maximum Width of TableBox
004	Check Minimum Width of TableBox
005	Check Maximum Height of TableBox
006	Check Width of Controls in TableBox
007	Check Width of Controls in TabControl
008	Check Label Height
009	Check Position of Controls on TabControl
010	Check Overlapping Controls
011	Check Missing Glue Properties
012	Check Access Key on Buttons
013	Check Double Access Keys
014	Check Double Shortcut Keys
015	Check Lowercase Shortcut Keys
016	Check Label with Caption
017	Check No Border on SubForm
018	Check No. of Tabs in TabControl
019	Check Missing Help Button
020	Check Unnecessary Properties
021	Check Usage Wrong Keys
022	Check Unused Control Names

18.3.1. Check Form Margins

Checks the margins that the controls have with the parent form.

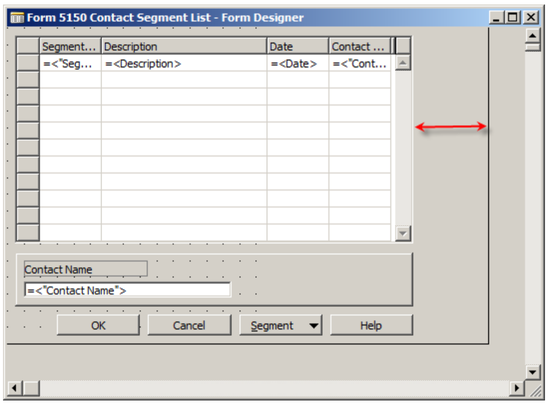
Comment: Form to height

The margin between the buttons and the lower edge of the form should be 220.



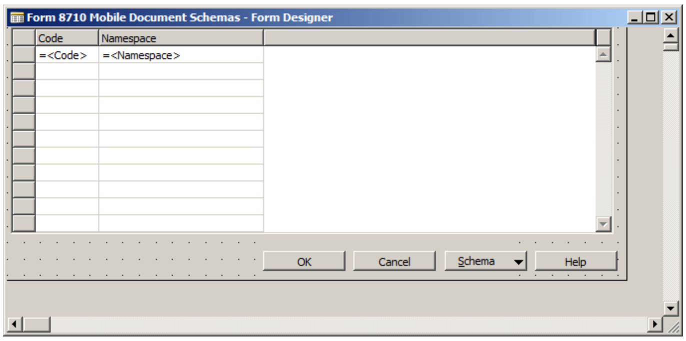
Comment: Form to wide

The margin between the controls and the right edge of the form should be 220.



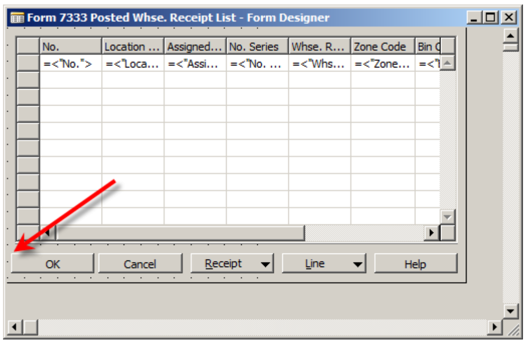
Comment: Keep topmargin of 220

The topmargin should be 220.



Comment: Keep leftmargin of 220

The OK button should have a margin of 220 with the left of the form.

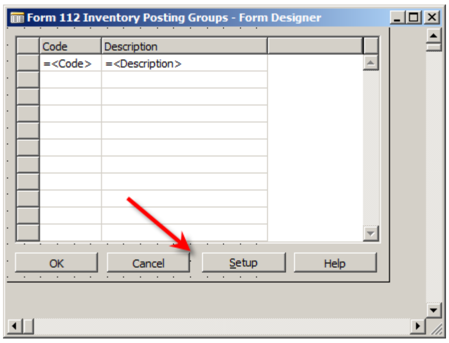


18.3.2. Check Margins Between Controls

Checks the margins between controls.

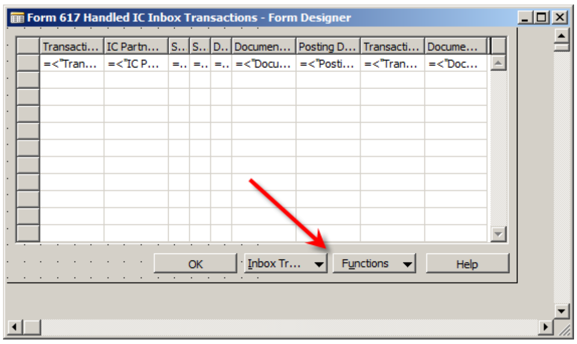
Comment: Rightmargin with CommandButton Setup to big

The margin between this controls should be 220.



Comment: Keep 220 margin with CommandButton 20

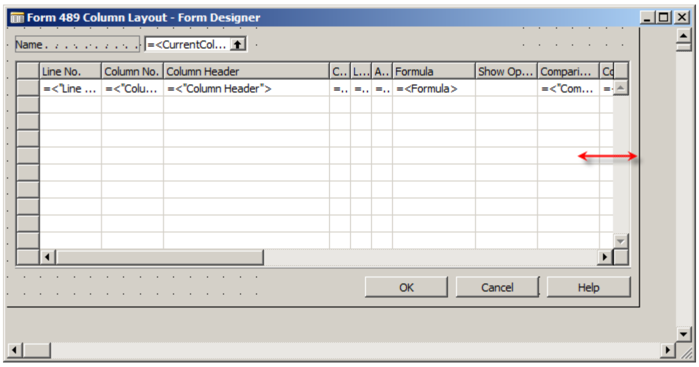
The margin between this controls should be 220.



18.3.3. Check Maximum Width of TableBox

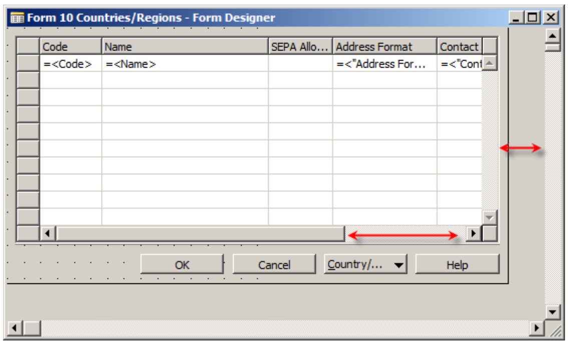
Check the width of tableboxes. The maximum width of a tablebox is 16060 in a main form and 15730 in a subform.

Comment: The maximum width of a TableBox is 16060



18.3.4. Check Minimum Width of TableBox

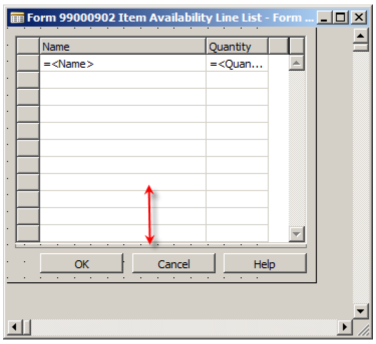
If a tablebox has a lot of visible textboxes and the width of the tablebox is less than 16060 you should extend this width to 16060.



18.3.5. Check Maximum Height of TableBox

Checks the height of tableboxes. The maximum height of a tablebox is 5500 in a main form and 3300 in a subform.

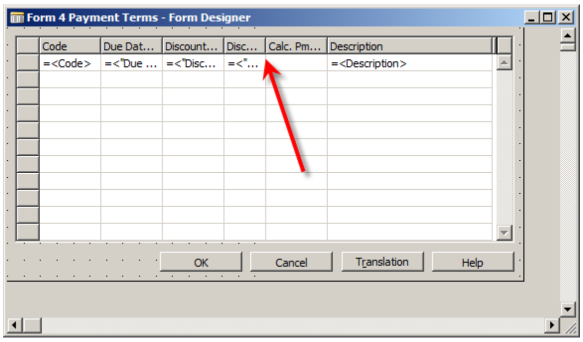
Comment: The maximum height of a TableBox is 5500



18.3.6. Check Width of Controls in TableBox

Checks Control widths in tablebox. Textboxes should have a width of 550, 1700, 2200, 2750 or 4400 in a tablebox or matrixbox.

Comment: A Textbox on a TableBox must have a width of 550, 1700, 2200, 2750 or 4400



Should be: This textbox should have a width of 2200.

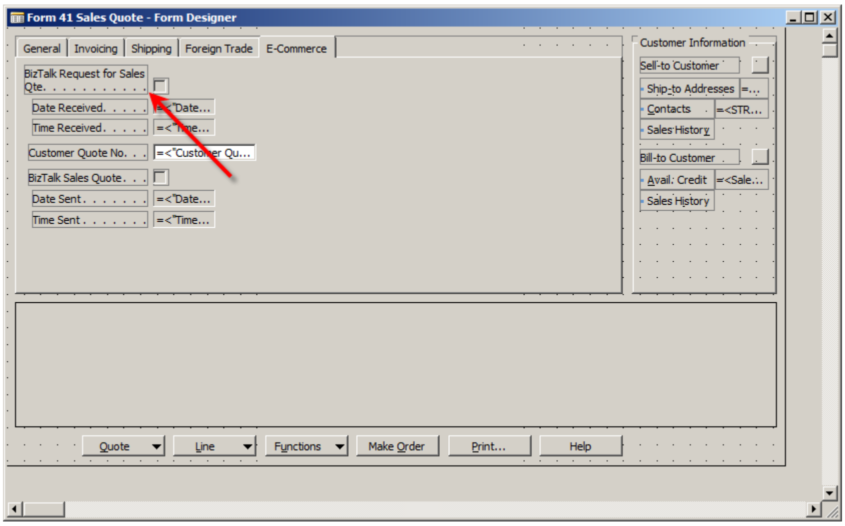
18.3.7. Check Width of Controls in TabControl

Checks the width of controls in Tabcontrols.
A Textbox on a TabControl must have a width of 1700, 2090, 2200, 2640, 2750, 4950, 5500

18.3.8. Check Label Height

Checks height label with corresponding textbox and vice versa.

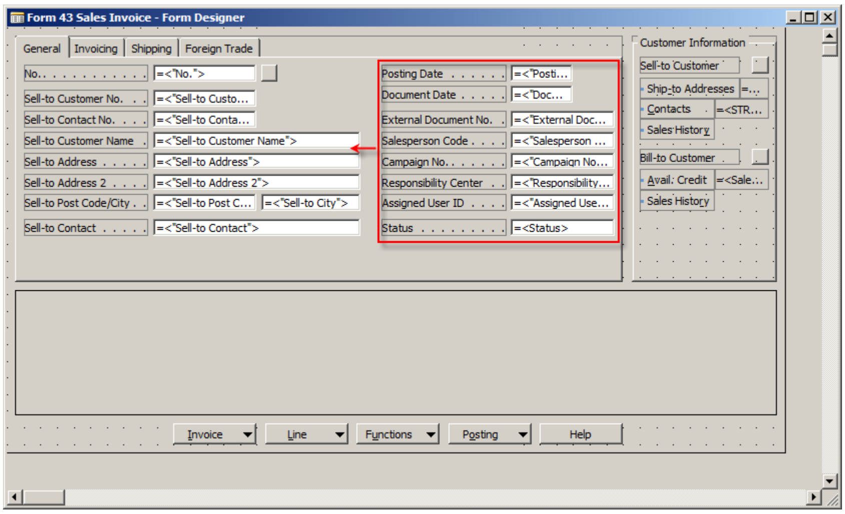
Comment: Height must be the same as CheckBox "BizTalk Request for Sales Qte."



18.3.9. Check Position of Controls on TabControl

Checks standard positions of (Card) form controls and if labels are aligned with their corresponding textboxes.

Comment: A Textbox on a tabcontrol must have an x pos of 3630 or 12760

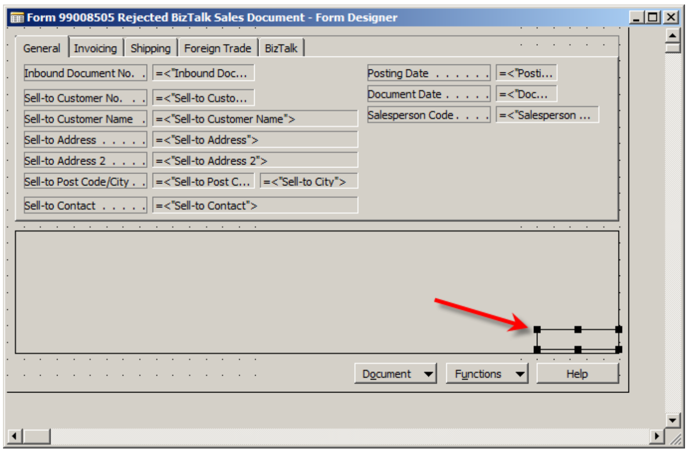


Should be: Right columns should be moved to the left to standard position

18.3.10. Check Overlapping Controls

Checks overlapping controls.

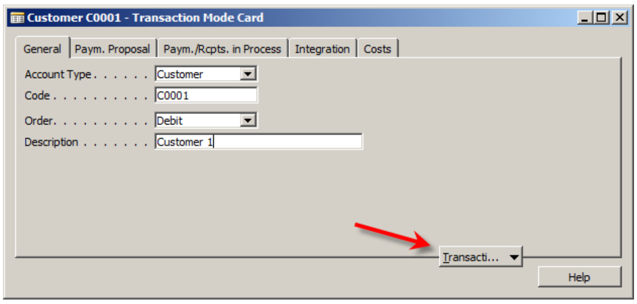
Comment: Overlapping with Subform 92



18.3.11. Check Missing Glue Properties

Checks for missing glue properties.
For example a new button on the form disappears because glue is not set when user expands his screen.

Comment: Add glue property. Control TabControl 1 is overlapping

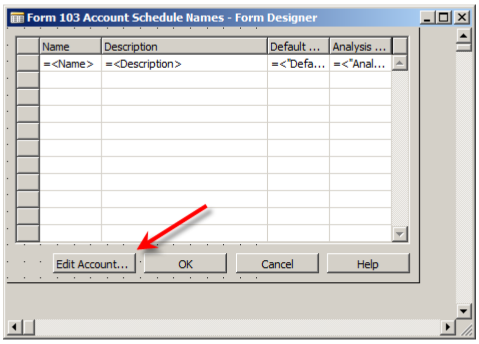


Account Type . . . . . . Customer
Code C0001
Order. . . . . Debit
Description . . . . Customer 1

18.3.12. Check Access Key on Buttons

Checks if all buttons have an acceskey to activate them. An Acceskey is ALT + underlined letter in a menu or a button

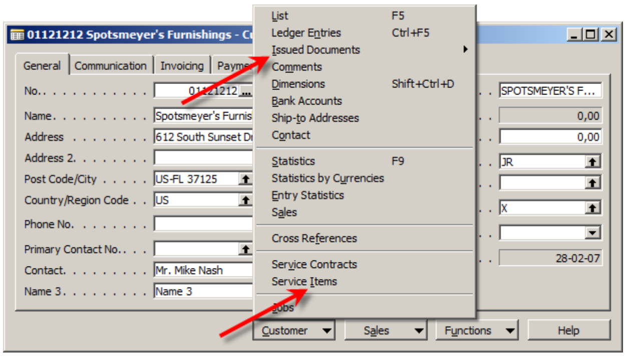
Comment: Add accesskey



18.3.13. Check Double Access Keys

Checks if Acceskeys are already used in the same object or menu.

Comment: Accesskey ALT+I (ENU) already in use, C available



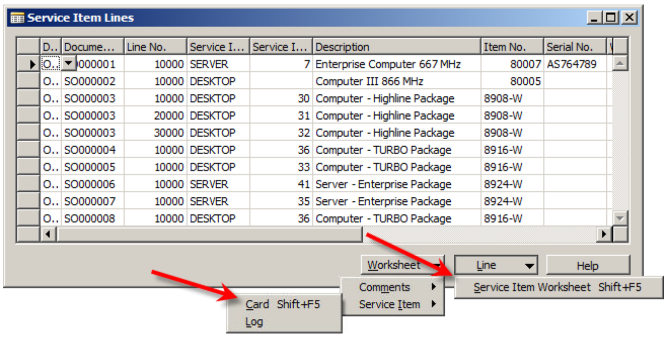
No.. 01121212
Name. . . . . Spotsmeyer's Furnis
Address . . . . 612 South Sunset Dr
Address 2. . . . .
Post Code/City . . . . . US-FL 37125
Country/Region Code . . US
Phone No. . . . .
Primary Contact No.. . .
Contact. . . . . Mr. Mike Nash
Name 3. . . . . Name 3

SPOTSMEYER'S F...
0,00
0,00
JR
X
28-02-07

18.3.14. Check Double Shortcut Keys

Checks if shortcut keys are used twice in the same object or a shortcut is used that is also used by NAV.
Shortcut keys are defined as CTRL or CTRL+SHIFT etc.

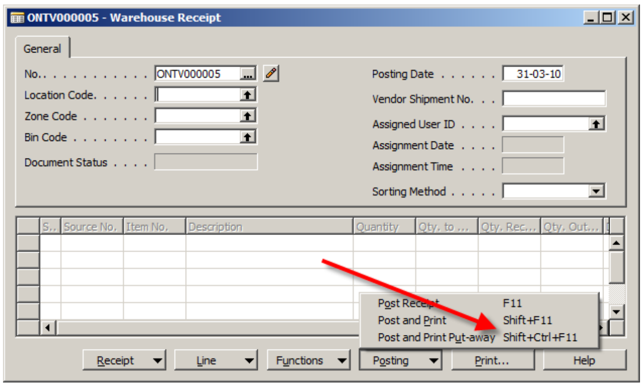
Comment: Shortcut key Shift+F5 already in use



D..	Docume...	Line No.	Service I...	Service I...	Description	Item No.	Serial No.
O..	000001	10000	SERVER	7	Enterprise Computer 667 MHz	80007	AS764789
O..	SO000002	10000	DESKTOP		Computer III 866 MHz	80005	
O..	SO000003	10000	DESKTOP	30	Computer - Highline Package	8908-W	
O..	SO000003	20000	DESKTOP	31	Computer - Highline Package	8908-W	
O..	SO000003	30000	DESKTOP	32	Computer - Highline Package	8908-W	
O..	SO000004	10000	DESKTOP	36	Computer - TURBO Package	8916-W	
O..	SO000005	10000	DESKTOP	33	Computer - TURBO Package	8916-W	
O..	SO000006	10000	SERVER	41	Server - Enterprise Package	8924-W	
O..	SO000007	10000	SERVER	35	Server - Enterprise Package	8924-W	
O..	SO000008	10000	DESKTOP	36	Computer - TURBO Package	8916-W	

(Example: 2 times SHIFT+F5 is used)

Comment: Shortcut key Shift+Ctrl+F11 already in use by NAV



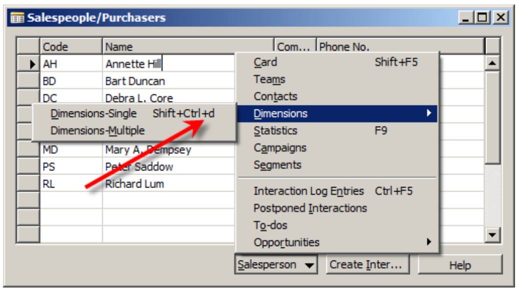
No. ONTV000005

Posting Date 31-03-10

18.3.15. Check Lowercase Shortcut Keys

Checks if lowercase shortcut keys are used.
It should always be uppercase.

Comment: Change Shortcut key to uppercase

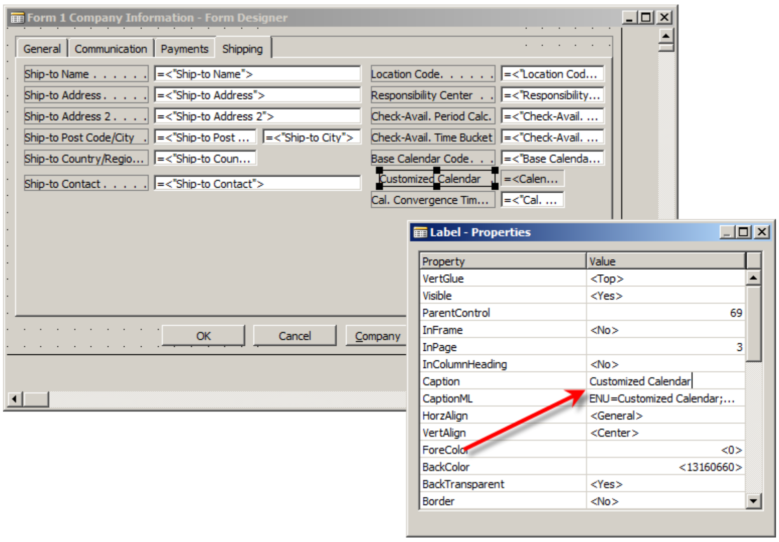


Code	Name
AH	Annette Hill
BD	Bart Duncan
DC	Debra L. Core
MD	Mary A. Dempsey
PS	Peter Saddow
RL	Richard Lum

18.3.16. Check Label with Caption

Checks if there are labels that have a caption that can be moved to the corresponding textbox.

Comment: Move caption to parent control



Ship-to Name . . . . . . =<"Ship-to Name">
Ship-to Address . . . . . =<"Ship-to Address">
Ship-to Address 2 . . . . =<"Ship-to Address 2">
Ship-to Post Code/City . =<"Ship-to Post ... =<"Ship-to City">
Ship-to Country/Regio... =<"Ship-to Coun...
Ship-to Contact . . . . . =<"Ship-to Contact">
Location Code. . . . . . =<"Location Cod...
Responsibility Center . . =<"Responsibility...
Check-Avail. Period Calc. =<"Check-Avail. ...
Check-Avail. Time Bucket =<"Check-Avail. ...
Base Calendar Code. . . =<"Base Calenda...
Customized Calendar =<Calen...
Cal. Convergence Tim... =<"Cal. ...

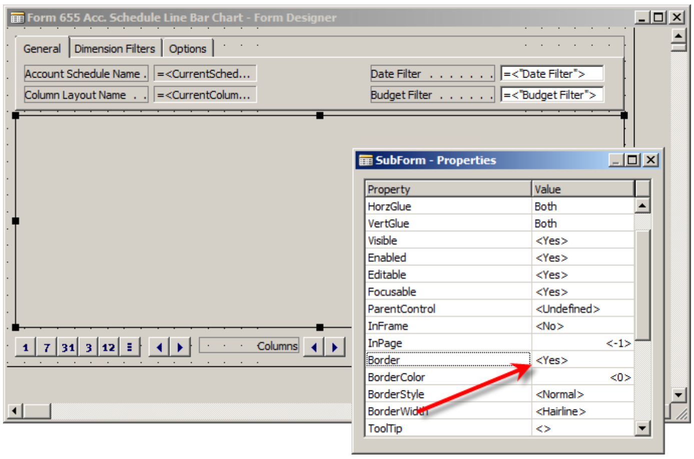
Property	Value
VertGlue	<Top>
Visible	<Yes>
ParentControl	69
InFrame	<No>
InPage	3
InColumnHeading	<No>
Caption	Customized Calendar
CaptionML	ENU=Customized Calendar;...
HorzAlign	<General>
VertAlign	<Center>
ForeColor	<0>
BackColor	<13160660>
BackTransparent	<Yes>
Border	<No>

Should be: This caption should be moved to the textbox.

18.3.17. Check No Border on SubForm

Checks if there is no border around the subform

Comment: Remove border



Account Schedule Name . =<CurrentSched...
Column Layout Name . . =<CurrentColum...
Date Filter . . . . =<"Date Filter">
Budget Filter . . . . . . =<"Budget Filter">

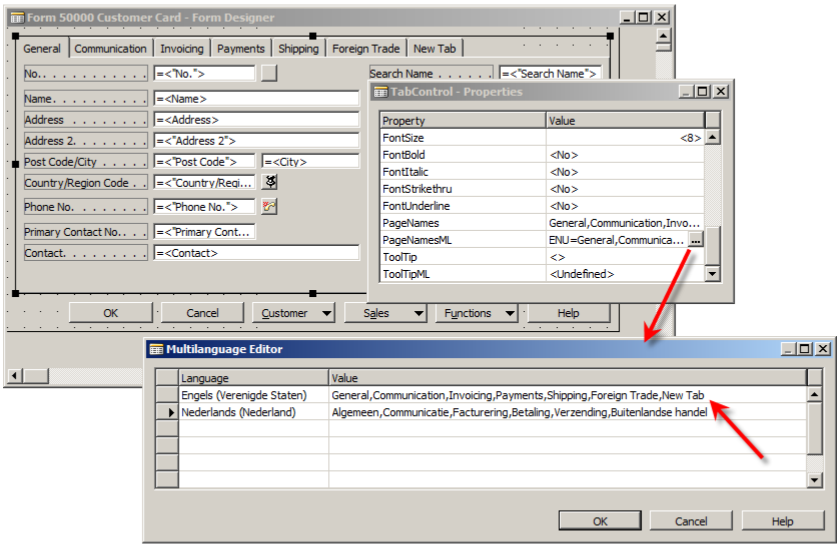
Property	Value
HorzGlue	Both
VertGlue	Both
Visible	<Yes>
Enabled	<Yes>
Editable	<Yes>
Focusable	<Yes>
ParentControl	<Undefined>
InFrame	<No>
InPage	<-1>
Border	<Yes>
BorderColor	<0>
BorderStyle	<Normal>
BorderWidth	<Hairline>
ToolTip	<>

Should be: The border must be removed for this subform.

18.3.18. Check No. of Tabs in TabControl

Checks if the number of PageNamesML per language is equal to the number of PageNames.

Comment: No. of tabs not the same



No. =<"No.">
Name =<Name>
Address =<Address>
Address 2 =<"Address 2">
Post Code/City =<"Post Code"> =<City>
Country/Region Code =<"Country/Regi…
Phone No. =<"Phone No.">
Primary Contact No. =<"Primary Cont…
Contact =<Contact>
Search Name =<"Search Name">

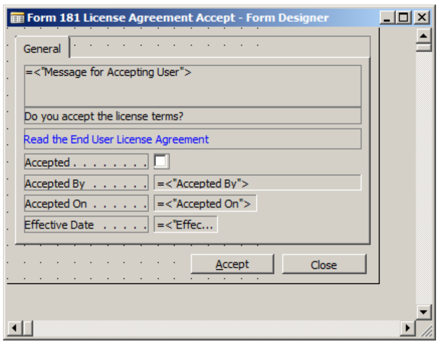
Property	Value
FontSize	<8>
FontBold	<No>
FontItalic	<No>
FontStrikethru	<No>
FontUnderline	<No>
PageNames	General,Communication,Invo…
PageNamesML	ENU=General,Communica…
ToolTip	<>
ToolTipML	<Undefined>

Language	Value
Engels (Verenigde Staten)	General,Communication,Invoicing,Payments,Shipping,Foreign Trade,New Tab
Nederlands (Nederland)	Algemeen,Communicatie,Facturering,Betaling,Verzending,Buitenlandse handel

18.3.19. Check Missing Help Button

Checks if on forms with a tablebox, matrixbox and/or tabcontrol the Help button is in the right corner.

Comment: Add help button

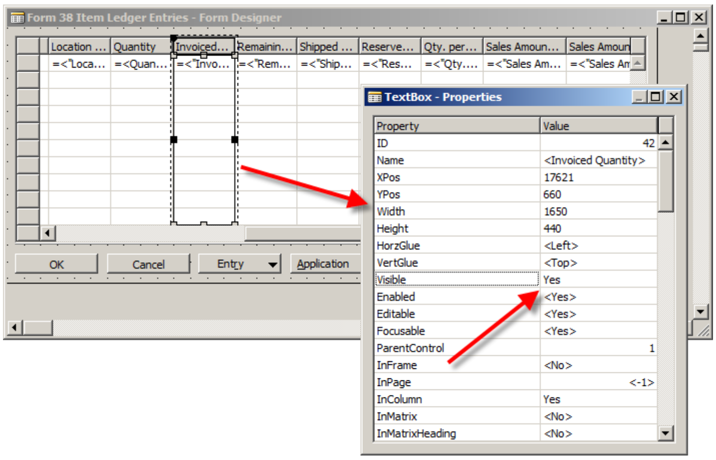


=<"Message for Accepting User">
Do you accept the license terms?
Read the End User License Agreement
Accepted ☐
Accepted By =<"Accepted By">
Accepted On =<"Accepted On">
Effective Date =<"Effec…

18.3.20. Check Unnecessary Properties

Checks for properties which are not needed because they are the same as the default.

Comment: Remove property: 'Yes' is default for 'Visible'



Form 38 Item Ledger Entries - Form Designer

TextBox - Properties

Property	Value
ID	42
Name	<Invoiced Quantity>
XPos	17621
YPos	660
Width	1650
Height	440
HorzGlue	<Left>
VertGlue	<Top>
Visible	Yes
Enabled	<Yes>
Editable	<Yes>
Focusable	<Yes>
ParentControl	1
InFrame	<No>
InPage	<-1>
InColumn	Yes
InMatrix	<No>
InMatrixHeading	<No>

18.3.21. Check Usage Wrong Keys

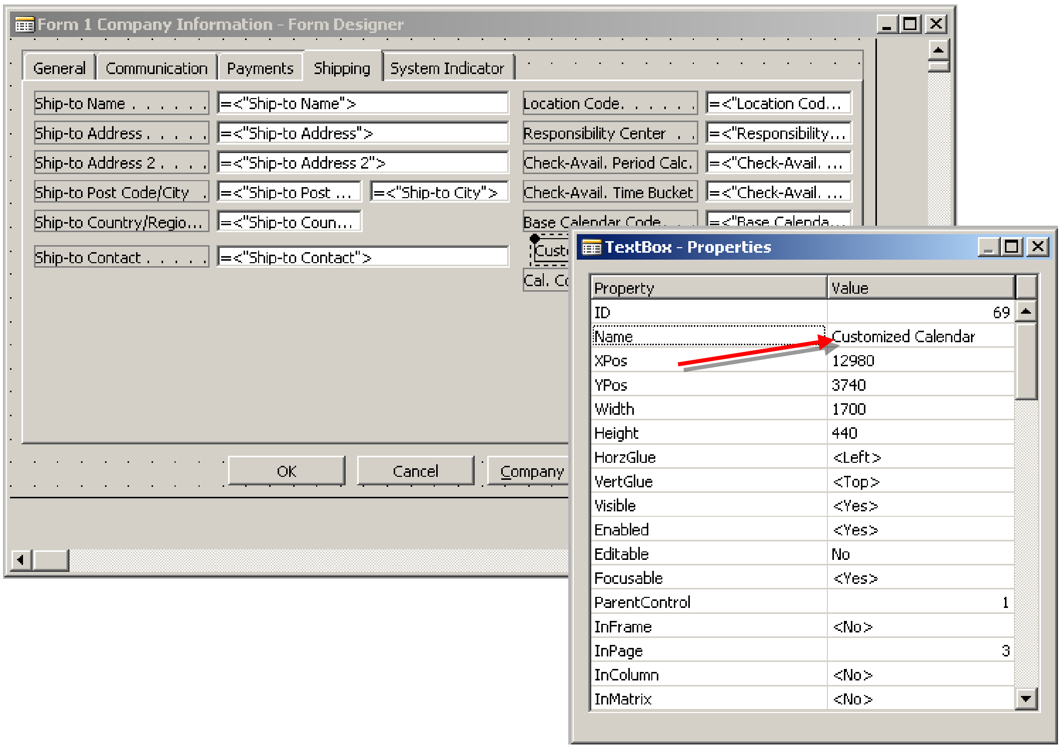
Checks if in a SourceTableView, SubFormView, RunFormView etc. non existing key is used.

Comment: Not Existing Key

18.3.22. Check Unused Control Names

Checks if a named control is being used.

Comment: Remove control name



Form 1 Company Information - Form Designer

Ship-to Name	=<"Ship-to Name">
Ship-to Address	=<"Ship-to Address">
Ship-to Address 2	=<"Ship-to Address 2">
Ship-to Post Code/City	=<"Ship-to Post … =<"Ship-to City">
Ship-to Country/Regio…	=<"Ship-to Coun…
Ship-to Contact	=<"Ship-to Contact">
Location Code.	=<"Location Cod…
Responsibility Center .	=<"Responsibility…
Check-Avail. Period Calc.	=<"Check-Avail. …
Check-Avail. Time Bucket	=<"Check-Avail. …
Base Calendar Code . .	=<"Base Calenda…
Cal. Co	Custo

TextBox - Properties

Property	Value
ID	69
Name	Customized Calendar
XPos	12980
YPos	3740
Width	1700
Height	440
HorzGlue	<Left>
VertGlue	<Top>
Visible	<Yes>
Enabled	<Yes>
Editable	No
Focusable	<Yes>
ParentControl	1
InFrame	<No>
InPage	3
InColumn	<No>
InMatrix	<No>

18.4. Data Checks

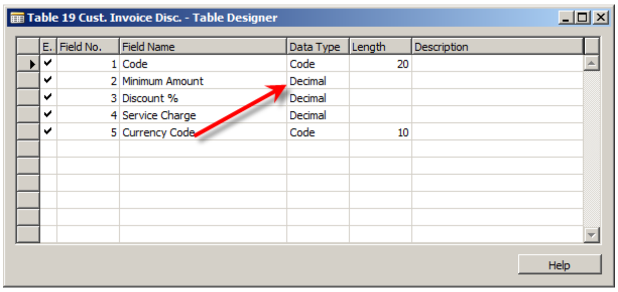
Code	Name
001	Check Key Fields are Integer, Code or Option
002	Check NotBlank on Key Fields
003	Check Testfield on Key Fields
004	Check Flowfields Not Editable
005	Check No. of Options in Field
006	Check 'SETRANGE' on Form
007	Check Primary Key in Table Relation
008	Check Field Types in Relations

18.4.1. Check Key Fields are Integer, Code, Option or Date

Checks if keyfields are of type:

- Integer
- Code
- Option
- Date

Comment: Try to avoid 'Decimal' in primary key.



E.	Field No.	Field Name	Data Type	Length	Description
✓	1	Code	Code	20	
✓	2	Minimum Amount	Decimal		
✓	3	Discount %	Decimal		
✓	4	Service Charge	Decimal		
✓	5	Currency Code	Code	10	

18.4.2. Check NotBlank on Key Fields

Checks if property of keyfield is set to NotBlank. This check is only done on tables with one primary keyfield of type code.

Comment: Add NotBlank property

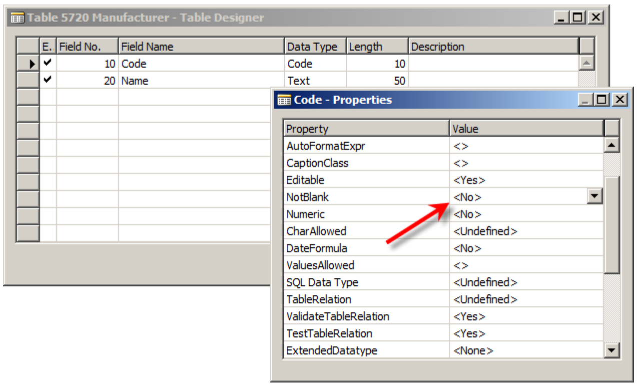


Table 5720 Manufacturer - Table Designer

E.	Field No.	Field Name	Data Type	Length	Description
✓	10	Code	Code	10	
✓	20	Name	Text	50	

Code - Properties

Property	Value
AutoFormatExpr	<>
CaptionClass	<>
Editable	<Yes>
NotBlank	<No>
Numeric	<No>
CharAllowed	<Undefined>
DateFormula	<No>
ValuesAllowed	<>
SQL Data Type	<Undefined>
TableRelation	<Undefined>
ValidateTableRelation	<Yes>
TestTableRelation	<Yes>
ExtendedDatatype	<None>

18.4.3. Check Testfield on Key Fields

Checks if TESTFIELD function is set on key fields in the OnInsert trigger of the table. This check is only done on tables with one primary key field of type code.

Comment: Add TESTFIELD(Code);

18.4.4. Check FlowFields Not Editable

Checks if the Editable property of flowfields is set to No.

Comment: Make not editable

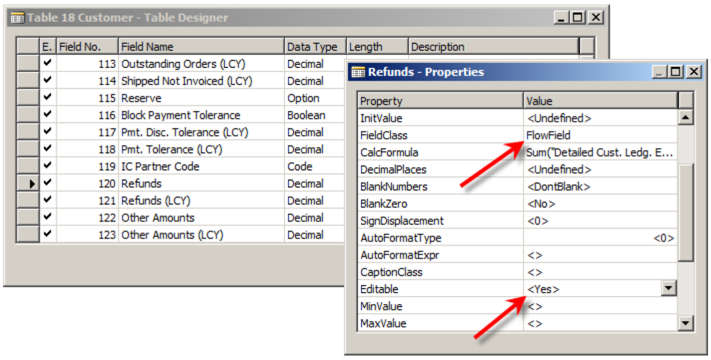


Table 18 Customer - Table Designer

E.	Field No.	Field Name	Data Type	Length	Description
✓	113	Outstanding Orders (LCY)	Decimal		
✓	114	Shipped Not Invoiced (LCY)	Decimal		
✓	115	Reserve	Option		
✓	116	Block Payment Tolerance	Boolean		
✓	117	Pmt. Disc. Tolerance (LCY)	Decimal		
✓	118	Pmt. Tolerance (LCY)	Decimal		
✓	119	IC Partner Code	Code		
✓	120	Refunds	Decimal		
✓	121	Refunds (LCY)	Decimal		
✓	122	Other Amounts	Decimal		
✓	123	Other Amounts (LCY)	Decimal		

Refunds - Properties

Property	Value
InitValue	<Undefined>
FieldClass	FlowField
CalcFormula	Sum("Detailed Cust. Ledg. E...
DecimalPlaces	<Undefined>
BlankNumbers	<DontBlank>
BlankZero	<No>
SignDisplacement	<0>
AutoFormatType	<0>
AutoFormatExpr	<>
CaptionClass	<>
Editable	<Yes>
MinValue	<>
MaxValue	<>

18.4.5. Check No. of Options in Field

Checks if an option in a OptionCaptionML is missing.

Comment: No. of options not the same

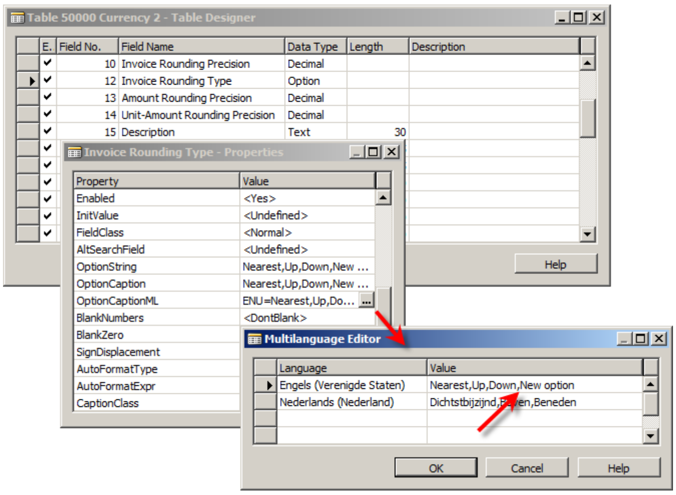


Table 50000 Currency 2 - Table Designer
Invoice Rounding Type - Properties
Multilanguage Editor
Help
OK
Cancel
Help

E.	Field No.	Field Name	Data Type	Length	Description
✓	10	Invoice Rounding Precision	Decimal		
✓	12	Invoice Rounding Type	Option		
✓	13	Amount Rounding Precision	Decimal		
✓	14	Unit-Amount Rounding Precision	Decimal		
✓	15	Description	Text	30	

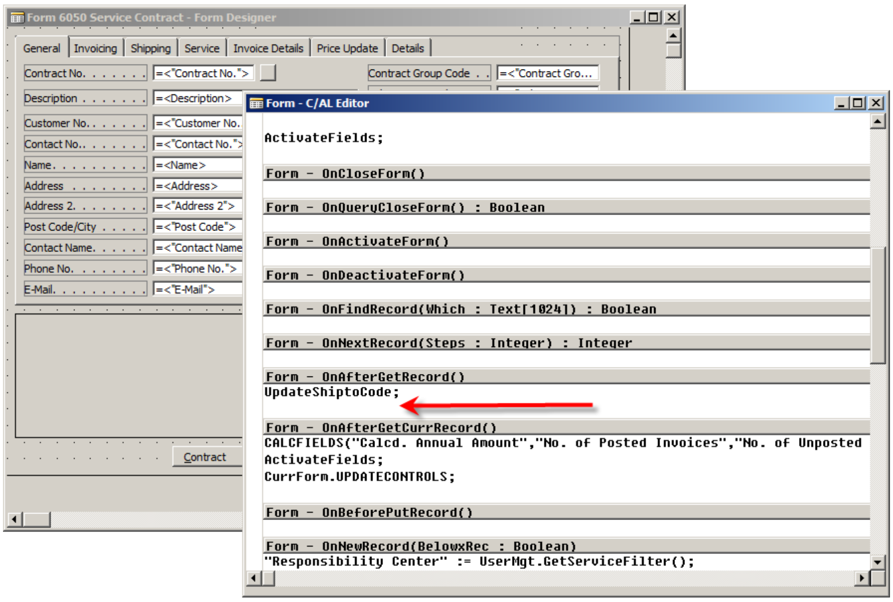
Property	Value
Enabled	<Yes>
InitValue	<Undefined>
FieldClass	<Normal>
AltSearchField	<Undefined>
OptionString	Nearest,Up,Down,New ...
OptionCaption	Nearest,Up,Down,New ...
OptionCaptionML	ENU=Nearest,Up,Do...
BlankNumbers	<DontBlank>
BlankZero	
SignDisplacement	
AutoFormatType	
AutoFormatExpr	
CaptionClass	

Language	Value
Engels (Verenigde Staten)	Nearest,Up,Down,New option
Nederlands (Nederland)	Dichtstbijzijnd,Boven,Beneden

18.4.6. Check 'SETRANGE' on Form

Checks if function SETRANGE("Primary Key"); is in the OnAfterGetRecord trigger. This check is only done on forms with a TabControl and a sourcetable.

Comment: Add SETRANGE("Primary Key");



Contract No.	=<"Contract No.">
Description	=<Description>
Customer No.	=<"Customer No.
Contact No.	=<"Contact No.">
Name	=<Name>
Address	=<Address>
Address 2	=<"Address 2">
Post Code/City	=<"Post Code">
Contact Name	=<"Contact Name
Phone No.	=<"Phone No.">
E-Mail	=<"E-Mail">
Contract Group Code	=<"Contract Gro...

```
ActivateFields;

Form - OnCloseForm()

Form - OnQueryCloseForm() : Boolean

Form - OnActivateForm()

Form - OnDeactivateForm()

Form - OnFindRecord(Which : Text[1024]) : Boolean

Form - OnNextRecord(Steps : Integer) : Integer

Form - OnAfterGetRecord()
UpdateShiptoCode;

Form - OnAfterGetCurrRecord()
CALCFIELDS("Calcd. Annual Amount","No. of Posted Invoices","No. of Unposted
ActivateFields;
CurrForm.UPDATECONTROLS;

Form - OnBeforePutRecord()

Form - OnNewRecord(BelowxRec : Boolean)
"Responsibility Center" := UserMgt.GetServiceFilter();
```

18.4.7. Check Primary Key in Table Relation

Checks if in a table relation the primary key field is used. If it is a relation to a table with a single primary key this is not needed and should be removed.

Comment: Remove Primary Key Field 'Code' from TableRelation

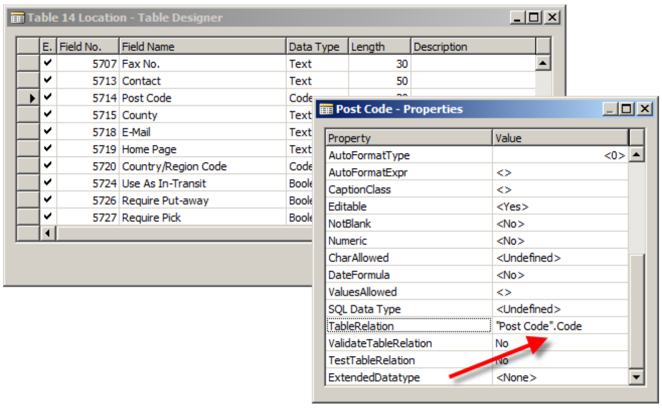


Table 14 Location - Table Designer

E.	Field No.	Field Name	Data Type	Length	Description
✓	5707	Fax No.	Text	30	
✓	5713	Contact	Text	50	
✓	5714	Post Code	Code		
✓	5715	County	Text		
✓	5718	E-Mail	Text		
✓	5719	Home Page	Text		
✓	5720	Country/Region Code	Code		
✓	5724	Use As In-Transit	Boole		
✓	5726	Require Put-away	Boole		
✓	5727	Require Pick	Boole		

Post Code - Properties

Property	Value
AutoFormatType	<0>
AutoFormatExpr	<>
CaptionClass	<>
Editable	<Yes>
NotBlank	<No>
Numeric	<No>
CharAllowed	<Undefined>
DateFormula	<No>
ValuesAllowed	<>
SQL Data Type	<Undefined>
TableRelation	"Post Code".Code
ValidateTableRelation	No
TestTableRelation	No
ExtendedDatatype	<None>

18.4.1. Check Field Types in Relations

Checks fields if related fields in table relation and FlowField CalcFormulas have same data type.

Comment: The related field in table 'G/L Account' is 'Text30'

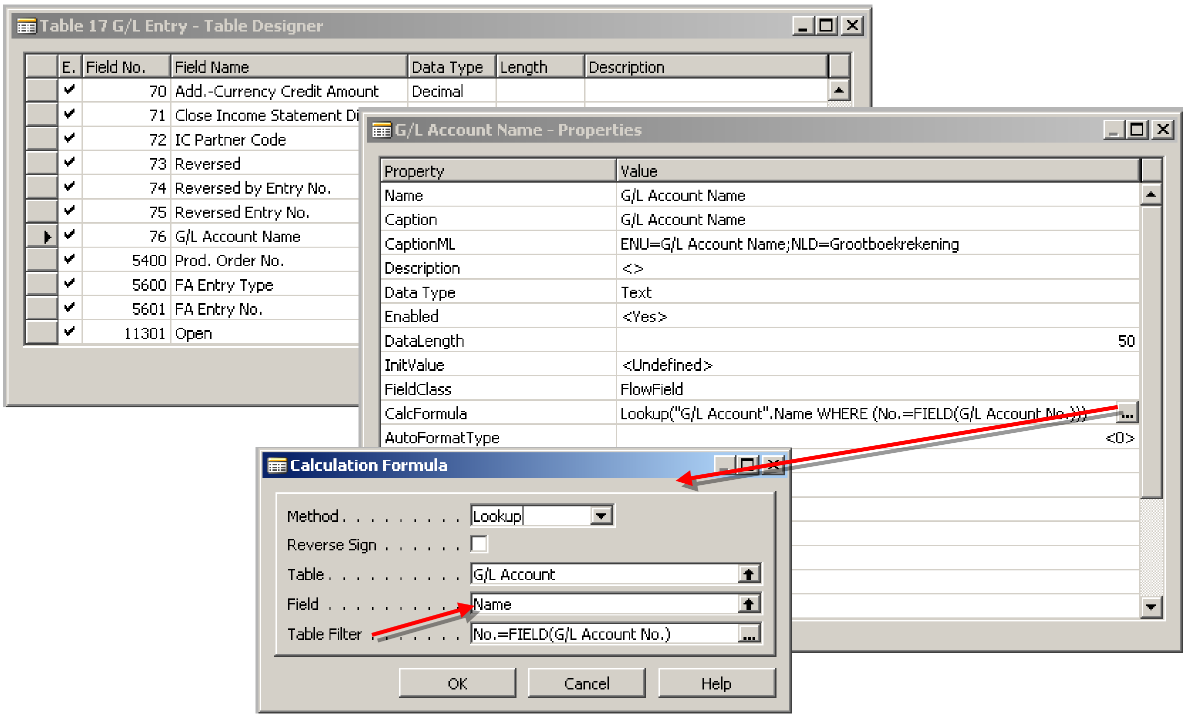


Table 17 G/L Entry - Table Designer

E.	Field No.	Field Name	Data Type	Length	Description
✓	70	Add.-Currency Credit Amount	Decimal		
✓	71	Close Income Statement Di			
✓	72	IC Partner Code			
✓	73	Reversed			
✓	74	Reversed by Entry No.			
✓	75	Reversed Entry No.			
✓	76	G/L Account Name			
✓	5400	Prod. Order No.			
✓	5600	FA Entry Type			
✓	5601	FA Entry No.			
✓	11301	Open			

G/L Account Name - Properties

Property	Value
Name	G/L Account Name
Caption	G/L Account Name
CaptionML	ENU=G/L Account Name;NLD=Grootboekrekening
Description	<>
Data Type	Text
Enabled	<Yes>
DataLength	50
InitValue	<Undefined>
FieldClass	FlowField
CalcFormula	Lookup("G/L Account".Name WHERE (No.=FIELD(G/L Account No.)))
AutoFormatType	<0>

Calculation Formula

Method . . . . Lookup

Reverse Sign . . . . . . ☐

Table . . . . G/L Account

Field . . . . Name

Table Filter . . . . No.=FIELD(G/L Account No.)

The “G/L Account Name” field on “G/L Entry” table has data type Text50, while the Name field on the “G/L Account” has data type Text30.

18.5. Naming Checks

This feature checks the correct naming of:

- Temporary records
- Variables (Local and Global)
- Object Names
- Fieldnames
- Functions

Code	Name
001	Check 'tmp' in Name of Temporary Records
002	Check Variable Names
003	Check Object Names
004	Check Field Names
005	Check Reserved Names
006	Check Function Names

18.5.1. Check 'tmp' in Name of Temporary Records

Checks if ‘tmp’ is in the name of a temporary record. A warning will be given if a record variable has ‘tmp’ in its name and is not a temporary record. Other way around when a temporary record has no ‘tmp’ in its name also a warning appears. The following name parts will be seen as temporary:

- Tmp
- Temp
- Buf
- Buffer

Comment: Add 'tmp', 'temp' or 'buffer' to variablename

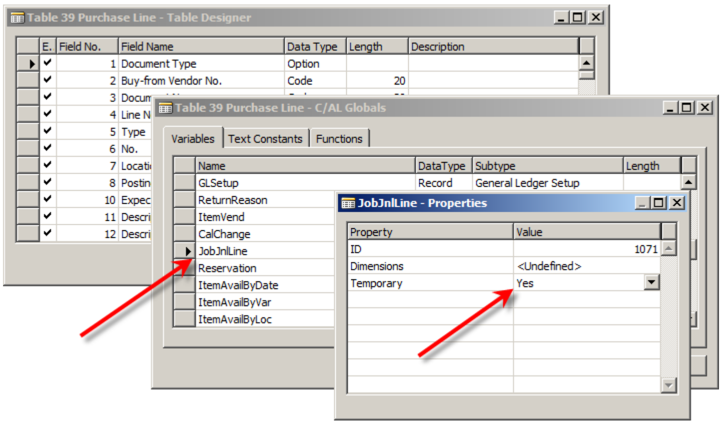


Table 39 Purchase Line - Table Designer

E.	Field No.	Field Name	Data Type	Length	Description
✓	1	Document Type	Option		
✓	2	Buy-from Vendor No.	Code	20	
✓	3	Docum			
✓	4	Line N			
✓	5	Type			
✓	6	No.			
✓	7	Locati			
✓	8	Postin			
✓	10	Expec			
✓	11	Descri			
✓	12	Descri			

Table 39 Purchase Line - C/AL Globals
Variables
Text Constants
Functions

Name	DataType	Subtype	Length
GLSetup	Record	General Ledger Setup	
ReturnReason			
ItemVend			
CalcChange			
JobJnlLine			
Reservation			
ItemAvailByDate			
ItemAvailByVar			
ItemAvailByLoc			

Property	Value
ID	1071
Dimensions	<Undefined>
Temporary	Yes

Comment: Remove 'Temp' from variablename

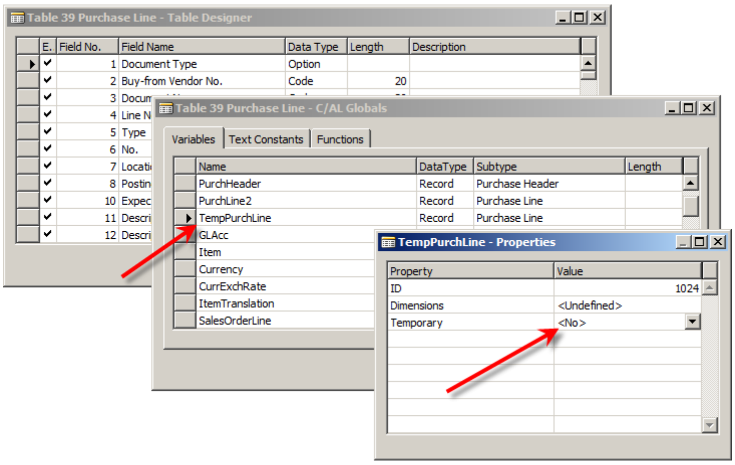


Table 39 Purchase Line - Table Designer

E.	Field No.	Field Name	Data Type	Length	Description
✓	1	Document Type	Option		
✓	2	Buy-from Vendor No.	Code	20	
✓	3	Docum			
✓	4	Line N			
✓	5	Type			
✓	6	No.			
✓	7	Locati			
✓	8	Postin			
✓	10	Expec			
✓	11	Descri			
✓	12	Descri			

Table 39 Purchase Line - C/AL Globals
Variables
Text Constants
Functions

Name	DataType	Subtype	Length
PurchHeader	Record	Purchase Header	
PurchLine2	Record	Purchase Line	
TempPurchLine	Record	Purchase Line	
GLAcc			
Item			
Currency			
CurrExchRate			
ItemTranslation			
SalesOrderLine			

Property	Value
ID	1024
Dimensions	<Undefined>
Temporary	<No>

(Example: TempCustLedgerEntry is not a temporary record so Temp must be removed from the name)

18.5.2. Check Variable Names

Variable names must start with capital letter. The first and second letter of the name can be lowercase because of the common use of prefixes.

Comment: Start a variablename with a capital

- dimensionValue: Not allowed
- pCustomer: Allowed

Also spaces in the variable name will be noticed.

Comment: Remove space from variablename

18.5.3. Check Object Names

Object names have the same check as variable names except for spaces. Double spaces however are noticed.

18.5.4. Check Field Names

Same checks as Object Names.

18.5.5. Check Reserved Names

Checks if variable-, field- and function names are conflicting with existing functions and commands.

Comment: 'CopyFilters' is a reserved command.

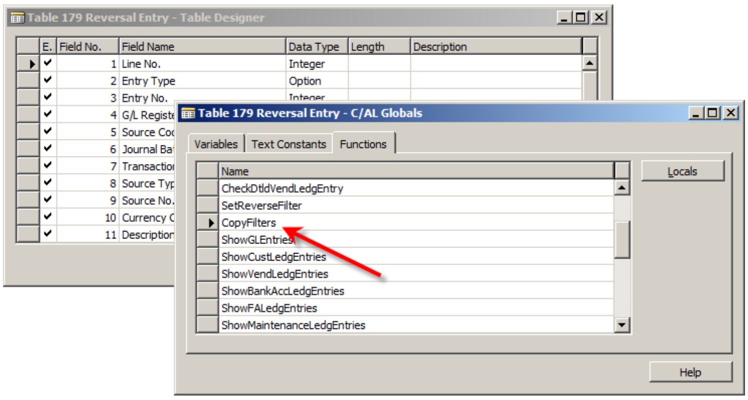


Table 179 Reversal Entry - Table Designer
Table 179 Reversal Entry - C/AL Globals
Variables
Text Constants
Functions
Locals
Help

E.	Field No.	Field Name	Data Type	Length	Description
✓	1	Line No.	Integer		
✓	2	Entry Type	Option		
✓	3	Entry No.	Integer		
✓	4	G/L Registe			
✓	5	Source Coc			
✓	6	Journal Ba			
✓	7	Transactio			
✓	8	Source Typ			
✓	9	Source No.			
✓	10	Currency C			
✓	11	Description			

Name
CheckDtldVendLedgEntry
SetReverseFilter
CopyFilters
ShowGLEntries
ShowCustLedgEntries
ShowVendLedgEntries
ShowBankAccLedgEntries
ShowFALedgEntries
ShowMaintenanceLedgEntries

18.5.6. Check Function Names

Checks if a function name contains spaces.

Comment: Remove space from function name

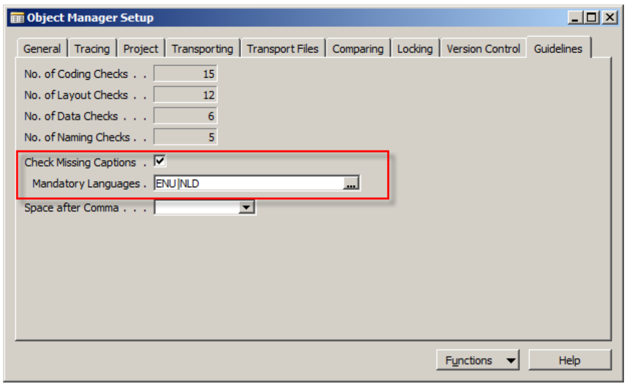
```
PROCEDURE "Export XML"@4();
```

Should be:

```
PROCEDURE ExportXML@4();
```

18.6. Caption Checks

Checks all objects for missing captions and redundant spaces in captions.
To enable this feature check mark “Check Missing Captions” on the “Object Manager Setup” window. Caption checks will only be executed in the range of languages the user has defined in the setup.

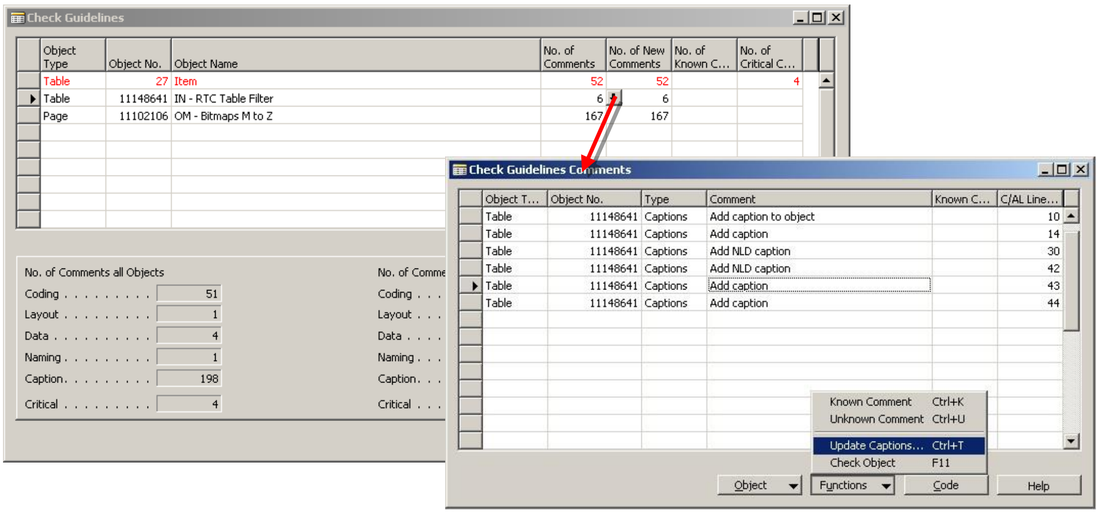


No. of Coding Checks . . 15
No. of Layout Checks . . 12
No. of Data Checks . . . 6
No. of Naming Checks . . 5
Check Missing Captions . ☑
Mandatory Languages . ENU|NLD
Space after Comma . . .

Checking the caption typically can show the following comments:

- Add caption to object
  The object checked does not have any captions defined for.
- Add caption
  The field or control checked does not have any captions defined for.
- Add <language> caption
  The field or control checked does not have a caption defined for the designated language.
- Remove space from caption
  The caption contains a redundant space.

Use the “Update Captions” window to view the individual check results and update them:



Object Type	Object No.	Object Name	No. of Comments	No. of New Comments	No. of Known C…	No. of Critical C…
Table	27	Item	52	52		4
Table	11148641	IN - RTC Table Filter	6	6		
Page	11102106	OM - Bitmaps M to Z	167	167		

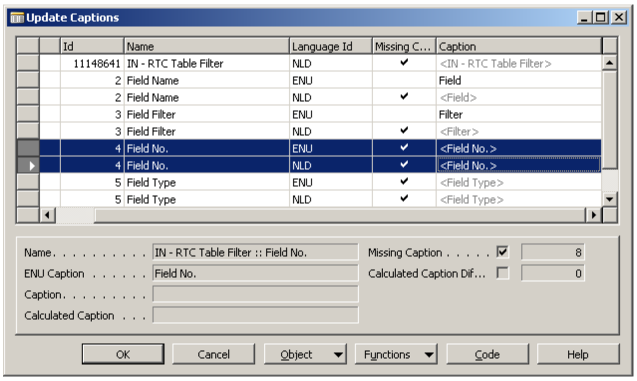
No. of Comments all Objects

Coding 51
Layout 1
Data 4
Naming 1
Caption. 198
Critical 4

No. of Comme

Coding . . .
Layout . . .
Data . . . .
Naming . . .
Caption. . .
Critical . . .

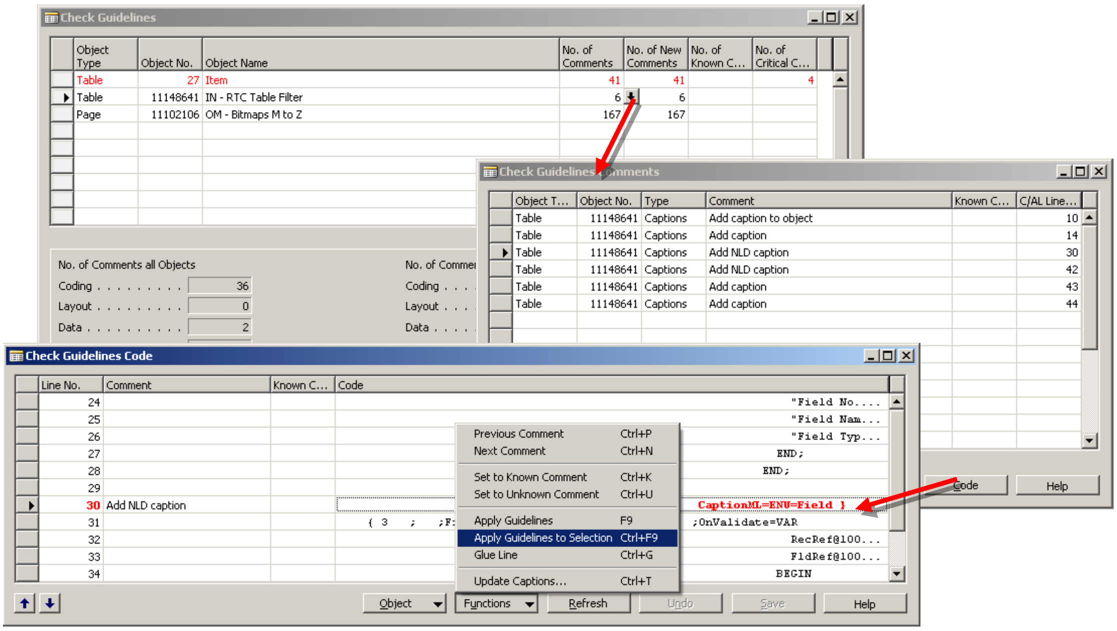
Object T…	Object No.	Type	Comment	Known C…	C/AL Line…
Table	11148641	Captions	Add caption to object		10
Table	11148641	Captions	Add caption		14
Table	11148641	Captions	Add NLD caption		30
Table	11148641	Captions	Add NLD caption		42
Table	11148641	Captions	Add caption		43
Table	11148641	Captions	Add caption		44



Id	Name	Language Id	Missing C…	Caption
11148641	IN - RTC Table Filter	NLD	✓	<IN - RTC Table Filter>
2	Field Name	ENU		Field
2	Field Name	NLD	✓	<Field>
3	Field Filter	ENU		Filter
3	Field Filter	NLD	✓	<Filter>
4	Field No.	ENU	✓	<Field No.>
4	Field No.	NLD	✓	<Field No.>
5	Field Type	ENU	✓	<Field Type>
5	Field Type	NLD	✓	<Field Type>

Name. . . . . IN - RTC Table Filter :: Field No.
ENU Caption . . . . . . Field No.
Caption. . . . .
Calculated Caption . . .
Missing Caption . . . . . ☑ 8
Calculated Caption Dif… ☐ 0

If a translation for a specific language has been defined as a translation rule you can insert a missing caption for that language using the “Apply Guidelines to Selection” feature on the “Check Guidelines Code” window.



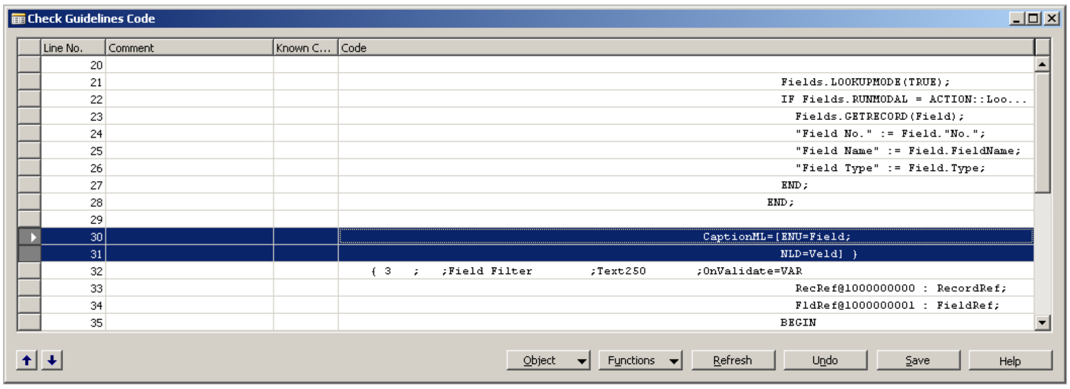
Object Type	Object No.	Object Name	No. of Comments	No. of New Comments	No. of Known C...	No. of Critical C...
Table	27	Item	41	41		4
Table	11148641	IN - RTC Table Filter	6	6		
Page	11102106	OM - Bitmaps M to Z	167	167		

No. of Comments all Objects
Coding 36
Layout 0
Data 2

No. of Commer
Coding . . .
Layout . . .
Data . . . .

Object T...	Object No.	Type	Comment	Known C...	C/AL Line...
Table	11148641	Captions	Add caption to object		10
Table	11148641	Captions	Add caption		14
Table	11148641	Captions	Add NLD caption		30
Table	11148641	Captions	Add NLD caption		42
Table	11148641	Captions	Add caption		43
Table	11148641	Captions	Add caption		44

Line No.	Comment	Known C...	Code
24			"Field No....
25			"Field Nam...
26			"Field Typ...
27			END;
28			END;
29			
30	Add NLD caption		CaptionML=ENU=Field }
31			{ 3 ; ;F: ;OnValidate=VAR
32			RecRef@100...
33			FldRef@100...
34			BEGIN



Line No.	Comment	Known C...	Code
20			
21			Fields.LOOKUPMODE(TRUE);
22			IF Fields.RUNMODAL = ACTION::Loo...
23			Fields.GETRECORD(Field);
24			"Field No." := Field."No.";
25			"Field Name" := Field.FieldName;
26			"Field Type" := Field.Type;
27			END;
28			END;
29			
30			CaptionML=[ENU=Field;
31			NLD=Veld] }
32			{ 3 ; ;Field Filter ;Text250 ;OnValidate=VAR
33			RecRef@1000000000 : RecordRef;
34			FldRef@1000000001 : FieldRef;
35			BEGIN

See also section 26.1.2 - Rules.

If a lot of captions are missing you can add those with the translation tool. For more information see chapter 26 - Translation Tool.

18.7. Space after Comma

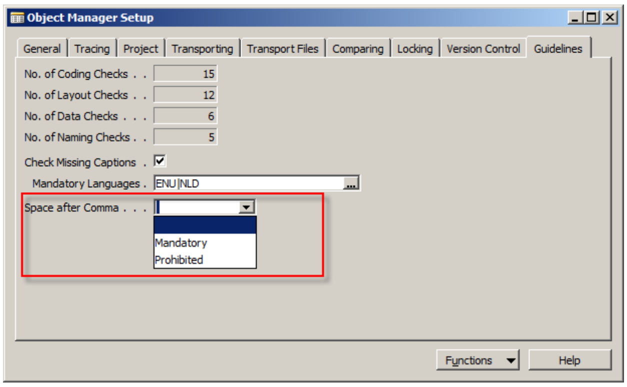
The guidelines of NAV says that you cannot put a space after a comma. Many developers like to place there a comma anyway. In the setup you can choose which method the Object Manager has to check. If you choose the option Mandatory your code have to look like this:

```
IF NOT PrinterSelection.GET(USERID, ReportID) THEN
  IF NOT PrinterSelection.GET('', ReportID) THEN
    IF NOT PrinterSelection.GET(USERID, 0) THEN
      IF PrinterSelection.GET('', 0) THEN;
```

If you choose the Prohibited setting your code has to look like this:

```
IF NOT PrinterSelection.GET(USERID,ReportID) THEN
  IF NOT PrinterSelection.GET('',ReportID) THEN
    IF NOT PrinterSelection.GET(USERID,0) THEN
IF PrinterSelection.GET('',0) THEN;
```

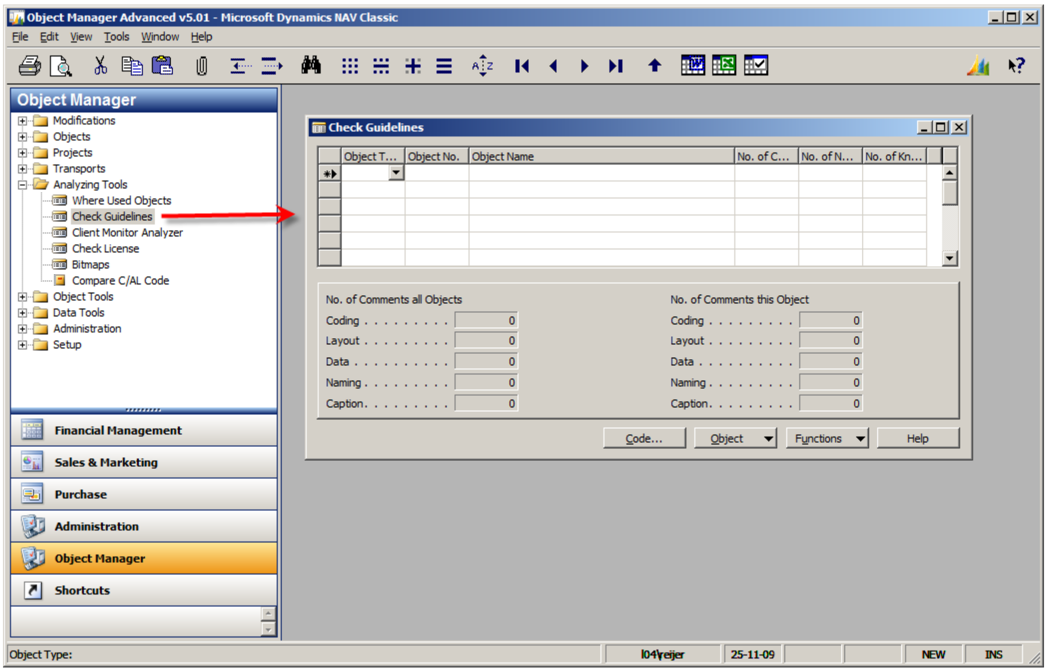
It is also possible to leave this option blank. The Object Manager will not check for spaces after comma's.



No. of Coding Checks . . 15
No. of Layout Checks . . 12
No. of Data Checks . . . 6
No. of Naming Checks . . 5
Check Missing Captions . ☑
Mandatory Languages . ENU|NLD
Space after Comma . . .

18.8. Check Guidelines

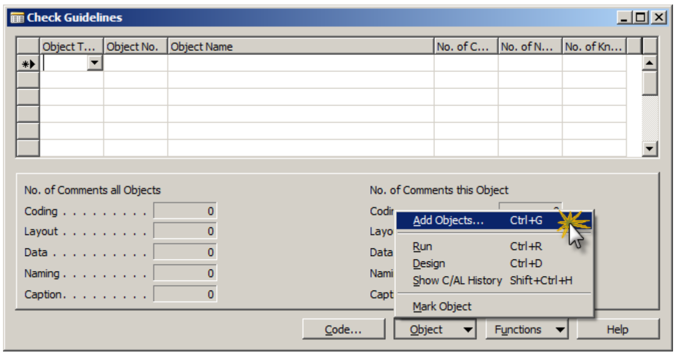
The Check Guidelines tool can be found in menu Analyzing Tools - Check Guidelines.



No. of Comments all Objects
Coding 0
Layout 0
Data 0
Naming 0
Caption. 0

No. of Comments this Object
Coding 0
Layout 0
Data 0
Naming 0
Caption. 0

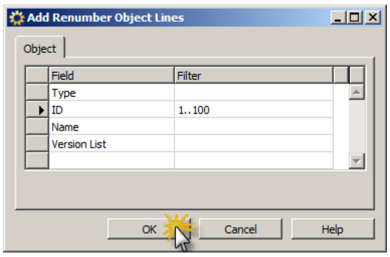
Add the objects you want to check. If you have a range of objects you want to add you also can use the function "Add Objects".


Check Guidelines
Object T... | Object No. | Object Name | No. of C... | No. of N... | No. of Kn...
Add Objects... Ctrl+G
Run Ctrl+R
Design Ctrl+D
Show C/AL History Shift+Ctrl+H
Mark Object
Code...
Object
Functions
Help

No. of Comments all Objects
Coding 0
Layout 0
Data 0
Naming 0
Caption. 0

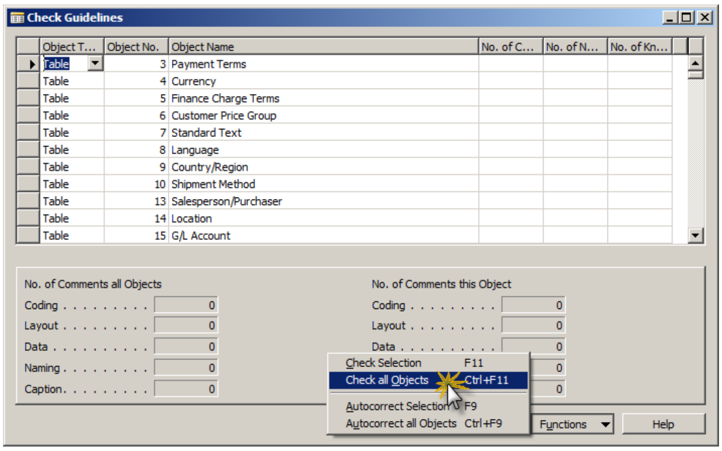
No. of Comments this Object

Enter your filters:



Field	Filter
Type	
ID	1..100
Name	
Version List	

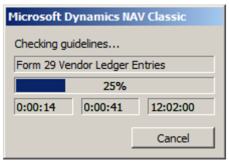
The objects will be added in the worksheet. To check the object(s) start function Check Selection or Check all Objects.



Object T...	Object No.	Object Name	No. of C...	No. of N...	No. of Kn...
Table	3	Payment Terms			
Table	4	Currency			
Table	5	Finance Charge Terms			
Table	6	Customer Price Group			
Table	7	Standard Text			
Table	8	Language			
Table	9	Country/Region			
Table	10	Shipment Method			
Table	13	Salesperson/Purchaser			
Table	14	Location			
Table	15	G/L Account			

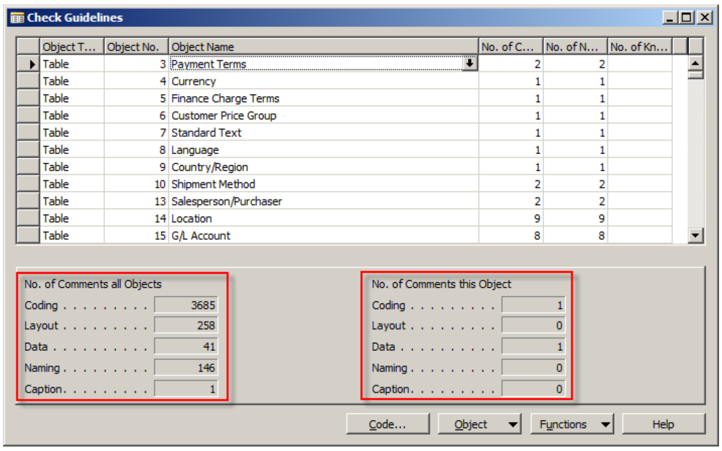
No. of Comments all Objects
Coding 0
Layout 0
Data 0
Naming 0
Caption. 0

No. of Comments this Object
Coding 0
Layout 0
Data 0
0
0



Checking guidelines...
Form 29 Vendor Ledger Entries
25%
0:00:14 0:00:41 12:02:00

When finished all criteria you selected in the setup is checked:



Object T…	Object No.	Object Name	No. of C…	No. of N…	No. of Kn…
Table	3	Payment Terms	2	2	
Table	4	Currency	1	1	
Table	5	Finance Charge Terms	1	1	
Table	6	Customer Price Group	1	1	
Table	7	Standard Text	1	1	
Table	8	Language	1	1	
Table	9	Country/Region	1	1	
Table	10	Shipment Method	2	2	
Table	13	Salesperson/Purchaser	2	2	
Table	14	Location	9	9	
Table	15	G/L Account	8	8	

No. of Comments all Objects

Coding 3685
Layout 258
Data 41
Naming 146
Caption. 1

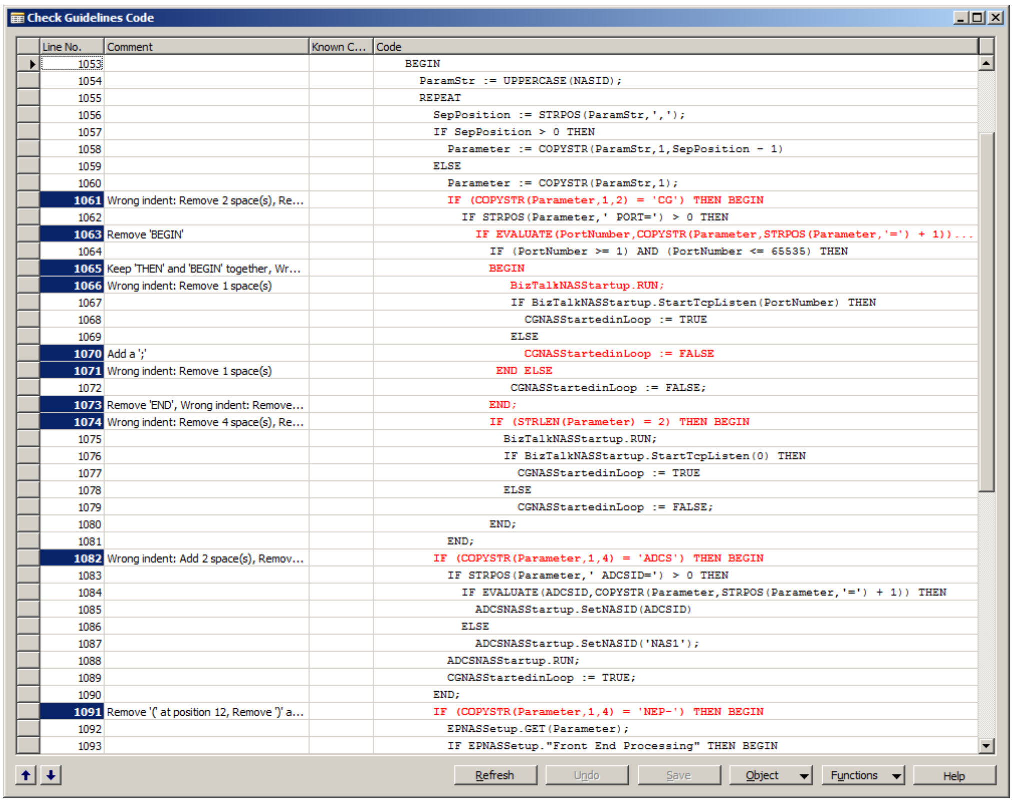
No. of Comments this Object

Coding 1
Layout 0
Data 1
Naming 0
Caption. 0

The No. of Comments per Object can also be seen in the lines.

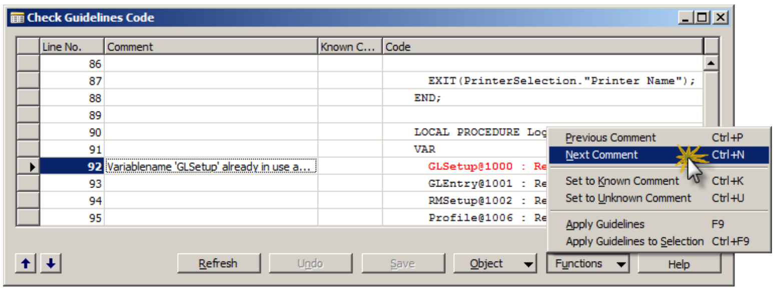
By using the Code button in this form on the selected object you can go to the C/AL where the Check Guidelines tool found the comments.

Here you can see the comments on a piece of code of codeunit 1 – Application Management.



```
Line No. Comment                                    Known C… Code
1053                                                          BEGIN
1054                                                            ParamStr := UPPERCASE(NASID);
1055                                                            REPEAT
1056                                                              SepPosition := STRPOS(ParamStr,',');
1057                                                              IF SepPosition > 0 THEN
1058                                                                Parameter := COPYSTR(ParamStr,1,SepPosition - 1)
1059                                                              ELSE
1060                                                                Parameter := COPYSTR(ParamStr,1);
1061     Wrong indent: Remove 2 space(s), Re…                       IF (COPYSTR(Parameter,1,2) = 'CG') THEN BEGIN
1062                                                                  IF STRPOS(Parameter,' PORT=') > 0 THEN
1063     Remove 'BEGIN'                                                 IF EVALUATE(PortNumber,COPYSTR(Parameter,STRPOS(Parameter,'=') + 1))...
1064                                                                      IF (PortNumber >= 1) AND (PortNumber <= 65535) THEN
1065     Keep 'THEN' and 'BEGIN' together, Wr…                            BEGIN
1066     Wrong indent: Remove 1 space(s)                                     BizTalkNASStartup.RUN;
1067                                                                         IF BizTalkNASStartup.StartTcpListen(PortNumber) THEN
1068                                                                           CGNASStartedinLoop := TRUE
1069                                                                         ELSE
1070     Add a ';'                                                           CGNASStartedinLoop := FALSE
1071     Wrong indent: Remove 1 space(s)                                   END ELSE
1072                                                                         CGNASStartedinLoop := FALSE;
1073     Remove 'END', Wrong indent: Remove…                              END;
1074     Wrong indent: Remove 4 space(s), Re…                             IF (STRLEN(Parameter) = 2) THEN BEGIN
1075                                                                        BizTalkNASStartup.RUN;
1076                                                                        IF BizTalkNASStartup.StartTcpListen(0) THEN
1077                                                                          CGNASStartedinLoop := TRUE
1078                                                                        ELSE
1079                                                                          CGNASStartedinLoop := FALSE;
1080                                                                      END;
1081                                                                END;
1082     Wrong indent: Add 2 space(s), Remov…                     IF (COPYSTR(Parameter,1,4) = 'ADCS') THEN BEGIN
1083                                                                IF STRPOS(Parameter,' ADCSID=') > 0 THEN
1084                                                                  IF EVALUATE(ADCSID,COPYSTR(Parameter,STRPOS(Parameter,'=') + 1)) THEN
1085                                                                    ADCSNASStartup.SetNASID(ADCSID)
1086                                                                  ELSE
1087                                                                    ADCSNASStartup.SetNASID('NAS1');
1088                                                                ADCSNASStartup.RUN;
1089                                                                CGNASStartedinLoop := TRUE;
1090                                                              END;
1091     Remove '(' at position 12, Remove ')' a…                 IF (COPYSTR(Parameter,1,4) = 'NEP-') THEN BEGIN
1092                                                                EPNASSetup.GET(Parameter);
1093                                                                IF EPNASSetup."Front End Processing" THEN BEGIN
```

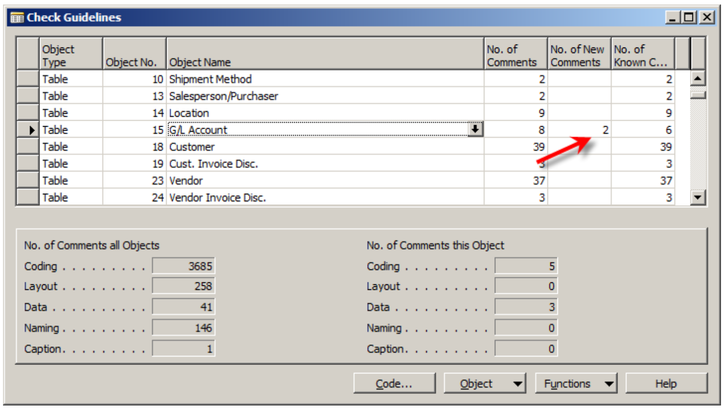
If you want to loop through all comments you can use the function button to go to previous and next comment.



Line No.	Comment	Known C...	Code
86			
87			EXIT(PrinterSelection."Printer Name");
88			END;
89			
90			LOCAL PROCEDURE Log
91			VAR
92	Variablename 'GLSetup' already in use a...		GLSetup@1000 : Re
93			GLEntry@1001 : Re
94			RMSetup@1002 : Re
95			Profile@1006 : Re

18.9. Known comments

You can set a comment as Known Comment if you don’t want to solve the comment. This is often done when the comments apply to standard NAV objects. The benefit of using the know comment option is that you can easily see a new comment because this comment is not set to ‘known’.



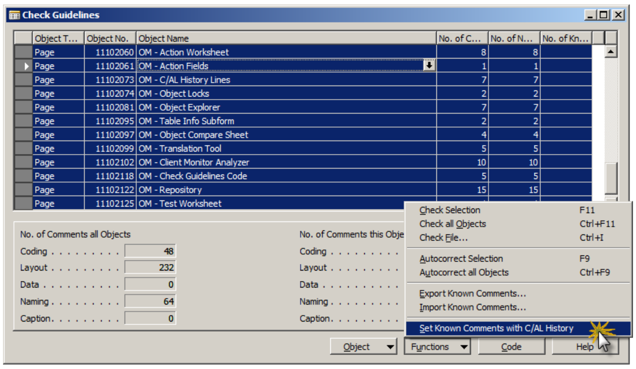
Object Type	Object No.	Object Name	No. of Comments	No. of New Comments	No. of Known C...
Table	10	Shipment Method	2		2
Table	13	Salesperson/Purchaser	2		2
Table	14	Location	9		9
Table	15	G/L Account	8	2	6
Table	18	Customer	39		39
Table	19	Cust. Invoice Disc.	3		3
Table	23	Vendor	37		37
Table	24	Vendor Invoice Disc.	3		3

No. of Comments all Objects
Coding 3685
Layout 258
Data 41
Naming 146
Caption. 1

No. of Comments this Object
Coding 5
Layout 0
Data 3
Naming 0
Caption. 0

18.9.1. Set Known Comments with C/AL History

With this function you can set all comments to known that were already present in a point of time. This can come handy when you work on a particular project for a week and you want to know which comments were newly created in this week.

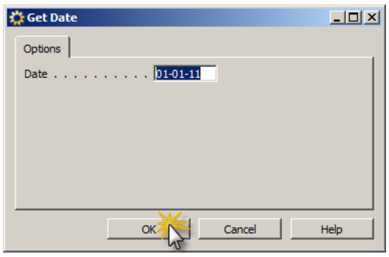


Object T...	Object No.	Object Name	No. of C...	No. of N...	No. of Kn...
Page	11102060	OM - Action Worksheet	8	8	
Page	11102061	OM - Action Fields	1	1	
Page	11102073	OM - C/AL History Lines	7	7	
Page	11102074	OM - Object Locks	2	2	
Page	11102081	OM - Object Explorer	7	7	
Page	11102095	OM - Table Info Subform	2	2	
Page	11102097	OM - Object Compare Sheet	4	4	
Page	11102099	OM - Translation Tool	5	5	
Page	11102102	OM - Client Monitor Analyzer	10	10	
Page	11102118	OM - Check Guidelines Code	5	5	
Page	11102122	OM - Repository	15	15	
Page	11102125	OM - Test Worksheet			

No. of Comments all Objects
Coding 48
Layout 232
Data 0
Naming 64
Caption. 0

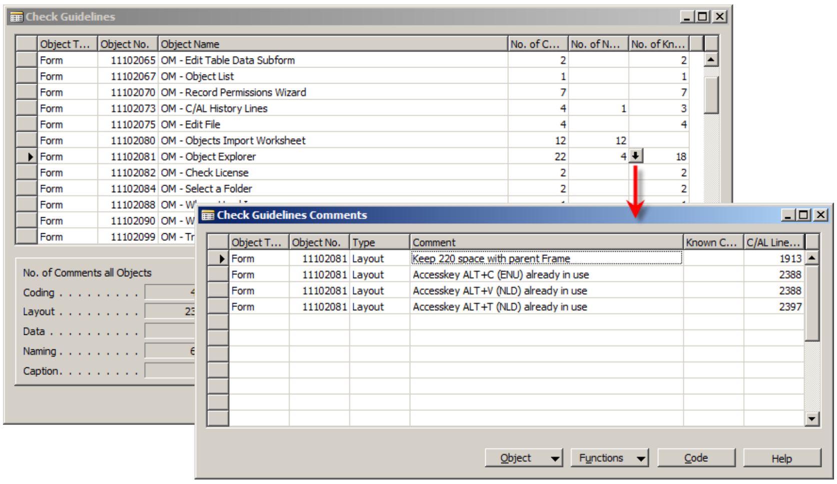
No. of Comments this Obje
Coding . . . .
Layout . . . .
Data . . . .
Naming . . . .
Caption. . . . .

Set the date and press OK.



Date 01-01-11

All comments that were already in the objects at January 1st will be set the known. The comments that will remain as new are the comments that are newly created since January 1st.



Object T...	Object No.	Object Name	No. of C...	No. of N...	No. of Kn...
Form	11102065	OM - Edit Table Data Subform	2		2
Form	11102067	OM - Object List	1		1
Form	11102070	OM - Record Permissions Wizard	7		7
Form	11102073	OM - C/AL History Lines	4	1	3
Form	11102075	OM - Edit File	4		4
Form	11102080	OM - Objects Import Worksheet	12	12	
Form	11102081	OM - Object Explorer	22	4	18
Form	11102082	OM - Check License	2		2
Form	11102084	OM - Select a Folder	2		2
Form	11102088	OM - W			
Form	11102090	OM - W			
Form	11102099	OM - Tr			

No. of Comments all Objects

Coding 4

Layout 23

Data . . . .

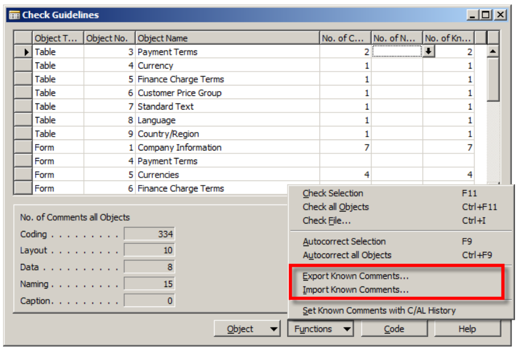
Naming 6

Caption. . . . .

Object T...	Object No.	Type	Comment	Known C...	C/AL Line...
Form	11102081	Layout	Keep 220 space with parent Frame		1913
Form	11102081	Layout	Accesskey ALT+C (ENU) already in use		2388
Form	11102081	Layout	Accesskey ALT+V (NLD) already in use		2388
Form	11102081	Layout	Accesskey ALT+T (NLD) already in use		2397

18.9.2. Import and Export Known Comments

With this function you can im- and export all your known comments from or to other databases. So you don't have to set them again in a new database.



Object T...	Object No.	Object Name	No. of C...	No. of N...	No. of Kn...
Table	3	Payment Terms	2		2
Table	4	Currency	1		1
Table	5	Finance Charge Terms	1		1
Table	6	Customer Price Group	1		1
Table	7	Standard Text	1		1
Table	8	Language	1		1
Table	9	Country/Region	1		1
Form	1	Company Information	7		7
Form	4	Payment Terms			
Form	5	Currencies	4		4
Form	6	Finance Charge Terms			

No. of Comments all Objects

Coding 334

Layout 10

Data 8

Naming 15

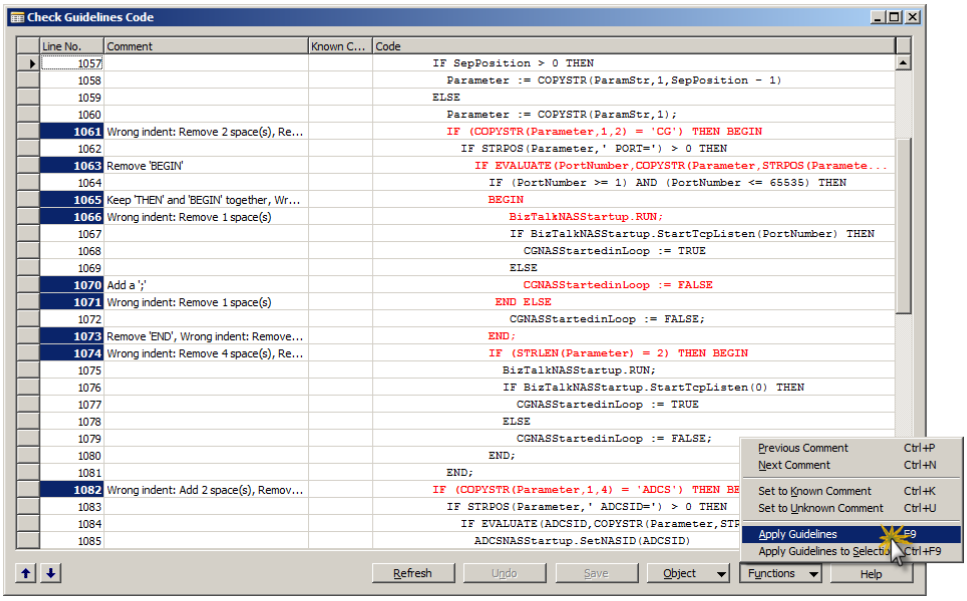
Caption. 0

18.10. Auto Apply Guidelines

You have the ability to let the Object Manager correct most of the check coding guidelines comments. Beware that automatic changing by the tool isn't possible for all guideline checks.

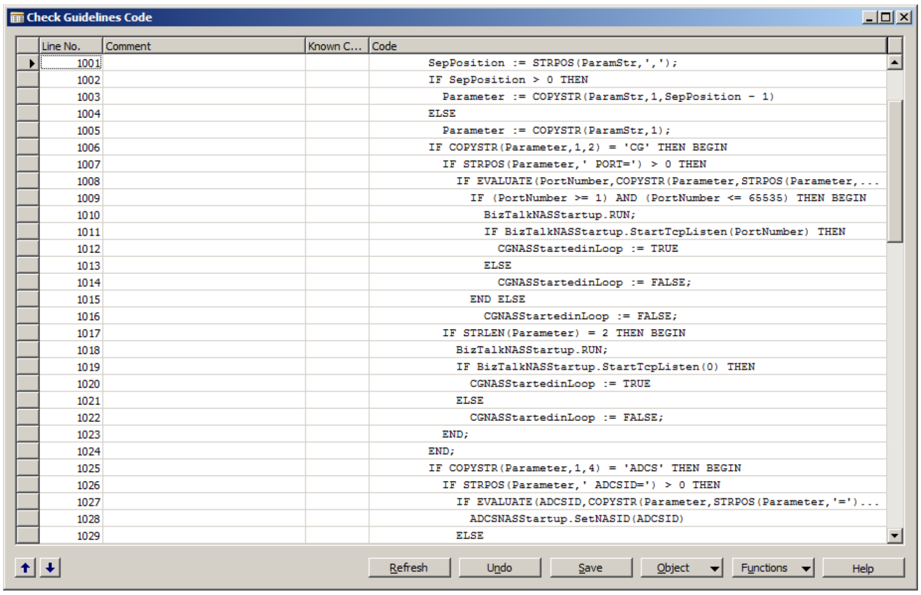
NOTE: Only auto apply guidelines in a development environment and test the changes before transporting your objects to the production environment.

Before:



```
Line No. Comment                              Known C… Code
1057                                                   IF SepPosition > 0 THEN
1058                                                     Parameter := COPYSTR(ParamStr,1,SepPosition - 1)
1059                                                   ELSE
1060                                                     Parameter := COPYSTR(ParamStr,1);
1061     Wrong indent: Remove 2 space(s), Re…            IF (COPYSTR(Parameter,1,2) = 'CG') THEN BEGIN
1062                                                       IF STRPOS(Parameter,' PORT=') > 0 THEN
1063     Remove 'BEGIN'                                      IF EVALUATE(PortNumber,COPYSTR(Parameter,STRPOS(Paramete...
1064                                                           IF (PortNumber >= 1) AND (PortNumber <= 65535) THEN
1065     Keep 'THEN' and 'BEGIN' together, Wr…                 BEGIN
1066     Wrong indent: Remove 1 space(s)                          BizTalkNASStartup.RUN;
1067                                                              IF BizTalkNASStartup.StartTcpListen(PortNumber) THEN
1068                                                                CGNASStartedinLoop := TRUE
1069                                                              ELSE
1070     Add a ';'                                                  CGNASStartedinLoop := FALSE
1071     Wrong indent: Remove 1 space(s)                        END ELSE
1072                                                              CGNASStartedinLoop := FALSE;
1073     Remove 'END', Wrong indent: Remove…                   END;
1074     Wrong indent: Remove 4 space(s), Re…                  IF (STRLEN(Parameter) = 2) THEN BEGIN
1075                                                             BizTalkNASStartup.RUN;
1076                                                             IF BizTalkNASStartup.StartTcpListen(0) THEN
1077                                                               CGNASStartedinLoop := TRUE
1078                                                             ELSE
1079                                                               CGNASStartedinLoop := FALSE;
1080                                                           END;
1081                                                     END;
1082     Wrong indent: Add 2 space(s), Remov…          IF (COPYSTR(Parameter,1,4) = 'ADCS') THEN BE
1083                                                     IF STRPOS(Parameter,' ADCSID=') > 0 THEN
1084                                                       IF EVALUATE(ADCSID,COPYSTR(Parameter,STF
1085                                                         ADCSNASStartup.SetNASID(ADCSID)
```

After:



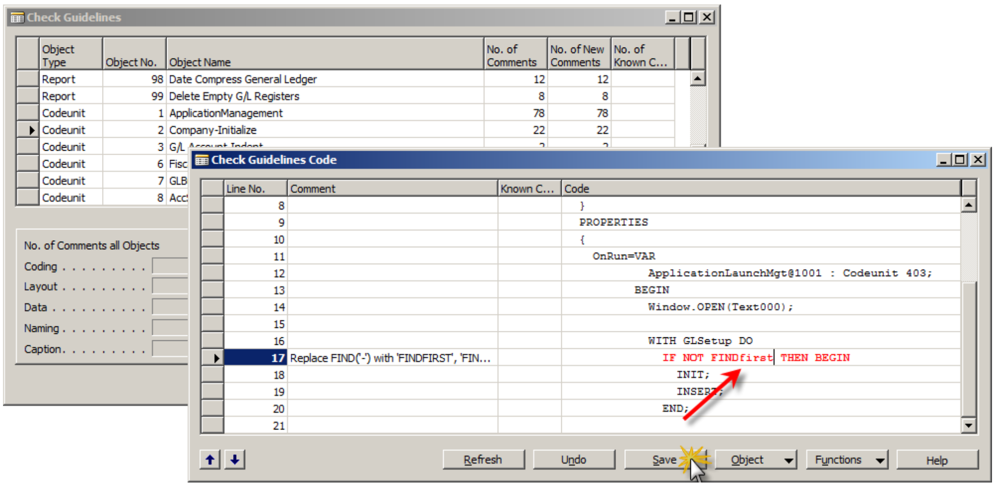
```
Line No. Comment                              Known C… Code
1001                                                   SepPosition := STRPOS(ParamStr,',');
1002                                                   IF SepPosition > 0 THEN
1003                                                     Parameter := COPYSTR(ParamStr,1,SepPosition - 1)
1004                                                   ELSE
1005                                                     Parameter := COPYSTR(ParamStr,1);
1006                                                   IF COPYSTR(Parameter,1,2) = 'CG' THEN BEGIN
1007                                                     IF STRPOS(Parameter,' PORT=') > 0 THEN
1008                                                       IF EVALUATE(PortNumber,COPYSTR(Parameter,STRPOS(Parameter,...
1009                                                         IF (PortNumber >= 1) AND (PortNumber <= 65535) THEN BEGIN
1010                                                           BizTalkNASStartup.RUN;
1011                                                           IF BizTalkNASStartup.StartTcpListen(PortNumber) THEN
1012                                                             CGNASStartedinLoop := TRUE
1013                                                           ELSE
1014                                                             CGNASStartedinLoop := FALSE;
1015                                                         END ELSE
1016                                                           CGNASStartedinLoop := FALSE;
1017                                                     IF STRLEN(Parameter) = 2 THEN BEGIN
1018                                                       BizTalkNASStartup.RUN;
1019                                                       IF BizTalkNASStartup.StartTcpListen(0) THEN
1020                                                         CGNASStartedinLoop := TRUE
1021                                                       ELSE
1022                                                         CGNASStartedinLoop := FALSE;
1023                                                     END;
1024                                                   END;
1025                                                   IF COPYSTR(Parameter,1,4) = 'ADCS' THEN BEGIN
1026                                                     IF STRPOS(Parameter,' ADCSID=') > 0 THEN
1027                                                       IF EVALUATE(ADCSID,COPYSTR(Parameter,STRPOS(Parameter,'=')...
1028                                                         ADCSNASStartup.SetNASID(ADCSID)
1029                                                       ELSE
```

It is possible to apply the guidelines to a selection of code. All comments of the objects. Or a selection of objects.

18.11. Manual Apply Guidelines

There are also coding guidelines checks that cannot be auto applied. E.g. the “Check 'FIND(`-`)'” check can be replaced with more possibilities so this can only be done manually.

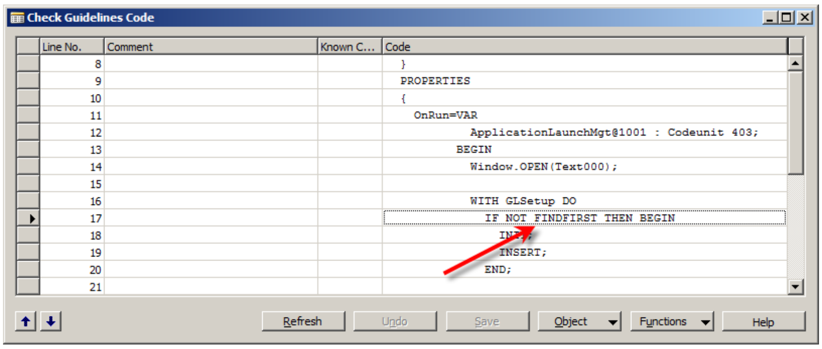
The fastest way to do this is directly in the code form.



Object Type	Object No.	Object Name	No. of Comments	No. of New Comments	No. of Known C...
Report	98	Date Compress General Ledger	12	12	
Report	99	Delete Empty G/L Registers	8	8	
Codeunit	1	ApplicationManagement	78	78	
Codeunit	2	Company-Initialize	22	22	
Codeunit	3	G/L			
Codeunit	6	Fisc			
Codeunit	7	GLB			
Codeunit	8	Acc			

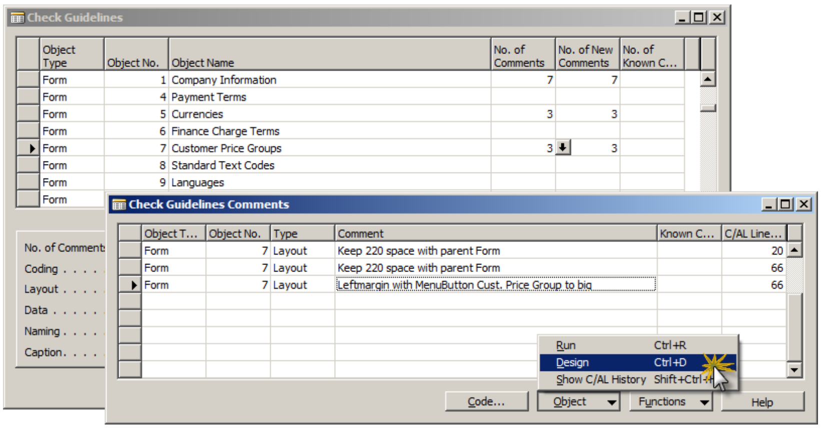
Line No.	Comment	Known C...	Code
8			}
9			PROPERTIES
10			{
11			OnRun=VAR
12			ApplicationLaunchMgt@1001 : Codeunit 403;
13			BEGIN
14			Window.OPEN(Text000);
15			
16			WITH GLSetup DO
17	Replace FIND('-') with 'FINDFIRST', 'FIN...		IF NOT FINDfirst THEN BEGIN
18			INIT;
19			INSERT;
20			END;
21			

Here it is possible to edit a line of code and press the save button. The objects will be saved and the guidelines will be rechecked and the comment will be deleted.



Line No.	Comment	Known C...	Code
8			}
9			PROPERTIES
10			{
11			OnRun=VAR
12			ApplicationLaunchMgt@1001 : Codeunit 403;
13			BEGIN
14			Window.OPEN(Text000);
15			
16			WITH GLSetup DO
17			IF NOT FINDFIRST THEN BEGIN
18			INIT;
19			INSERT;
20			END;
21			

Layout checks can be best solved in the Object Designer. To open the object in design mode you can press Ctrl+D.



Object Type	Object No.	Object Name	No. of Comments	No. of New Comments	No. of Known C...
Form	1	Company Information	7	7	
Form	4	Payment Terms			
Form	5	Currencies	3	3	
Form	6	Finance Charge Terms			
Form	7	Customer Price Groups	3	3	
Form	8	Standard Text Codes			
Form	9	Languages			
Form					

No. of Comments
Coding . . . .
Layout . . . .
Data . . . . .
Naming . . . .
Caption. . . .

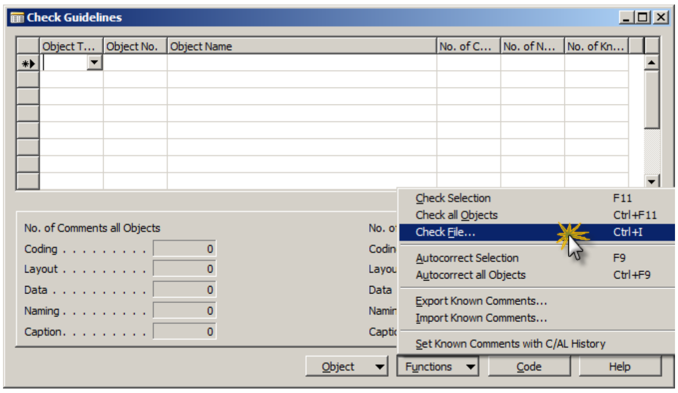
Object T...	Object No.	Type	Comment	Known C...	C/AL Line...
Form	7	Layout	Keep 220 space with parent Form		20
Form	7	Layout	Keep 220 space with parent Form		66
Form	7	Layout	Leftmargin with MenuButton Cust. Price Group to big		66

The Object Designer will open the selected object in design modus.

Also Data and Naming checks have to be done directly in the object itself or by modifying the results in the Check Guidelines Code form.

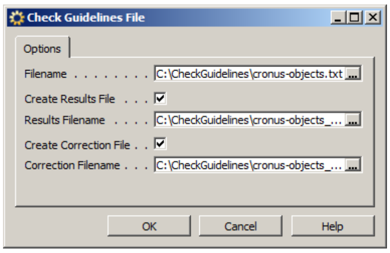
18.12. Perform Guidelines on Text File

The function Check File gives the possibility to check objects in text format.

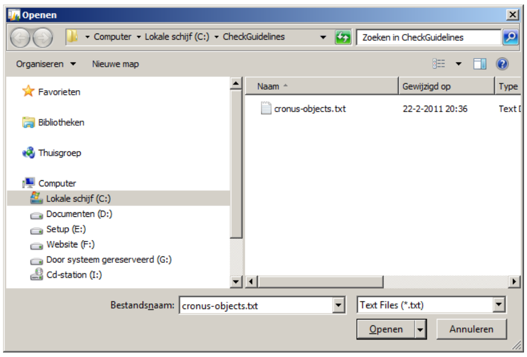


No. of Comments all Objects
Coding 0
Layout 0
Data 0
Naming 0
Caption. 0

Select the file you want to check.



Filename . . . . C:\CheckGuidelines\cronus-objects.txt
Create Results File . . . ☑
Results Filename . . . . C:\CheckGuidelines\cronus-objects_...
Create Correction File . . ☑
Correction Filename . . . C:\CheckGuidelines\cronus-objects_...



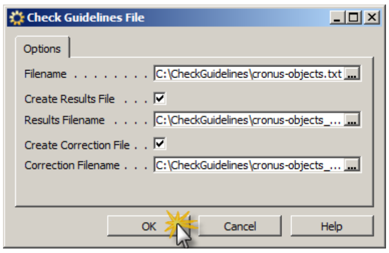
Naam	Gewijzigd op	Type
cronus-objects.txt	22-2-2011 20:36	Text

Bestandsnaam: cronus-objects.txt
Text Files (*.txt)

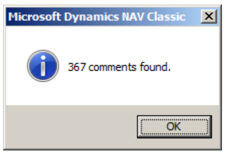
- **Create Results File**
  If you set Create Results File the comments will be published to a text file.

- **Create Correction File**
  If you set Create Correction File the Guidelines Tool will automatically correct (there where autocorrect is possible) the text file.

Press OK

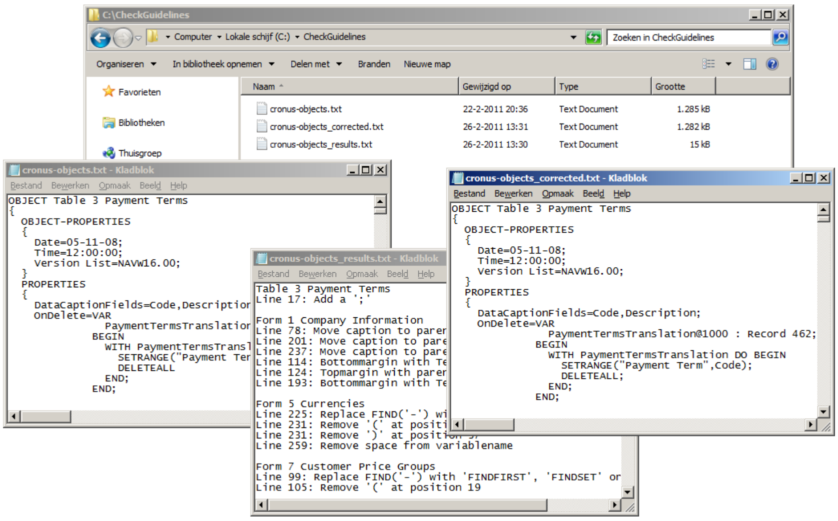


Filename . . . . C:\CheckGuidelines\cronus-objects.txt
Create Results File . . . ☑
Results Filename . . . . C:\CheckGuidelines\cronus-objects_...
Create Correction File . . ☑
Correction Filename . . . C:\CheckGuidelines\cronus-objects_...



367 comments found.

The results and corrected file are created.



Naam	Gewijzigd op	Type	Grootte
cronus-objects.txt	22-2-2011 20:36	Text Document	1.285 kB
cronus-objects_corrected.txt	26-2-2011 13:31	Text Document	1.282 kB
cronus-objects_results.txt	26-2-2011 13:30	Text Document	15 kB

```
OBJECT Table 3 Payment Terms
{
  OBJECT-PROPERTIES
  {
    Date=05-11-08;
    Time=12:00:00;
    Version List=NAVW16.00;
  }
  PROPERTIES
  {
    DataCaptionFields=Code,Descriptior
    OnDelete=VAR
               PaymentTermsTranslatior
             BEGIN
               WITH PaymentTermsTransl
                 SETRANGE("Payment Ter
                 DELETEALL
               END;
             END;
```

```
Table 3 Payment Terms
Line 17: Add a ';'

Form 1 Company Information
Line 78: Move caption to parer
Line 201: Move caption to pare
Line 237: Move caption to pare
Line 114: Bottommargin with Te
Line 124: Topmargin with parer
Line 193: Bottommargin with Te

Form 5 Currencies
Line 225: Replace FIND('-') wi
Line 231: Remove '(' at positi
Line 231: Remove ')' at positi
Line 259: Remove space from variablename

Form 7 Customer Price Groups
Line 99: Replace FIND('-') with 'FINDFIRST', 'FINDSET' or
Line 105: Remove '(' at position 19
```

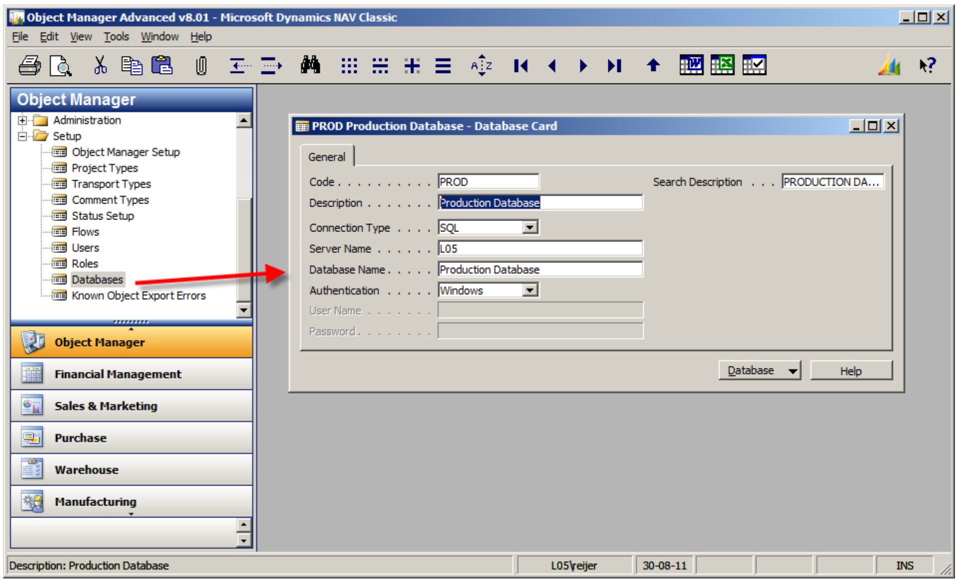
```
OBJECT Table 3 Payment Terms
{
  OBJECT-PROPERTIES
  {
    Date=05-11-08;
    Time=12:00:00;
    Version List=NAVW16.00;
  }
  PROPERTIES
  {
    DataCaptionFields=Code,Description;
    OnDelete=VAR
               PaymentTermsTranslation@1000 : Record 462;
             BEGIN
               WITH PaymentTermsTranslation DO BEGIN
                 SETRANGE("Payment Term",Code);
                 DELETEALL;
               END;
             END;
```

19. Compare Databases

With the Compare Database functionality the user is able to compare objects of two separate databases. You are able to compare current database with another or compare 2 databases other than the current you are in.

19.1. Setup

Open Setup > Databases:



Code . . . . PROD

Description . . . . Production Database

Connection Type . . . . SQL

Server Name L05

Database Name . . . . . Production Database

Authentication . . . . . Windows

User Name . . . .

Password . . . .

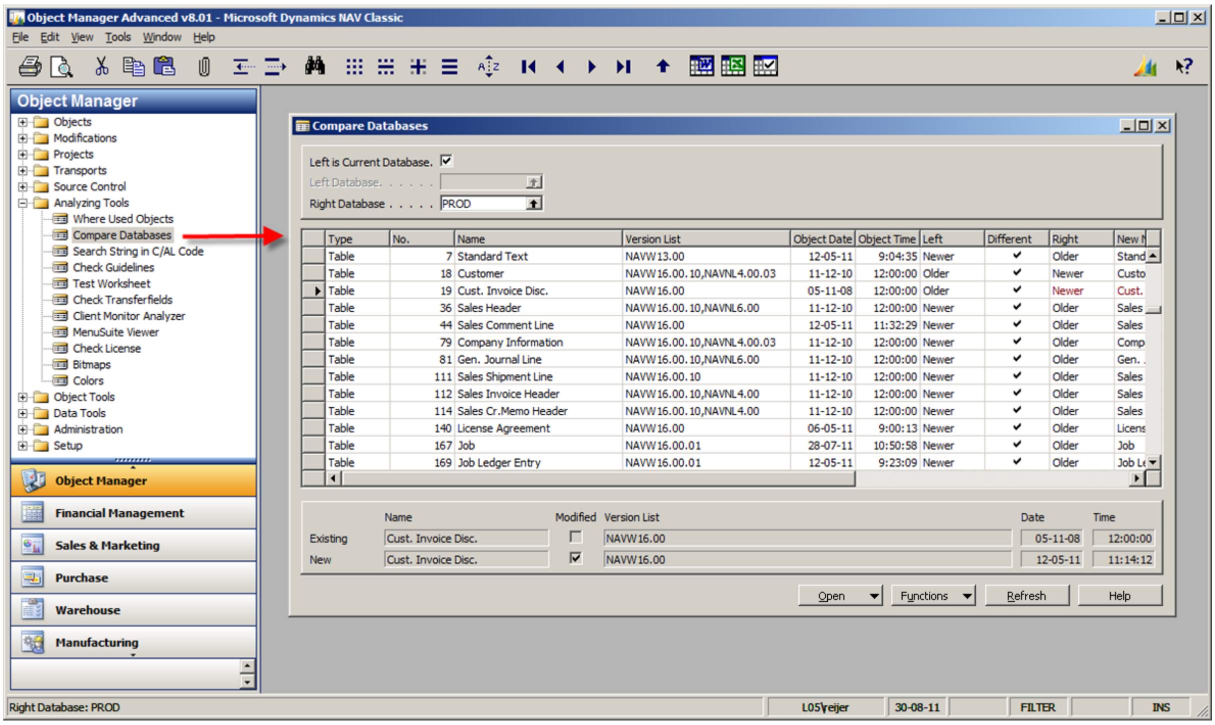
Search Description . . . PRODUCTION DA...

Description: Production Database

In the Database Card you set the properties for the databases to you want to connect to compare.

19.2. Comparing Databases

Open Analyzing Tools > Compare Databases.



Left is Current Database. ☑
Left Database. . . . . .
Right Database . . . . . PROD

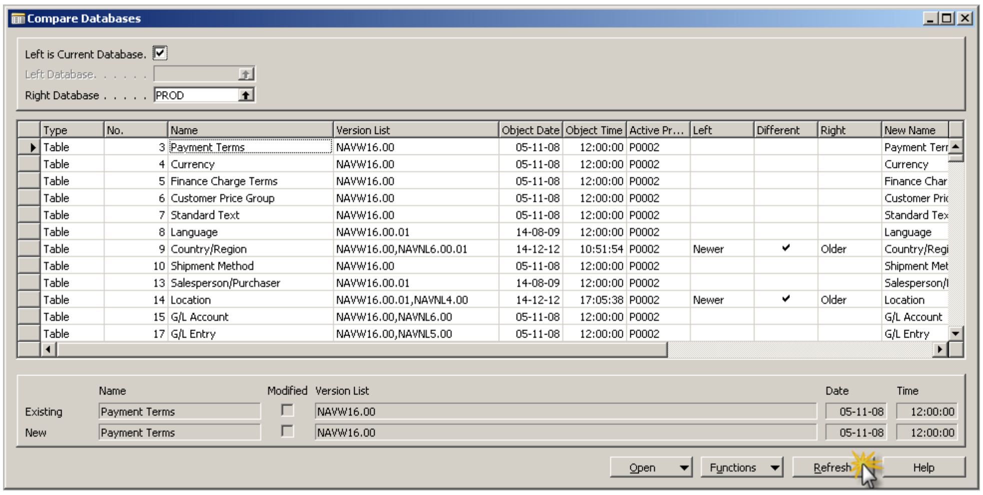
Type	No.	Name	Version List	Object Date	Object Time	Left	Different	Right	New N
Table	7	Standard Text	NAVW13.00	12-05-11	9:04:35	Newer	✓	Older	Stand
Table	18	Customer	NAVW16.00.10,NAVNL4.00.03	11-12-10	12:00:00	Older	✓	Newer	Custo
Table	19	Cust. Invoice Disc.	NAVW16.00	05-11-08	12:00:00	Older	✓	Newer	Cust.
Table	36	Sales Header	NAVW16.00.10,NAVNL6.00	11-12-10	12:00:00	Newer	✓	Older	Sales
Table	44	Sales Comment Line	NAVW16.00	12-05-11	11:32:29	Newer	✓	Older	Sales
Table	79	Company Information	NAVW16.00.10,NAVNL4.00.03	11-12-10	12:00:00	Newer	✓	Older	Comp
Table	81	Gen. Journal Line	NAVW16.00.10,NAVNL6.00	11-12-10	12:00:00	Newer	✓	Older	Gen. .
Table	111	Sales Shipment Line	NAVW16.00.10	11-12-10	12:00:00	Newer	✓	Older	Sales
Table	112	Sales Invoice Header	NAVW16.00.10,NAVNL4.00	11-12-10	12:00:00	Newer	✓	Older	Sales
Table	114	Sales Cr.Memo Header	NAVW16.00.10,NAVNL4.00	11-12-10	12:00:00	Newer	✓	Older	Sales
Table	140	License Agreement	NAVW16.00	06-05-11	9:00:13	Newer	✓	Older	Licens
Table	167	Job	NAVW16.00.01	28-07-11	10:50:58	Newer	✓	Older	Job
Table	169	Job Ledger Entry	NAVW16.00.01	12-05-11	9:23:09	Newer	✓	Older	Job Le

	Name	Modified	Version List	Date	Time
Existing	Cust. Invoice Disc.	☐	NAVW16.00	05-11-08	12:00:00
New	Cust. Invoice Disc.	☑	NAVW16.00	12-05-11	11:14:12

If you want to compare your current database set “Left is Current Database”. If you want to compare another database you select a database in the field “Left Database”.

Select a database in the “Right Database” you want to compare.

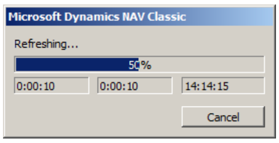
Press Refresh.



Left is Current Database. ☑
Left Database. . . . . .
Right Database . . . . . PROD

Type	No.	Name	Version List	Object Date	Object Time	Active Pr...	Left	Different	Right	New Name
Table	3	Payment Terms	NAVW16.00	05-11-08	12:00:00	P0002				Payment Terr
Table	4	Currency	NAVW16.00	05-11-08	12:00:00	P0002				Currency
Table	5	Finance Charge Terms	NAVW16.00	05-11-08	12:00:00	P0002				Finance Char
Table	6	Customer Price Group	NAVW16.00	05-11-08	12:00:00	P0002				Customer Pric
Table	7	Standard Text	NAVW16.00	05-11-08	12:00:00	P0002				Standard Tex
Table	8	Language	NAVW16.00.01	14-08-09	12:00:00	P0002				Language
Table	9	Country/Region	NAVW16.00,NAVNL6.00.01	14-12-12	10:51:54	P0002	Newer	✓	Older	Country/Regi
Table	10	Shipment Method	NAVW16.00	05-11-08	12:00:00	P0002				Shipment Met
Table	13	Salesperson/Purchaser	NAVW16.00.01	14-08-09	12:00:00	P0002				Salesperson/I
Table	14	Location	NAVW16.00.01,NAVNL4.00	14-12-12	17:05:38	P0002	Newer	✓	Older	Location
Table	15	G/L Account	NAVW16.00,NAVNL6.00	05-11-08	12:00:00	P0002				G/L Account
Table	17	G/L Entry	NAVW16.00,NAVNL5.00	05-11-08	12:00:00	P0002				G/L Entry

	Name	Modified	Version List	Date	Time
Existing	Payment Terms	☐	NAVW16.00	05-11-08	12:00:00
New	Payment Terms	☐	NAVW16.00	05-11-08	12:00:00



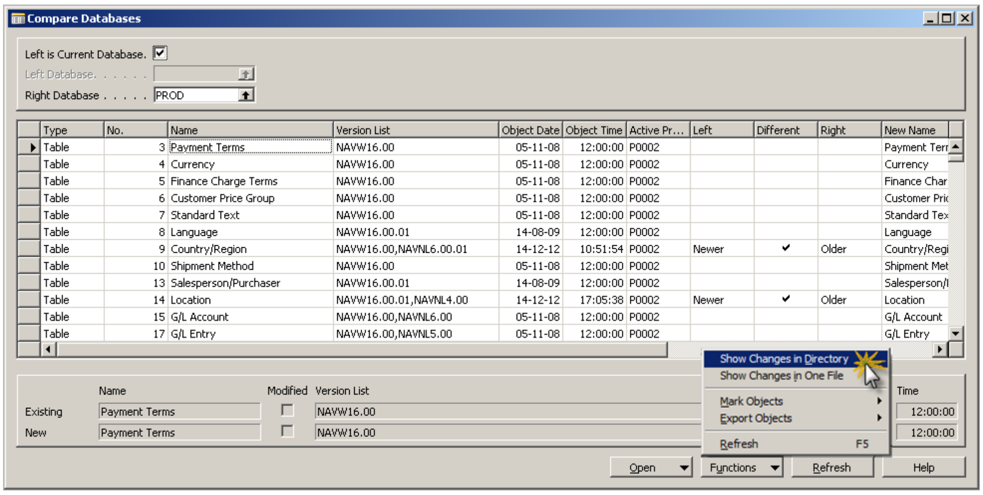
Refreshing...

50%

0:00:10 0:00:10 14:14:15

You can set a filter on “Different” to show only the objects which are not equal or do not exist in one of the databases.

You can analyze the differences with your compare tool with the menu options “Show Changes in Directory” or “Show Changes in One File”



Left is Current Database. ☑
Left Database. . . . . .
Right Database . . . . . PROD

Type	No.	Name	Version List	Object Date	Object Time	Active Pr...	Left	Different	Right	New Name
Table	3	Payment Terms	NAVW16.00	05-11-08	12:00:00	P0002				Payment Terr
Table	4	Currency	NAVW16.00	05-11-08	12:00:00	P0002				Currency
Table	5	Finance Charge Terms	NAVW16.00	05-11-08	12:00:00	P0002				Finance Char
Table	6	Customer Price Group	NAVW16.00	05-11-08	12:00:00	P0002				Customer Pri
Table	7	Standard Text	NAVW16.00	05-11-08	12:00:00	P0002				Standard Tex
Table	8	Language	NAVW16.00.01	14-08-09	12:00:00	P0002				Language
Table	9	Country/Region	NAVW16.00,NAVNL6.00.01	14-12-12	10:51:54	P0002	Newer	✓	Older	Country/Regi
Table	10	Shipment Method	NAVW16.00	05-11-08	12:00:00	P0002				Shipment Met
Table	13	Salesperson/Purchaser	NAVW16.00.01	14-08-09	12:00:00	P0002				Salesperson/
Table	14	Location	NAVW16.00.01,NAVNL4.00	14-12-12	17:05:38	P0002	Newer	✓	Older	Location
Table	15	G/L Account	NAVW16.00,NAVNL6.00	05-11-08	12:00:00	P0002				G/L Account
Table	17	G/L Entry	NAVW16.00,NAVNL5.00	05-11-08	12:00:00	P0002				G/L Entry

	Name	Modified	Version List	Time
Existing	Payment Terms	☐	NAVW16.00	12:00:00
New	Payment Terms	☐	NAVW16.00	12:00:00

For any of the objects in the left database the “Active Projects” column lists all active projects the object is part of.

Note: If lines are red the C/AL History has to be updated in the database.

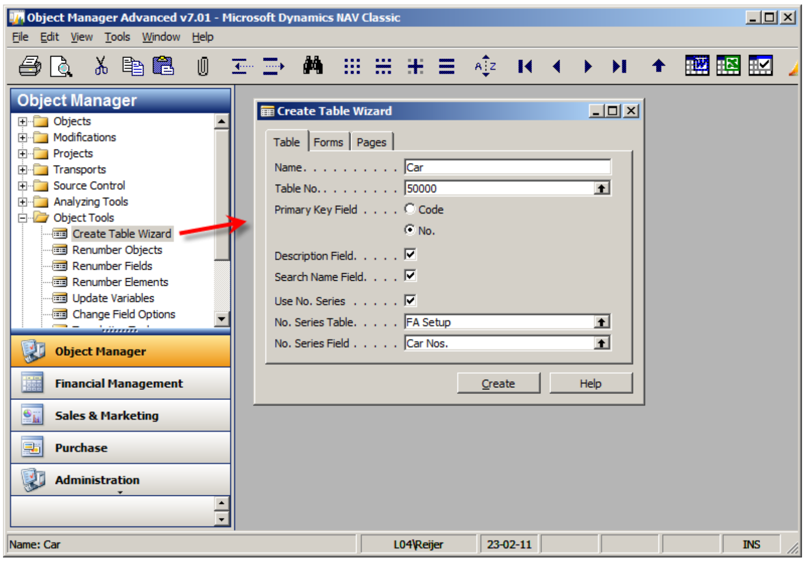
19.3. Comparing Files

By opening fob, fib, object .txt and obp files as left and right you can compare the content with each other.

20. Create Table Wizard

In the Object Designer there are wizards to create forms, reports and pages but none for tables. With this tool you can create a table with a wizard and even the corresponding forms and pages will be created.

20.1. Create a new Table with the Wizard

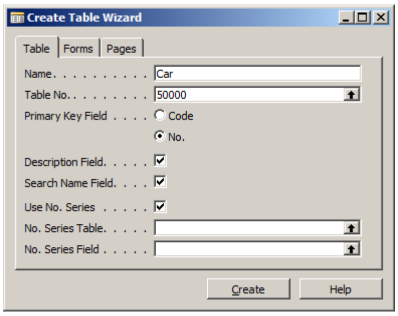


Name. . . . . Car
Table No. 50000
Primary Key Field . . . . ○ Code ● No.
Description Field. . . . . ☑
Search Name Field. . . . ☑
Use No. Series . . . . . ☑
No. Series Table. . . . . FA Setup
No. Series Field . . . . . Car Nos.

Name: Car | L04\Reijer | 23-02-11 | INS

Just like the forms and page wizard you fill in the name and table number of the new table.

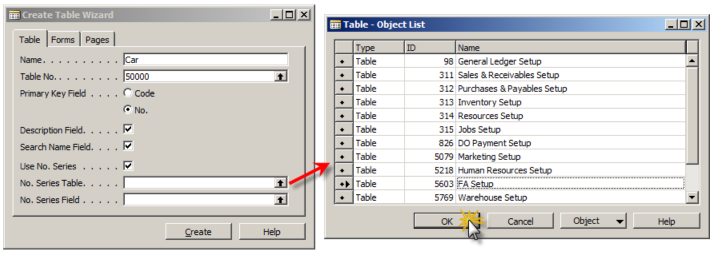
You can choose a Code or a No. field as primary key.



Name. . . . . Car
Table No. 50000
Primary Key Field . . . . ○ Code ● No.
Description Field. . . . . ☑
Search Name Field. . . . ☑
Use No. Series . . . . . ☑
No. Series Table. . . . .
No. Series Field . . . . .

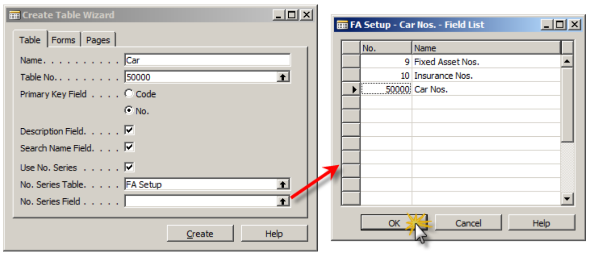
Optional is creating additional Description and “Search Name” fields.

If you use a No. field you can link the No. to a “No. Series” field in an existing table.



Name. . . . . Car
Table No. 50000
Primary Key Field . . . . ○ Code
● No.
Description Field. . . . . ☑
Search Name Field. . . . ☑
Use No. Series . . . . . ☑
No. Series Table. . . . .
No. Series Field . . . . .

Type	ID	Name
Table	98	General Ledger Setup
Table	311	Sales & Receivables Setup
Table	312	Purchases & Payables Setup
Table	313	Inventory Setup
Table	314	Resources Setup
Table	315	Jobs Setup
Table	826	DO Payment Setup
Table	5079	Marketing Setup
Table	5218	Human Resources Setup
Table	5603	FA Setup
Table	5769	Warehouse Setup



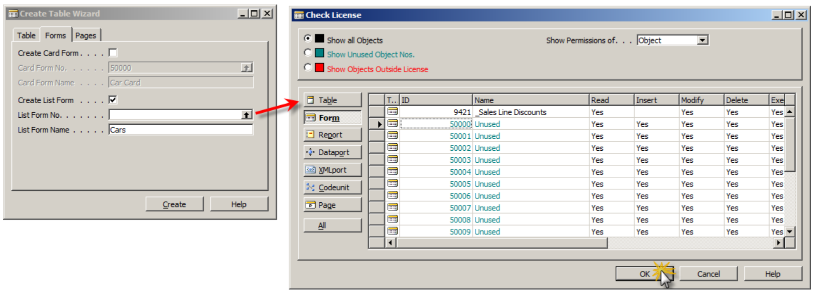
Name. . . . . Car
Table No. 50000
Primary Key Field . . . . ○ Code
● No.
Description Field. . . . . ☑
Search Name Field. . . . ☑
Use No. Series . . . . . ☑
No. Series Table. . . . . FA Setup
No. Series Field . . . . .

No.	Name
9	Fixed Asset Nos.
10	Insurance Nos.
50000	Car Nos.

20.1.1. Create Additional Forms

On the forms tab you have two options.

- Create Card Form
- Create List Form

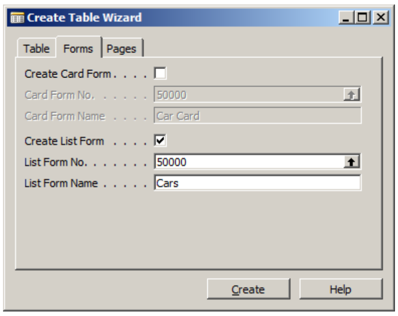


Create Card Form . . . . ☐
Card Form No. 50000
Card Form Name . . . . Car Card
Create List Form . . . . ☑
List Form No. . . . . . .
List Form Name . . . . . Cars

● Show all Objects
○ Show Unused Object Nos.
○ Show Objects Outside License
Show Permissions of. . . Object

T..	ID	Name	Read	Insert	Modify	Delete	Exe
	9421	_Sales Line Discounts	Yes		Yes	Yes	Yes
	50000	Unused	Yes	Yes	Yes	Yes	Yes
	50001	Unused	Yes	Yes	Yes	Yes	Yes
	50002	Unused	Yes	Yes	Yes	Yes	Yes
	50003	Unused	Yes	Yes	Yes	Yes	Yes
	50004	Unused	Yes	Yes	Yes	Yes	Yes
	50005	Unused	Yes	Yes	Yes	Yes	Yes
	50006	Unused	Yes	Yes	Yes	Yes	Yes
	50007	Unused	Yes	Yes	Yes	Yes	Yes
	50008	Unused	Yes	Yes	Yes	Yes	Yes
	50009	Unused	Yes	Yes	Yes	Yes	Yes

When you use the Lookup for the card or list form number field, the Check License screen shows you which object numbers are still available.

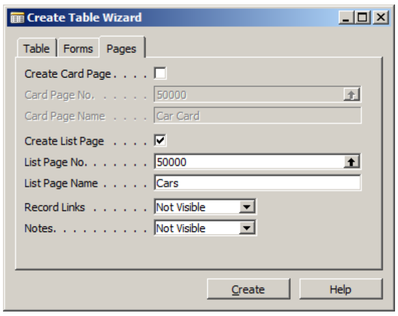


Create Card Form . . . . ☐
Card Form No. 50000
Card Form Name . . . . Car Card
Create List Form . . . . ☑
List Form No. 50000
List Form Name . . . . . Cars

20.1.2. Create Additional Pages

On the pages tab you have the same two options as on the forms tab.

- Create Card Page
- Create List Page

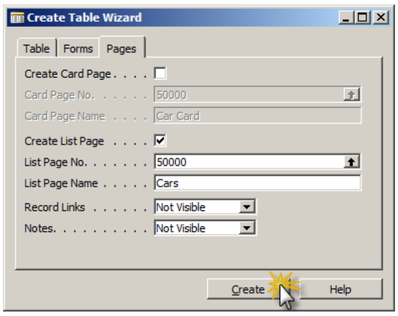


Create Card Page . . . . ☐
Card Page No. 50000
Card Page Name . . . . Car Card
Create List Page . . . . ☑
List Page No. 50000
List Page Name . . . . . Cars
Record Links . . . . . . Not Visible
Notes. . . . . Not Visible

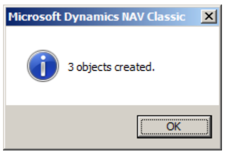
Extra options for pages are:

- Record Links
  - No
  - Visible
  - Not Visible
- Notes
  - No
  - Visible
  - Not Visisible

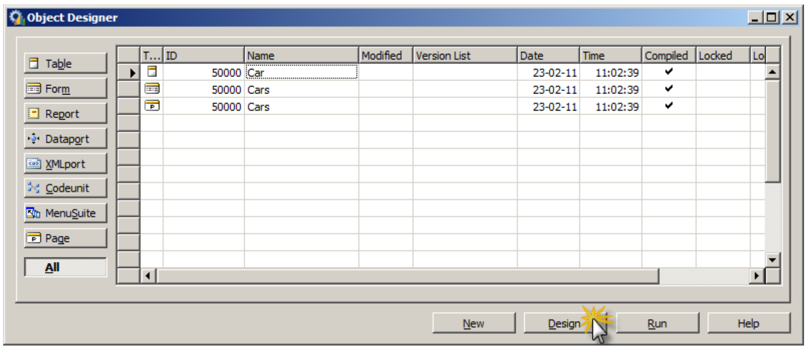
When you press Create all the required objects are created.



Create Card Page . . . . ☐
Card Page No. 50000
Card Page Name . . . . Car Card
Create List Page . . . . ☑
List Page No. 50000
List Page Name . . . . . Cars
Record Links . . . . . . Not Visible
Notes. . . . . Not Visible



3 objects created.



T...	ID	Name	Modified	Version List	Date	Time	Compiled	Locked	Lo
	50000	Car			23-02-11	11:02:39	✓		
	50000	Cars			23-02-11	11:02:39	✓		
	50000	Cars			23-02-11	11:02:39	✓		

With all additional C/AL code:

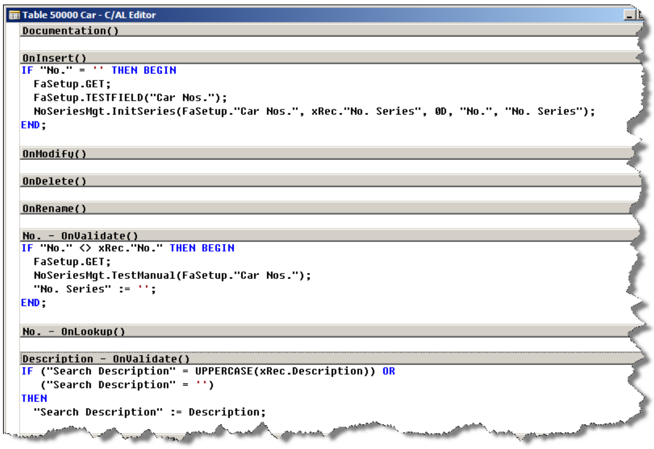


Table 50000 Car - C/AL Editor

```
Documentation()

OnInsert()
IF "No." = '' THEN BEGIN
  FaSetup.GET;
  FaSetup.TESTFIELD("Car Nos.");
  NoSeriesMgt.InitSeries(FaSetup."Car Nos.", xRec."No. Series", 0D, "No.", "No. Series");
END;

OnModify()

OnDelete()

OnRename()

No. - OnValidate()
IF "No." <> xRec."No." THEN BEGIN
  FaSetup.GET;
  NoSeriesMgt.TestManual(FaSetup."Car Nos.");
  "No. Series" := '';
END;

No. - OnLookup()

Description - OnValidate()
IF ("Search Description" = UPPERCASE(xRec.Description)) OR
   ("Search Description" = '')
THEN
  "Search Description" := Description;
```

20.1.3. Translation Rules and Table Wizard

If there are rules in the Translation Tool. The captions in the objects are automatically created according to these rules. For more information see section 26.1 - Rules.

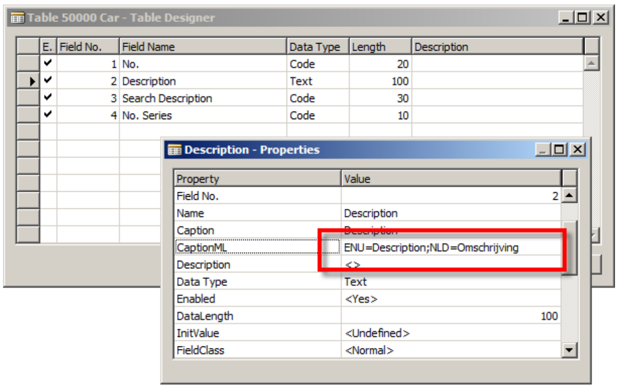


Table 50000 Car - Table Designer

E.	Field No.	Field Name	Data Type	Length	Description
✓	1	No.	Code	20	
✓	2	Description	Text	100	
✓	3	Search Description	Code	30	
✓	4	No. Series	Code	10	

Description - Properties

Property	Value
Field No.	2
Name	Description
Caption	Description
CaptionML	ENU=Description;NLD=Omschrijving
Description	<>
Data Type	Text
Enabled	<Yes>
DataLength	100
InitValue	<Undefined>
FieldClass	<Normal>

21. Renumber Objects

With this tool you can renumber objects. When another object has a connection with the renumbered object the reference(s) in these objects will also be changed. It even is possible to renumber tables with data in it.

There are three possible modes.

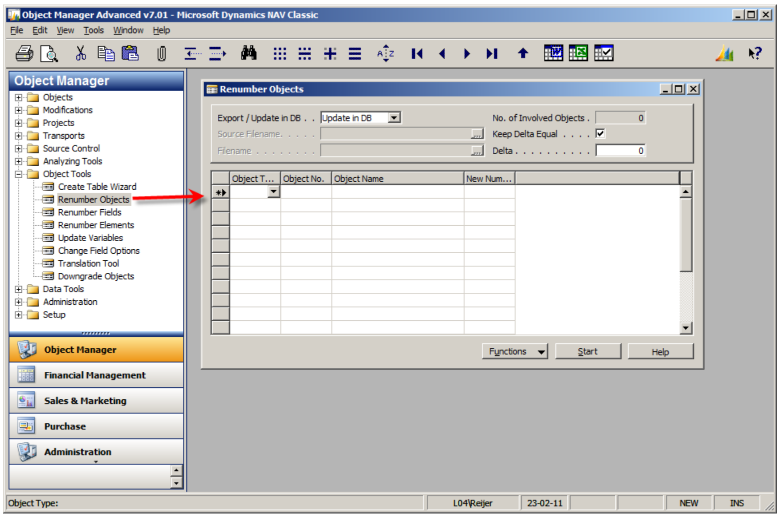
- Update in DB: Objects are renumbered in the database.
- Export: The objects are renumbered and exported in text format.
- Update text file: Objects are renumbered in a text file.

Update is recommended because all data in the Object Manager will also be renumbered. If you choose for the export mode, the data will stay the same.

NOTE: To use the first two modes you first have to update the “Where Used Objects”. For more information see section 15.1 - Setup**.**

21.1. Update in DB Mode

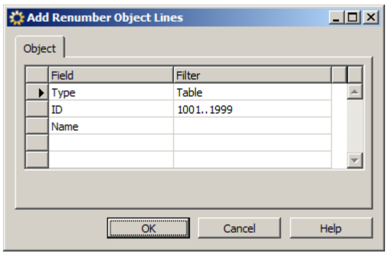
Press “Add Objects”.



Export / Update in DB . . Update in DB
Source Filename. . . . .
Filename . . . .
No. of Involved Objects . 0
Keep Delta Equal . . . . ☑
Delta 0

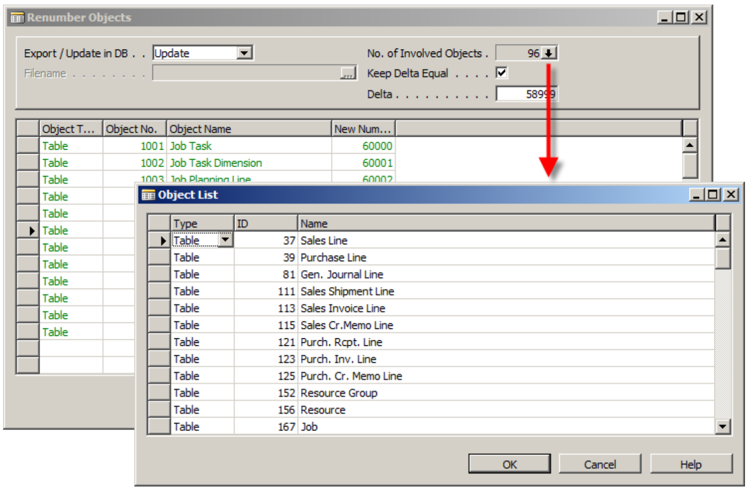
Object T...	Object No.	Object Name	New Num...

Object Type: L04\Reijer 23-02-11 NEW INS



Field	Filter
Type	Table
ID	1001..1999
Name	

In field “No. of Involved Objects” you can see which objects will be updated when you renumber your selection of objects.



Export / Update in DB . . Update

Filename . . . .

No. of Involved Objects . 96

Keep Delta Equal . . . . ☑

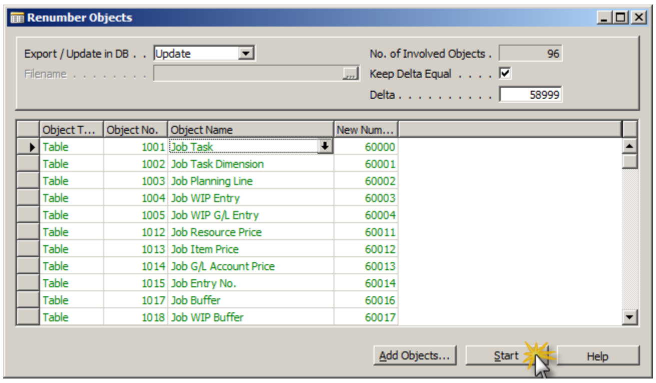
Delta 58999

Object T...	Object No.	Object Name	New Num...
Table	1001	Job Task	60000
Table	1002	Job Task Dimension	60001

Type	ID	Name
Table	37	Sales Line
Table	39	Purchase Line
Table	81	Gen. Journal Line
Table	111	Sales Shipment Line
Table	113	Sales Invoice Line
Table	115	Sales Cr.Memo Line
Table	121	Purch. Rcpt. Line
Table	123	Purch. Inv. Line
Table	125	Purch. Cr. Memo Line
Table	152	Resource Group
Table	156	Resource
Table	167	Job

With the option “Keep Delta Equal” you only have to change one “New Number”. The Object Manager will apply the same delta to all other lines. If you have selected a couple of lines the delta will only apply to the selected lines. Without the option “Keep Delta Equal” you can modify the lines separately.

Press Start.



Export / Update in DB . . Update

Filename . . . .

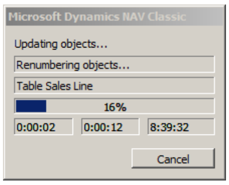
No. of Involved Objects . 96

Keep Delta Equal . . . . ☑

Delta 58999

Object T...	Object No.	Object Name	New Num...
Table	1001	Job Task	60000
Table	1002	Job Task Dimension	60001
Table	1003	Job Planning Line	60002
Table	1004	Job WIP Entry	60003
Table	1005	Job WIP G/L Entry	60004
Table	1012	Job Resource Price	60011
Table	1013	Job Item Price	60012
Table	1014	Job G/L Account Price	60013
Table	1015	Job Entry No.	60014
Table	1017	Job Buffer	60016
Table	1018	Job WIP Buffer	60017

Objects will be renumbered.



Updating objects...

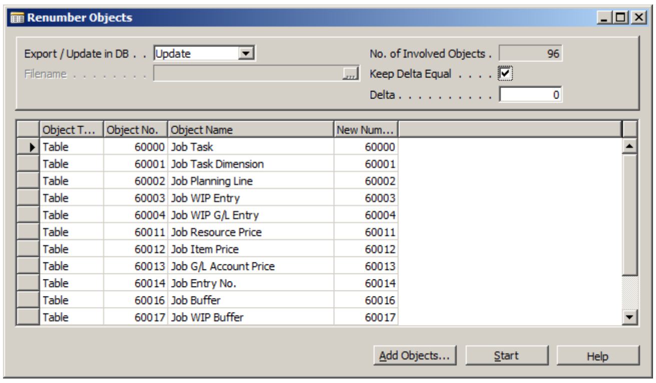
Renumbering objects...

Table Sales Line

16%

0:00:02 0:00:12 8:39:32

When the lines are colored black, the renumbering is finished.



Export / Update in DB . . Update

Filename . . . .

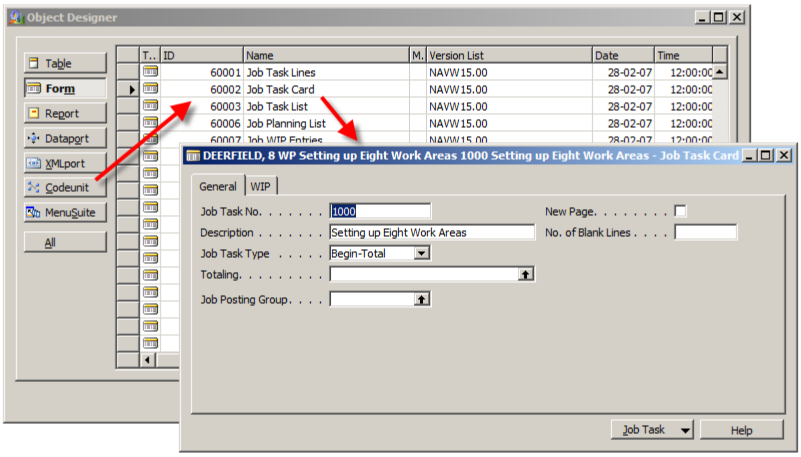
No. of Involved Objects . 96

Keep Delta Equal . . . . ☑

Delta 0

Object T...	Object No.	Object Name	New Num...
Table	60000	Job Task	60000
Table	60001	Job Task Dimension	60001
Table	60002	Job Planning Line	60002
Table	60003	Job WIP Entry	60003
Table	60004	Job WIP G/L Entry	60004
Table	60011	Job Resource Price	60011
Table	60012	Job Item Price	60012
Table	60013	Job G/L Account Price	60013
Table	60014	Job Entry No.	60014
Table	60016	Job Buffer	60016
Table	60017	Job WIP Buffer	60017

Also the references to these objects are updated.


Object Designer
Table
Form
Report
Dataport
XMLport
Codeunit
MenuSuite
All
DEERFIELD, 8 WP Setting up Eight Work Areas 1000 Setting up Eight Work Areas - Job Task Card
General
WIP
Job Task
Help

T..	ID	Name	M.	Version List	Date	Time
	60001	Job Task Lines		NAVW15.00	28-02-07	12:00:00
	60002	Job Task Card		NAVW15.00	28-02-07	12:00:00
	60003	Job Task List		NAVW15.00	28-02-07	12:00:00
	60006	Job Planning List		NAVW15.00	28-02-07	12:00:00
	60007	Job WIP Entries		NAVW15.00	28-02-07	12:00:00

Job Task No. 1000

Description . . . . Setting up Eight Work Areas

Job Task Type . . . . . Begin-Total

Totaling. . . . .

Job Posting Group. . . .

New Page. . . . . ☐

No. of Blank Lines . . . .

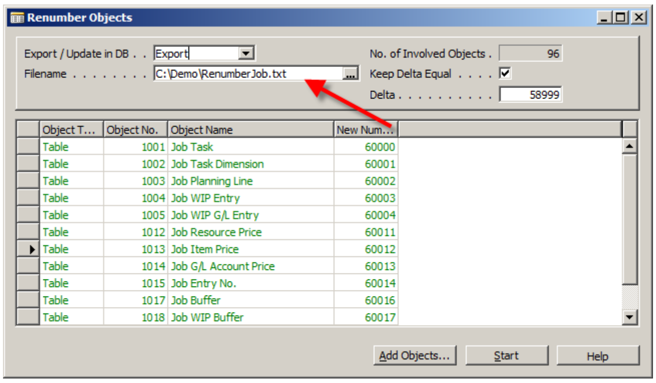
The lines in the renumber form can have the following colors:

- Black: New number is equal to original number
- Red: New number already exists
- Purple: New number is two times present
- Green: Object will be renumbered

21.2. Export Mode

This option works the same as the Updating option, but the objects in the original database are not renumbered. The renumbering is only done in the exported file. The exported file is a text file which you can import in the Object Designer.

Choose option Export.



Renumber Objects

Export / Update in DB . . Export

Filename . . . . C:\Demo\RenumberJob.txt

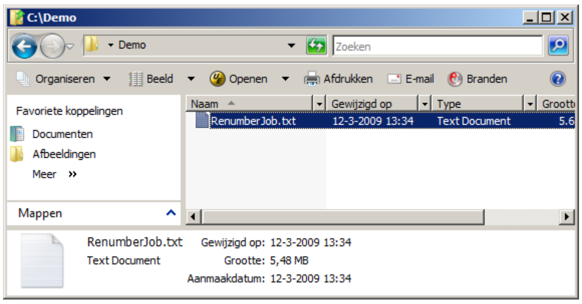
No. of Involved Objects . 96

Keep Delta Equal . . . . ☑

Delta 58999

Object T...	Object No.	Object Name	New Num...
Table	1001	Job Task	60000
Table	1002	Job Task Dimension	60001
Table	1003	Job Planning Line	60002
Table	1004	Job WIP Entry	60003
Table	1005	Job WIP G/L Entry	60004
Table	1012	Job Resource Price	60011
Table	1013	Job Item Price	60012
Table	1014	Job G/L Account Price	60013
Table	1015	Job Entry No.	60014
Table	1017	Job Buffer	60016
Table	1018	Job WIP Buffer	60017

Select the filename for the text file.



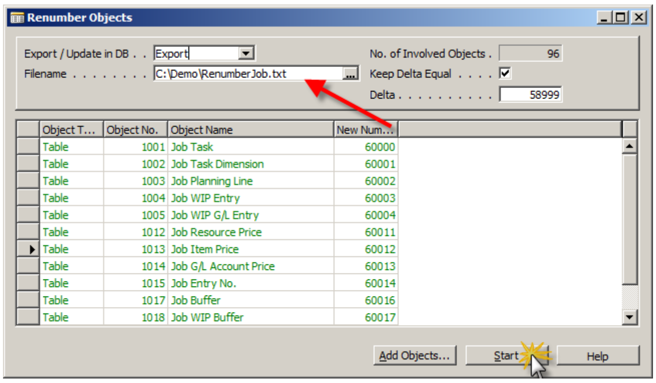
Naam	Gewijzigd op	Type	Groott
RenumberJob.txt	12-3-2009 13:34	Text Document	5.6

RenumberJob.txt Gewijzigd op: 12-3-2009 13:34

Text Document Grootte: 5,48 MB

Aanmaakdatum: 12-3-2009 13:34

Press Start to initiate the renumbering and the file will be created.



Renumber Objects

Export / Update in DB . . Export

Filename . . . . C:\Demo\RenumberJob.txt

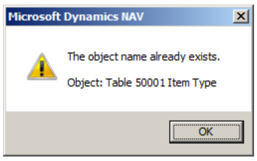
No. of Involved Objects . 96

Keep Delta Equal . . . . ☑

Delta 58999

Object T...	Object No.	Object Name	New Num...
Table	1001	Job Task	60000
Table	1002	Job Task Dimension	60001
Table	1003	Job Planning Line	60002
Table	1004	Job WIP Entry	60003
Table	1005	Job WIP G/L Entry	60004
Table	1012	Job Resource Price	60011
Table	1013	Job Item Price	60012
Table	1014	Job G/L Account Price	60013
Table	1015	Job Entry No.	60014
Table	1017	Job Buffer	60016
Table	1018	Job WIP Buffer	60017

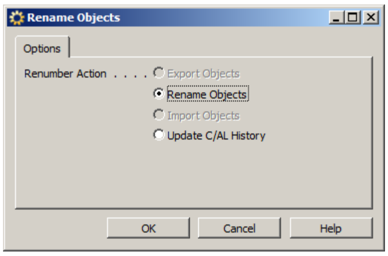
If you want to import the text file in the same database you must first run the report “OM - Rename Objects” else you will get errors like “The object name already exists”.



Microsoft Dynamics NAV

The object name already exists.

Object: Table 50001 Item Type



Rename Objects

Options

Renumber Action . . . . ○ Export Objects

● Rename Objects

○ Import Objects

○ Update C/AL History

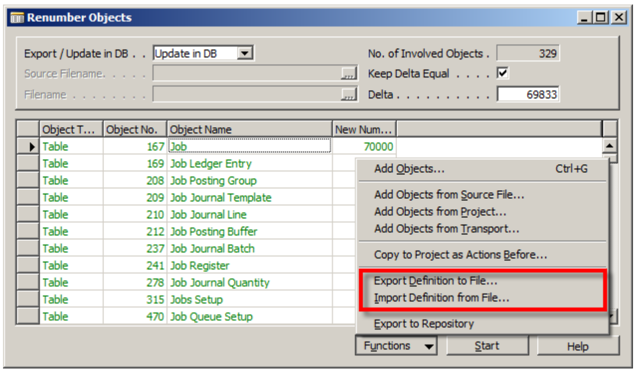
Because the renumbered objects have the same date and time the Object Manager sees this objects not as outdated and the “C/AL History will not be updated. Therefore it is recommended to run the report again to update the “C/AL History”.

21.3. Update Text File Mode

This mode works the same as the other two modes but the objects are renumbered in a text file. You enter a filename for the source file and a filename for the file that will be created with the renumbered objects.

21.4. Export Definition to File

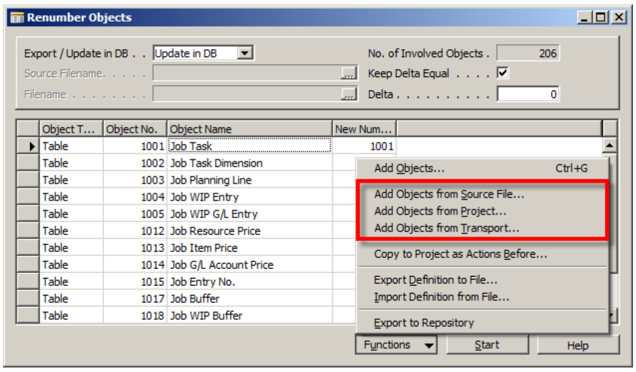
If you have to do the same renumbering in another database you can use the options Export and Import Definition to File.


Renumber Objects
Add Objects... Ctrl+G
Add Objects from Source File...
Add Objects from Project...
Add Objects from Transport...
Copy to Project as Actions Before...
Export Definition to File...
Import Definition from File...
Export to Repository
Functions
Start
Help

Export / Update in DB . . Update in DB
Source Filename. . . . .
Filename . . . .
No. of Involved Objects . 329
Keep Delta Equal . . . . ☑
Delta 69833

Object T...	Object No.	Object Name	New Num...
Table	167	Job	70000
Table	169	Job Ledger Entry	
Table	208	Job Posting Group	
Table	209	Job Journal Template	
Table	210	Job Journal Line	
Table	212	Job Posting Buffer	
Table	237	Job Journal Batch	
Table	241	Job Register	
Table	278	Job Journal Quantity	
Table	315	Jobs Setup	
Table	470	Job Queue Setup	

21.5. Add Objects from Source


Renumber Objects
Add Objects... Ctrl+G
Add Objects from Source File...
Add Objects from Project...
Add Objects from Transport...
Copy to Project as Actions Before...
Export Definition to File...
Import Definition from File...
Export to Repository
Functions
Start
Help

Export / Update in DB . . Update in DB
Source Filename. . . . .
Filename . . . .
No. of Involved Objects . 206
Keep Delta Equal . . . . ☑
Delta 0

Object T...	Object No.	Object Name	New Num...
Table	1001	Job Task	1001
Table	1002	Job Task Dimension	
Table	1003	Job Planning Line	
Table	1004	Job WIP Entry	
Table	1005	Job WIP G/L Entry	
Table	1012	Job Resource Price	
Table	1013	Job Item Price	
Table	1014	Job G/L Account Price	
Table	1015	Job Entry No.	
Table	1017	Job Buffer	
Table	1018	Job WIP Buffer	

22. Renumber Fields

With this tool you can renumber fields. When another object has a connection with the renumbered field the reference(s) in these objects will also be changed.

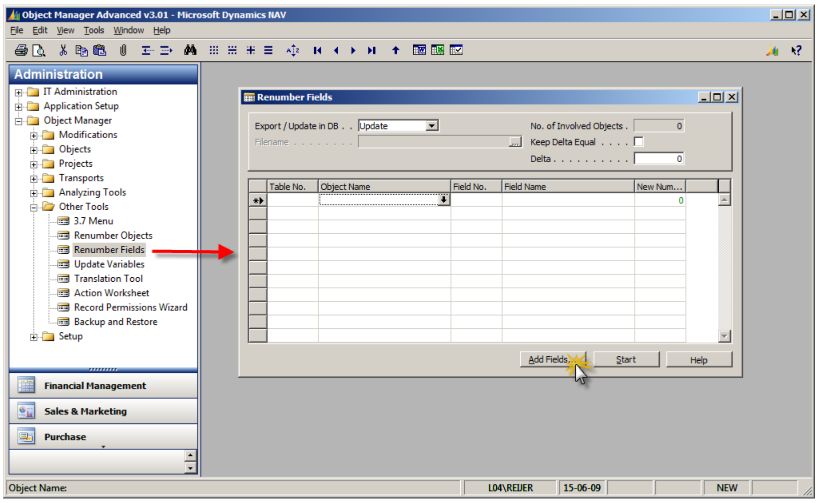
There are three possible modes.

- Update in DB: Fields are renumbered in the database.
- Export: Objects are exported in text format and fields are renumbered in the exported file.
- Update text file: Fields are exported in a text file.

NOTE: To use the first two modes you first have to update the "Where Used Objects". For more information see section 15.1 - Setup.

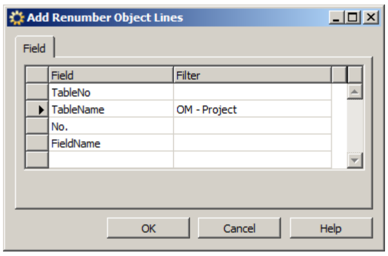
22.1. Update in DB Mode

Press "Add Fields".



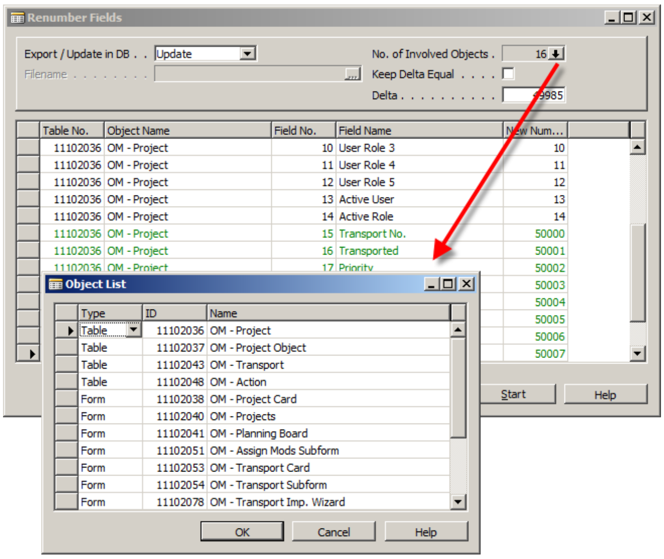
Export / Update in DB . . Update
Filename . . . .
No. of Involved Objects . 0
Keep Delta Equal . . . . ☐
Delta 0

Table No.	Object Name	Field No.	Field Name	New Num...
				0



Field	Filter
TableNo	
TableName	OM - Project
No.	
FieldName	

In field “No. of Involved Objects” you can see which objects will be updated when you renumber your selection of objects.



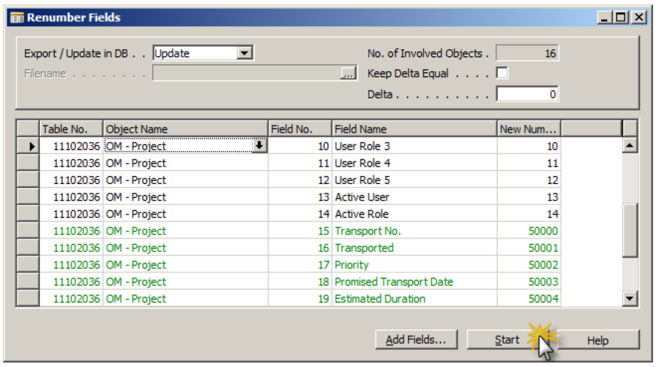
Export / Update in DB . . Update
Filename . . . .
No. of Involved Objects . 16
Keep Delta Equal . . . . ☐
Delta 49985

Table No.	Object Name	Field No.	Field Name	New Num...
11102036	OM - Project	10	User Role 3	10
11102036	OM - Project	11	User Role 4	11
11102036	OM - Project	12	User Role 5	12
11102036	OM - Project	13	Active User	13
11102036	OM - Project	14	Active Role	14
11102036	OM - Project	15	Transport No.	50000
11102036	OM - Project	16	Transported	50001
11102036	OM - Project	17	Priority	50002
				50003
				50004
				50005
				50006
				50007

Type	ID	Name
Table	11102036	OM - Project
Table	11102037	OM - Project Object
Table	11102043	OM - Transport
Table	11102048	OM - Action
Form	11102038	OM - Project Card
Form	11102040	OM - Projects
Form	11102041	OM - Planning Board
Form	11102051	OM - Assign Mods Subform
Form	11102053	OM - Transport Card
Form	11102054	OM - Transport Subform
Form	11102078	OM - Transport Imp. Wizard

With the option “Keep Delta Equal” you only have to change one “New Number”. The Object Manager will apply the same delta to all other lines. If you have selected a couple of lines the delta will only apply to the selected lines. Without the option “Keep Delta Equal” you can modify lines separately.

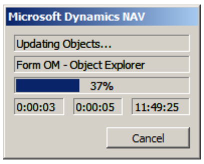
Press Start.



Export / Update in DB . . Update
Filename . . . .
No. of Involved Objects . 16
Keep Delta Equal . . . . ☐
Delta 0

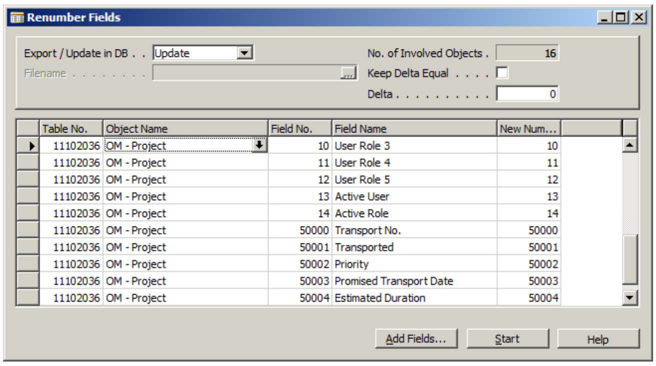
Table No.	Object Name	Field No.	Field Name	New Num...
11102036	OM - Project	10	User Role 3	10
11102036	OM - Project	11	User Role 4	11
11102036	OM - Project	12	User Role 5	12
11102036	OM - Project	13	Active User	13
11102036	OM - Project	14	Active Role	14
11102036	OM - Project	15	Transport No.	50000
11102036	OM - Project	16	Transported	50001
11102036	OM - Project	17	Priority	50002
11102036	OM - Project	18	Promised Transport Date	50003
11102036	OM - Project	19	Estimated Duration	50004

Fields will be renumbered.



Updating Objects...
Form OM - Object Explorer
37%
0:00:03 0:00:05 11:49:25

When all lines are black, the renumbering is finished.



Export / Update in DB . . Update
Filename . . . .
No. of Involved Objects . 16
Keep Delta Equal . . . . ☐
Delta 0

Table No.	Object Name	Field No.	Field Name	New Num...
11102036	OM - Project	10	User Role 3	10
11102036	OM - Project	11	User Role 4	11
11102036	OM - Project	12	User Role 5	12
11102036	OM - Project	13	Active User	13
11102036	OM - Project	14	Active Role	14
11102036	OM - Project	50000	Transport No.	50000
11102036	OM - Project	50001	Transported	50001
11102036	OM - Project	50002	Priority	50002
11102036	OM - Project	50003	Promised Transport Date	50003
11102036	OM - Project	50004	Estimated Duration	50004

The lines in the renumber form can have the following colors:

- Black: New number is equal to the original number
- Red: New number already exists
- Purple: New number is two times present
- Green: Field will be renumbered

22.2. Export Mode

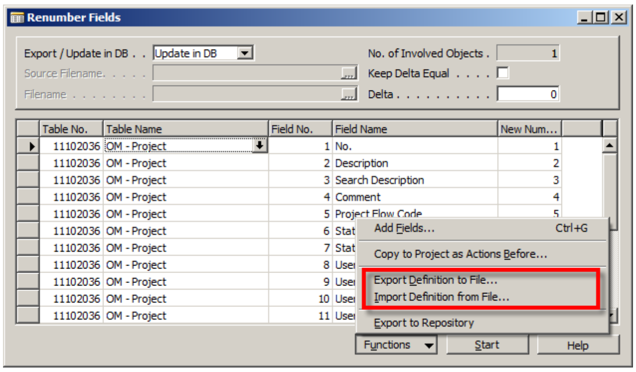
This option works the same as the Updating option, but the field in the original database are not renumbered. The renumbering is only done in the export file. The exported file is a text file which you can import in the Object Designer.

22.3. Update Text File Mode

This mode works the same as the other two modes but the fields are renumbered in a text file. You enter a filename for the source file and a filename for the file that will be created with the renumbered fields.

22.4. Export Definition to File

If you have to do the same renumbering in another database you can use the options Export and Import Definition to File.


Renumber Fields
Add Fields... Ctrl+G
Copy to Project as Actions Before...
Export Definition to File...
Import Definition from File...
Export to Repository
Functions
Start
Help

Export / Update in DB . . Update in DB
Source Filename. . . . .
Filename . . . .
No. of Involved Objects . 1
Keep Delta Equal . . . . ☐
Delta 0

Table No.	Table Name	Field No.	Field Name	New Num...
11102036	OM - Project	1	No.	1
11102036	OM - Project	2	Description	2
11102036	OM - Project	3	Search Description	3
11102036	OM - Project	4	Comment	4
11102036	OM - Project	5	Project Flow Code	5
11102036	OM - Project	6	Stat	
11102036	OM - Project	7	Stat	
11102036	OM - Project	8	Use	
11102036	OM - Project	9	Use	
11102036	OM - Project	10	Use	
11102036	OM - Project	11	Use	

23. Update Variables

With “Update Variables” you can check for unused variables in objects and clean up or sort them. You can perform this functionality on the database or export it to a text file.

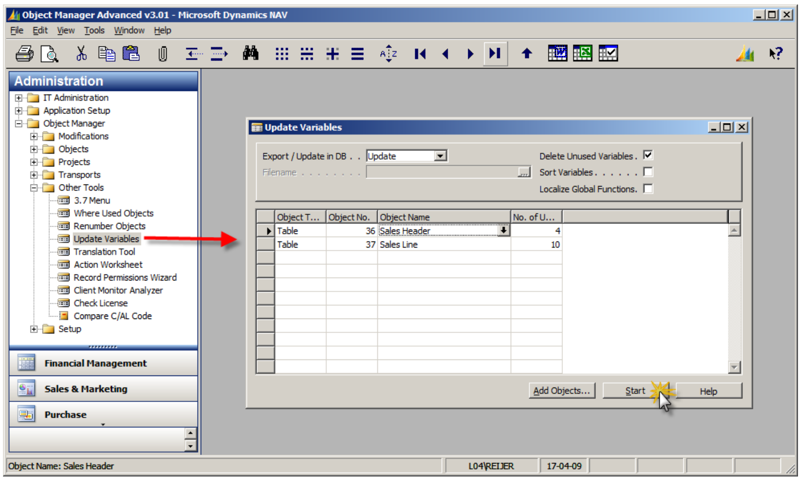
There are three possible modes.

- Update in DB: The objects are updated in the database.
- Export: The objects are exported in text format and the variables are updated in the exported file.
- Update text file: The variables are updates in text file.

NOTE: To use the first two modes you first have to update the “Where Used Objects”. For more information see section 15.1 - Setup.

23.1. Delete Unused Variables

Add the objects where you suspect unused variables. Press Start to delete the unused variables.



Export / Update in DB . . Update

Filename . . . .

Delete Unused Variables . ☑

Sort Variables . . . . . . ☐

Localize Global Functions. ☐

Object T...	Object No.	Object Name	No. of U...
Table	36	Sales Header	4
Table	37	Sales Line	10

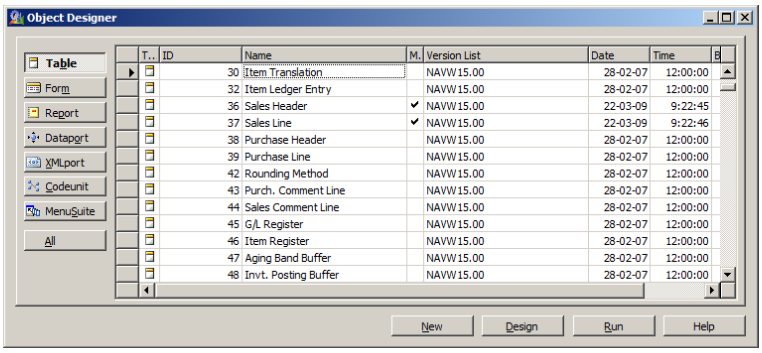
Object Name: Sales Header

L04\REIJER

17-04-09

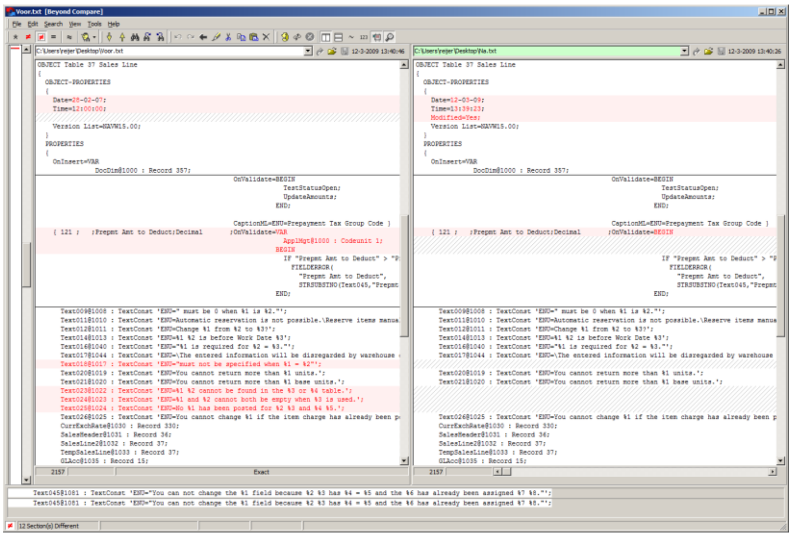
If something changes in an object the modify flag of the object is enabled so you can assign the modification to a project.

If an object is already ok the Object Manager skips it.



T..	ID	Name	M..	Version List	Date	Time	B
	30	Item Translation		NAVW15.00	28-02-07	12:00:00	
	32	Item Ledger Entry		NAVW15.00	28-02-07	12:00:00	
	36	Sales Header	✓	NAVW15.00	22-03-09	9:22:45	
	37	Sales Line	✓	NAVW15.00	22-03-09	9:22:46	
	38	Purchase Header		NAVW15.00	28-02-07	12:00:00	
	39	Purchase Line		NAVW15.00	28-02-07	12:00:00	
	42	Rounding Method		NAVW15.00	28-02-07	12:00:00	
	43	Purch. Comment Line		NAVW15.00	28-02-07	12:00:00	
	44	Sales Comment Line		NAVW15.00	28-02-07	12:00:00	
	45	G/L Register		NAVW15.00	28-02-07	12:00:00	
	46	Item Register		NAVW15.00	28-02-07	12:00:00	
	47	Aging Band Buffer		NAVW15.00	28-02-07	12:00:00	
	48	Invt. Posting Buffer		NAVW15.00	28-02-07	12:00:00	

To see what has changed you can compare the modified objects with the original using your compare tool.



Voor.txt [Beyond Compare]
File Edit Search View Tools Help
C:\Users\rejer\Desktop\Voor.txt
12-3-2009 13:40:46
C:\Users\rejer\Desktop\Na.txt
12-3-2009 13:40:26
2157
Exact
2157
12 Section(s) Different

```
OBJECT Table 37 Sales Line
{
  OBJECT-PROPERTIES
  {
    Date=28-02-07;
    Time=12:00:00;
    Version List=NAVW15.00;
  }
  PROPERTIES
  {
    OnInsert=VAR
               DocDim@1000 : Record 357;
                                                            OnValidate=BEGIN
                                                                         TestStatusOpen;
                                                                         UpdateAmounts;
                                                                       END;

                                                            CaptionML=ENU=Prepayment Tax Group Code }
    { 121 ;   ;Prepmt Amt to Deduct;Decimal       ;OnValidate=VAR
                                                                         ApplMgt@1000 : Codeunit 1;
                                                                       BEGIN
                                                                         IF "Prepmt Amt to Deduct" > "P
                                                                           FIELDERROR(
                                                                             "Prepmt Amt to Deduct",
                                                                             STRSUBSTNO(Text045,"Prepmt
                                                                       END;
      Text009@1008 : TextConst 'ENU=" must be 0 when %1 is %2."';
      Text011@1010 : TextConst 'ENU=Automatic reservation is not possible.\Reserve items manua
      Text012@1011 : TextConst 'ENU=Change %1 from %2 to %3?';
      Text014@1013 : TextConst 'ENU=%1 %2 is before Work Date %3';
      Text016@1040 : TextConst 'ENU="%1 is required for %2 = %3."';
      Text017@1044 : TextConst 'ENU=\The entered information will be disregarded by warehouse
      Text018@1017 : TextConst 'ENU="must not be specified when %1 = %2"';
      Text020@1019 : TextConst 'ENU=You cannot return more than %1 units.';
      Text021@1020 : TextConst 'ENU=You cannot return more than %1 base units.';
      Text023@1022 : TextConst 'ENU=%1 %2 cannot be found in the %3 or %4 table.';
      Text024@1023 : TextConst 'ENU=%1 and %2 cannot both be empty when %3 is used.';
      Text025@1024 : TextConst 'ENU=No %1 has been posted for %2 %3 and %4 %5.';
      Text026@1025 : TextConst 'ENU=You cannot change %1 if the item charge has already been p
      CurrExchRate@1030 : Record 330;
      SalesHeader@1031 : Record 36;
      SalesLine2@1032 : Record 37;
      TempSalesLine@1033 : Record 37;
      GLAcc@1035 : Record 15;
```

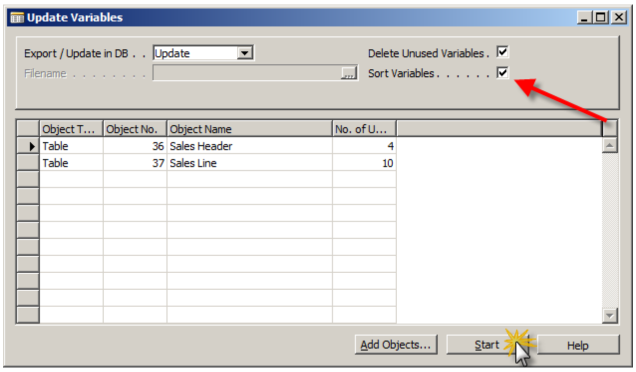
```
OBJECT Table 37 Sales Line
{
  OBJECT-PROPERTIES
  {
    Date=12-03-09;
    Time=13:39:23;
    Modified=Yes;
    Version List=NAVW15.00;
  }
  PROPERTIES
  {
    OnInsert=VAR
               DocDim@1000 : Record 357;
                                                            OnValidate=BEGIN
                                                                         TestStatusOpen;
                                                                         UpdateAmounts;
                                                                       END;

                                                            CaptionML=ENU=Prepayment Tax Group Code }
    { 121 ;   ;Prepmt Amt to Deduct;Decimal       ;OnValidate=BEGIN

                                                                         IF "Prepmt Amt to Deduct" > "P
                                                                           FIELDERROR(
                                                                             "Prepmt Amt to Deduct",
                                                                             STRSUBSTNO(Text045,"Prepmt
                                                                       END;
      Text009@1008 : TextConst 'ENU=" must be 0 when %1 is %2."';
      Text011@1010 : TextConst 'ENU=Automatic reservation is not possible.\Reserve items manua
      Text012@1011 : TextConst 'ENU=Change %1 from %2 to %3?';
      Text014@1013 : TextConst 'ENU=%1 %2 is before Work Date %3';
      Text016@1040 : TextConst 'ENU="%1 is required for %2 = %3."';
      Text017@1044 : TextConst 'ENU=\The entered information will be disregarded by warehouse

      Text020@1019 : TextConst 'ENU=You cannot return more than %1 units.';
      Text021@1020 : TextConst 'ENU=You cannot return more than %1 base units.';



      Text026@1025 : TextConst 'ENU=You cannot change %1 if the item charge has already been p
      CurrExchRate@1030 : Record 330;
      SalesHeader@1031 : Record 36;
      SalesLine2@1032 : Record 37;
      TempSalesLine@1033 : Record 37;
      GLAcc@1035 : Record 15;
```

```
      Text045@1081 : TextConst 'ENU="You can not change the %1 field because %2 %3 has %4 = %5 and the %6 has already been assigned %7 %8."';
      Text045@1081 : TextConst 'ENU="You can not change the %1 field because %2 %3 has %4 = %5 and the %6 has already been assigned %7 %8."';
```

23.2. Sort Variables

With “Update Variables” you can sort the variables in the object.

Set checkbox “Sort Variables”. Add the objects for which you want to sort variables and press Start.



Export / Update in DB . . Update
Filename . . . .
Delete Unused Variables . ☑
Sort Variables . . . . . ☑

Object T...	Object No.	Object Name	No. of U...
Table	36	Sales Header	4
Table	37	Sales Line	10

Before:

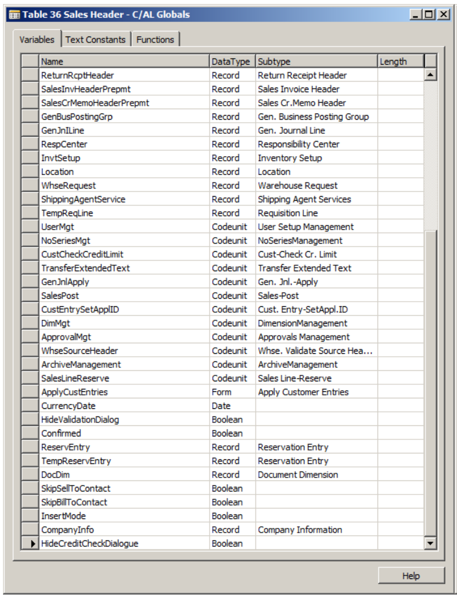


Table 36 Sales Header - C/AL Globals

Name	DataType	Subtype	Length
ReturnRcptHeader	Record	Return Receipt Header	
SalesInvHeaderPrepmt	Record	Sales Invoice Header	
SalesCrMemoHeaderPrepmt	Record	Sales Cr.Memo Header	
GenBusPostingGrp	Record	Gen. Business Posting Group	
GenJnlLine	Record	Gen. Journal Line	
RespCenter	Record	Responsibility Center	
InvtSetup	Record	Inventory Setup	
Location	Record	Location	
WhseRequest	Record	Warehouse Request	
ShippingAgentService	Record	Shipping Agent Services	
TempReqLine	Record	Requisition Line	
UserMgt	Codeunit	User Setup Management	
NoSeriesMgt	Codeunit	NoSeriesManagement	
CustCheckCreditLimit	Codeunit	Cust-Check Cr. Limit	
TransferExtendedText	Codeunit	Transfer Extended Text	
GenJnlApply	Codeunit	Gen. Jnl.-Apply	
SalesPost	Codeunit	Sales-Post	
CustEntrySetApplID	Codeunit	Cust. Entry-SetAppl.ID	
DimMgt	Codeunit	DimensionManagement	
ApprovalMgt	Codeunit	Approvals Management	
WhseSourceHeader	Codeunit	Whse. Validate Source Hea...	
ArchiveManagement	Codeunit	ArchiveManagement	
SalesLineReserve	Codeunit	Sales Line-Reserve	
ApplyCustEntries	Form	Apply Customer Entries	
CurrencyDate	Date		
HideValidationDialog	Boolean		
Confirmed	Boolean		
ReservEntry	Record	Reservation Entry	
TempReservEntry	Record	Reservation Entry	
DocDim	Record	Document Dimension	
SkipSellToContact	Boolean		
SkipBillToContact	Boolean		
InsertMode	Boolean		
CompanyInfo	Record	Company Information	
HideCreditCheckDialogue	Boolean		

After:

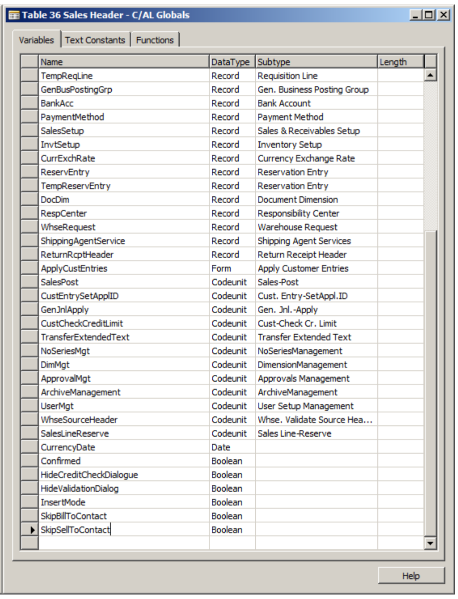


Table 36 Sales Header - C/AL Globals

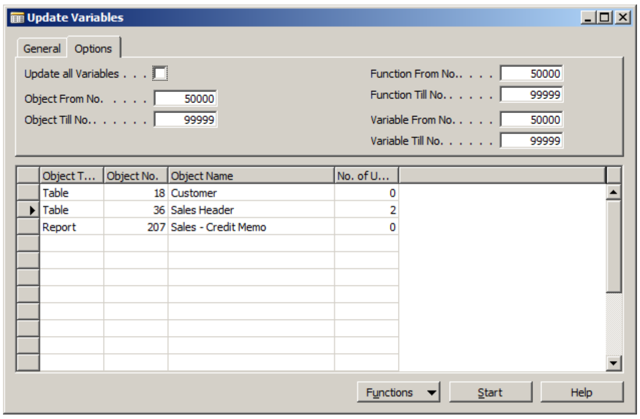
Name	DataType	Subtype	Length
TempReqLine	Record	Requisition Line	
GenBusPostingGrp	Record	Gen. Business Posting Group	
BankAcc	Record	Bank Account	
PaymentMethod	Record	Payment Method	
SalesSetup	Record	Sales & Receivables Setup	
InvtSetup	Record	Inventory Setup	
CurrExchRate	Record	Currency Exchange Rate	
ReservEntry	Record	Reservation Entry	
TempReservEntry	Record	Reservation Entry	
DocDim	Record	Document Dimension	
RespCenter	Record	Responsibility Center	
WhseRequest	Record	Warehouse Request	
ShippingAgentService	Record	Shipping Agent Services	
ReturnRcptHeader	Record	Return Receipt Header	
ApplyCustEntries	Form	Apply Customer Entries	
SalesPost	Codeunit	Sales-Post	
CustEntrySetApplID	Codeunit	Cust. Entry-SetAppl.ID	
GenJnlApply	Codeunit	Gen. Jnl.-Apply	
CustCheckCreditLimit	Codeunit	Cust-Check Cr. Limit	
TransferExtendedText	Codeunit	Transfer Extended Text	
NoSeriesMgt	Codeunit	NoSeriesManagement	
DimMgt	Codeunit	DimensionManagement	
ApprovalMgt	Codeunit	Approvals Management	
ArchiveManagement	Codeunit	ArchiveManagement	
UserMgt	Codeunit	User Setup Management	
WhseSourceHeader	Codeunit	Whse. Validate Source Hea...	
SalesLineReserve	Codeunit	Sales Line-Reserve	
CurrencyDate	Date		
Confirmed	Boolean		
HideCreditCheckDialogue	Boolean		
HideValidationDialog	Boolean		
InsertMode	Boolean		
SkipBillToContact	Boolean		
SkipSellToContact	Boolean		

Sorting Order

The variables will be first sorted on Type then "Object No." and last Name.

1	Record	19	OutStream
2	Form	20	DateFormula
3	Report	21	Date
4	Dataport	22	DateTime
5	XMLPort	23	Time
6	Codeunit	24	Duration
7	MenuSuite	25	Text
8	Page	26	Code
9	Automation	27	GUID
10	OCX	28	Char
11	Variant	29	Decimal
12	Binary	30	BigInteger
13	Dialog	31	Integer
14	File	32	Option
15	RecordRef	33	Action
16	FieldRef	34	Boolean
17	KeyRef	35	TextConst
18	Instream		

23.3. Set Variable Range



Update all Variables ☐

Object From No. 50000

Object Till No. 99999

Function From No. 50000

Function Till No. 99999

Variable From No. 50000

Variable Till No. 99999

Object T...	Object No.	Object Name	No. of U...
Table	18	Customer	0
Table	36	Sales Header	2
Report	207	Sales - Credit Memo	0

You can set a range or choose to update all variables. This makes it possible to update standard NAV objects and leave the standard variables as they are.

24. Change Field Options

With this tool you can update or change option fields in the database including the involved objects. All references to this field are also updated.

There are three possible modes.

- Update in DB: Options are changed in the database.
- Export: Objects are exported in text format and options are changed the exported file.
- Update text file: Options are changed in a text file.

NOTE: To use the first two modes you first have to update the "Where Used Objects". For more information see section 15.1 - Setup.

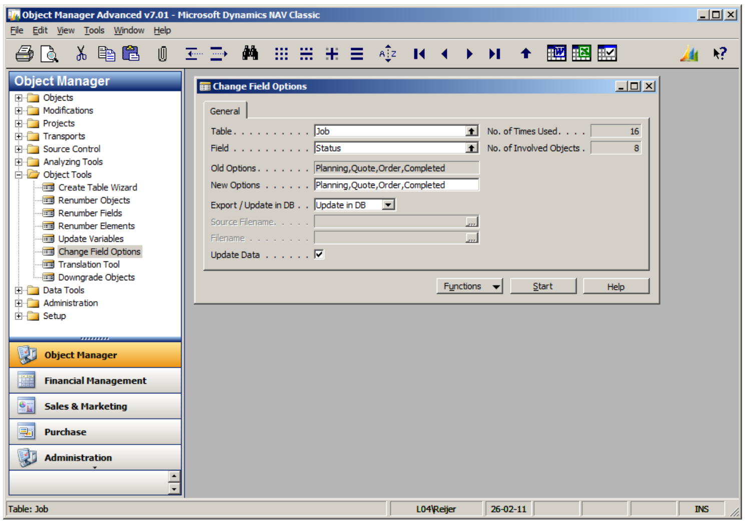


Table Job

Field Status

No. of Times Used 16

No. of Involved Objects 8

Old Options Planning,Quote,Order,Completed

New Options Planning,Quote,Order,Completed

Export / Update in DB Update in DB

Source Filename

Filename

Update Data ☑

Select the table and the field you want to change and alter the option string the way you like. It is possible to delete, insert, swap or rename an option. The captions will also be updated.

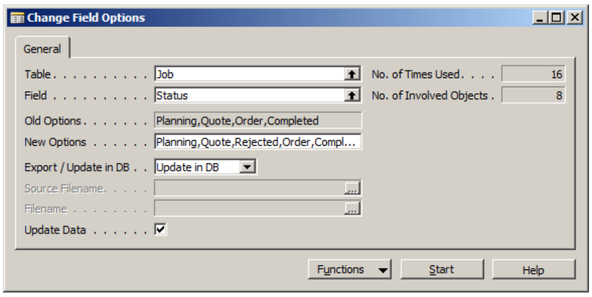


Table . . . . Job

Field . . . . Status

No. of Times Used. . . . 16

No. of Involved Objects . 8

Old Options . . . . Planning,Quote,Order,Completed

New Options . . . . . . Planning,Quote,Rejected,Order,Compl...

Export / Update in DB . . Update in DB

Source Filename. . . . .

Filename . . . .

Update Data . . . . . . ☑

If you change an existing option you can use the “No. of Times Used” option to check whether you have to modify more than only the references.

25. Renumber Elements

With this tool you can renumber the elements of an object. Elements are functions, variables and controls. When another object has a connection with the renumbered element the reference(s) in these objects will also be changed.

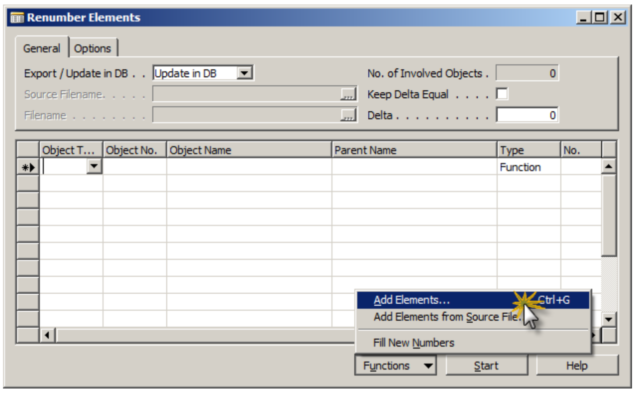
There are three possible modes.

- Update in DB
  Elements are renumbered in the database.
- Export
  Objects are exported in text format and elements are renumbered in the exported file.
- Update Text File
  Elements are renumbered in a text file.

NOTE: To use the first two modes you first have to update the "Where Used Objects". For more information see [section 15.1 - Update "Where Used Objects"].

25.1. Update in DB Mode

Press Functions > Add Elements.



Renumber Elements

Export / Update in DB . . Update in DB

Source Filename. . . . .

Filename . . . .

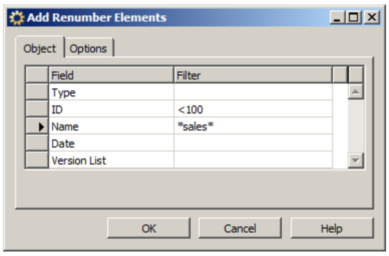
No. of Involved Objects . 0

Keep Delta Equal . . . . ☐

Delta 0

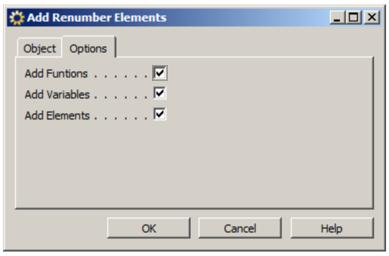
Object T...	Object No.	Object Name	Parent Name	Type	No.
				Function	

Select filters for the objects from which you want to renumber the elements.



Add Renumber Elements

Field	Filter
Type	
ID	<100
Name	*sales*
Date	
Version List	

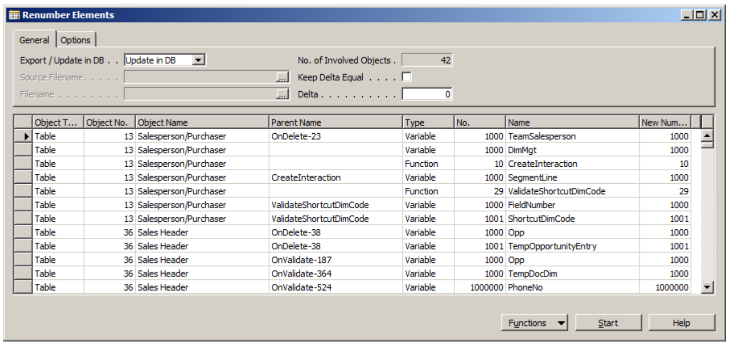


Add Renumber Elements

Add Funtions . . . . . . ☑

Add Variables . . . . . . ☑

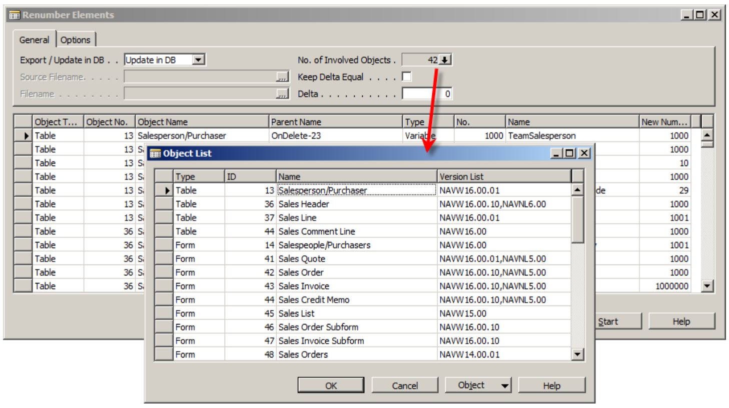
Add Elements . . . . . . ☑



Export / Update in DB: Update in DB
Source Filename:
Filename:
No. of Involved Objects: 42
Keep Delta Equal: ☐
Delta: 0

Object T...	Object No.	Object Name	Parent Name	Type	No.	Name	New Num...
Table	13	Salesperson/Purchaser	OnDelete-23	Variable	1000	TeamSalesperson	1000
Table	13	Salesperson/Purchaser		Variable	1000	DimMgt	1000
Table	13	Salesperson/Purchaser		Function	10	CreateInteraction	10
Table	13	Salesperson/Purchaser	CreateInteraction	Variable	1000	SegmentLine	1000
Table	13	Salesperson/Purchaser		Function	29	ValidateShortcutDimCode	29
Table	13	Salesperson/Purchaser	ValidateShortcutDimCode	Variable	1000	FieldNumber	1000
Table	13	Salesperson/Purchaser	ValidateShortcutDimCode	Variable	1001	ShortcutDimCode	1001
Table	36	Sales Header	OnDelete-38	Variable	1000	Opp	1000
Table	36	Sales Header	OnDelete-38	Variable	1001	TempOpportunityEntry	1001
Table	36	Sales Header	OnValidate-187	Variable	1000	Opp	1000
Table	36	Sales Header	OnValidate-364	Variable	1000	TempDocDim	1000
Table	36	Sales Header	OnValidate-524	Variable	1000000	PhoneNo	1000000

In field “No. of Involved Objects” you can see which objects will be updated when you renumber your selection of objects.

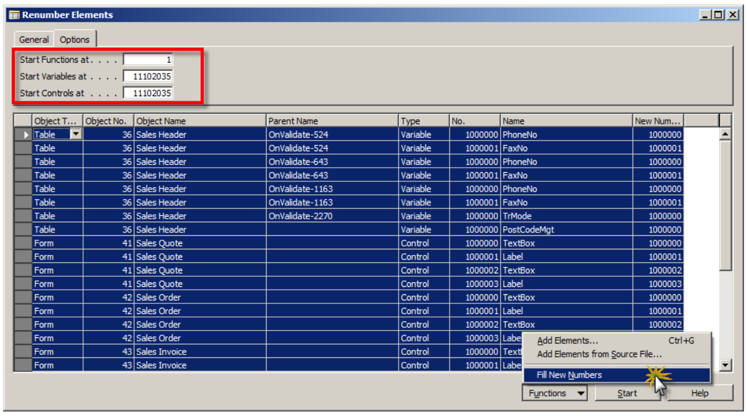


Export / Update in DB: Update in DB
No. of Involved Objects: 42
Keep Delta Equal: ☐
Delta: 0

Object List

Type	ID	Name	Version List
Table	13	Salesperson/Purchaser	NAVW16.00.01
Table	36	Sales Header	NAVW16.00.10,NAVNL6.00
Table	37	Sales Line	NAVW16.00.01
Table	44	Sales Comment Line	NAVW16.00
Form	14	Salespeople/Purchasers	NAVW16.00
Form	41	Sales Quote	NAVW16.00.01,NAVNL5.00
Form	42	Sales Order	NAVW16.00.10,NAVNL5.00
Form	43	Sales Invoice	NAVW16.00.10,NAVNL5.00
Form	44	Sales Credit Memo	NAVW16.00.10,NAVNL5.00
Form	45	Sales List	NAVW15.00
Form	46	Sales Order Subform	NAVW16.00.10
Form	47	Sales Invoice Subform	NAVW16.00.10
Form	48	Sales Orders	NAVW14.00.01

To give the elements a new number you can modify the lines manually but you can also select the lines you want to renumber and fill in the “Start at” fields on the Options tab and press Functions > Fill New Numbers.



Start Functions at: 1
Start Variables at: 11102035
Start Controls at: 11102035

Object T...	Object No.	Object Name	Parent Name	Type	No.	Name	New Num...
Table	36	Sales Header	OnValidate-524	Variable	1000000	PhoneNo	1000000
Table	36	Sales Header	OnValidate-524	Variable	1000001	FaxNo	1000001
Table	36	Sales Header	OnValidate-643	Variable	1000000	PhoneNo	1000000
Table	36	Sales Header	OnValidate-643	Variable	1000001	FaxNo	1000001
Table	36	Sales Header	OnValidate-1163	Variable	1000000	PhoneNo	1000000
Table	36	Sales Header	OnValidate-1163	Variable	1000001	FaxNo	1000001
Table	36	Sales Header	OnValidate-2270	Variable	1000000	TrMode	1000000
Table	36	Sales Header		Variable	1000000	PostCodeMgt	1000000
Form	41	Sales Quote		Control	1000000	TextBox	1000000
Form	41	Sales Quote		Control	1000001	Label	1000001
Form	41	Sales Quote		Control	1000002	TextBox	1000002
Form	41	Sales Quote		Control	1000003	Label	1000003
Form	42	Sales Order		Control	1000000	TextBox	1000000
Form	42	Sales Order		Control	1000001	Label	1000001
Form	42	Sales Order		Control	1000002	TextBox	1000002
Form	42	Sales Order		Control	1000003	Labe	
Form	43	Sales Invoice		Control	1000000	Text	
Form	43	Sales Invoice		Control	1000001	Labe	

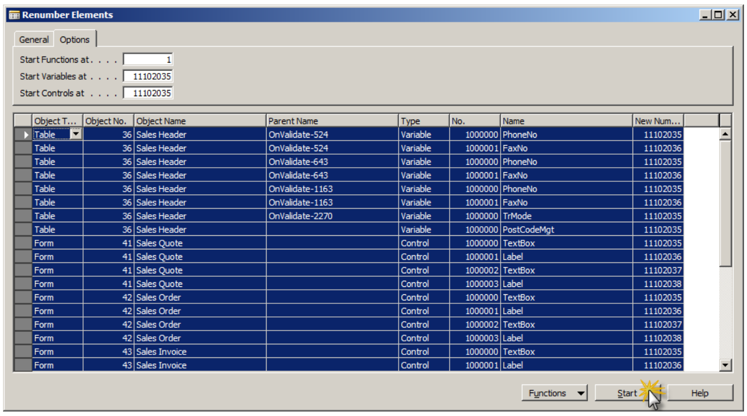
The elements will get their new number. Press Start to modify the objects.

NOTE: The “Start … at” fields on the Options tab are populated based on:

- **“UID Offset” field on the “Project Type Card” window in case modifications are traced for an active project or**
- **“Default UID Offset” field on the “Object Manager Setup” window**

If nothing is defined, the values will default to:

- **1 for “Start Functions at” and "Start Controls at”**
- **1000 for “Start Variables at”**

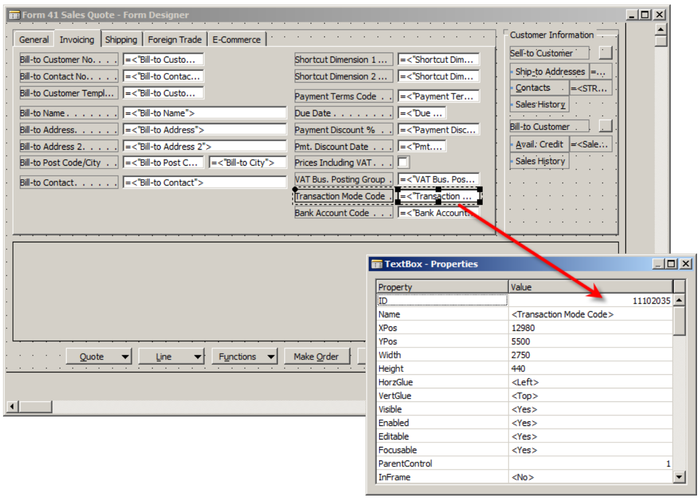


Renumber Elements

Start Functions at. . . . 1
Start Variables at 11102035
Start Controls at 11102035

Object T...	Object No.	Object Name	Parent Name	Type	No.	Name	New Num...
Table	36	Sales Header	OnValidate-524	Variable	1000000	PhoneNo	11102035
Table	36	Sales Header	OnValidate-524	Variable	1000001	FaxNo	11102036
Table	36	Sales Header	OnValidate-643	Variable	1000000	PhoneNo	11102035
Table	36	Sales Header	OnValidate-643	Variable	1000001	FaxNo	11102036
Table	36	Sales Header	OnValidate-1163	Variable	1000000	PhoneNo	11102035
Table	36	Sales Header	OnValidate-1163	Variable	1000001	FaxNo	11102036
Table	36	Sales Header	OnValidate-2270	Variable	1000000	TrMode	11102035
Table	36	Sales Header		Variable	1000000	PostCodeMgt	11102035
Form	41	Sales Quote		Control	1000000	TextBox	11102035
Form	41	Sales Quote		Control	1000001	Label	11102036
Form	41	Sales Quote		Control	1000002	TextBox	11102037
Form	41	Sales Quote		Control	1000003	Label	11102038
Form	42	Sales Order		Control	1000000	TextBox	11102035
Form	42	Sales Order		Control	1000001	Label	11102036
Form	42	Sales Order		Control	1000002	TextBox	11102037
Form	42	Sales Order		Control	1000003	Label	11102038
Form	43	Sales Invoice		Control	1000000	TextBox	11102035
Form	43	Sales Invoice		Control	1000001	Label	11102036

All the elements will be renumbered.



Form 41 Sales Quote - Form Designer

Bill-to Customer No. . . . =<"Bill-to Custo...
Bill-to Contact No.. . . . =<"Bill-to Contac...
Bill-to Customer Templ... =<"Bill-to Custo...
Bill-to Name . . . . =<"Bill-to Name">
Bill-to Address. . . . . . =<"Bill-to Address">
Bill-to Address 2. . . . . =<"Bill-to Address 2">
Bill-to Post Code/City . . =<"Bill-to Post C... =<"Bill-to City">
Bill-to Contact. . . . . . =<"Bill-to Contact">
Shortcut Dimension 1 ... =<"Shortcut Dim...
Shortcut Dimension 2 ... =<"Shortcut Dim...
Payment Terms Code . . =<"Payment Ter...
Due Date . . . . =<"Due ...
Payment Discount % . . =<"Payment Disc...
Pmt. Discount Date . . . =<"Pmt....
Prices Including VAT . . . ☐
VAT Bus. Posting Group . =<"VAT Bus. Pos...
Transaction Mode Code . =<"Transaction ...
Bank Account Code . . . =<"Bank Account...

TextBox - Properties

Property	Value
ID	11102035
Name	<Transaction Mode Code>
XPos	12980
YPos	5500
Width	2750
Height	440
HorzGlue	<Left>
VertGlue	<Top>
Visible	<Yes>
Enabled	<Yes>
Editable	<Yes>
Focusable	<Yes>
ParentControl	1
InFrame	<No>

The lines in the renumber form can have the following colors:

- Black: New number is equal to the original number
- Purple: New number is two times present
- Green: Field will be renumbered

25.1. Export Mode

This option works the same as the Updating option, but the element in the original database are not renumbered. The renumbering is only done in the export file. The exported file is a text file which you can import in the Object Designer.

25.2. Update Text File Mode

This mode works the same as the other two modes but the elements are renumbered for an existing text file. You enter a filename for the source file and a filename for the file that will be created with the renumbered elements. You can add the elements of the source file with the menu option “Add Elements from Source File”.

26. Translation Tool

With the Translation Tool you can easily translate all captions and names that you have used in your objects. It can even use translations used in existing objects to translate missing values automatically.

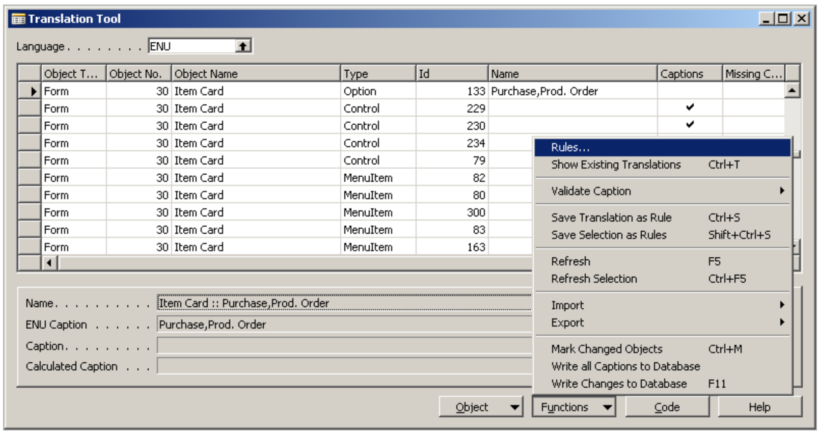
26.1. Setup

26.1.1. Captions

To translate the captions used in your objects you first have to add them to the Translation Tool. Press Object – Add Objects … and run the Import Captions batch job.
Now that you have abstracted all the captions you can use them to create translation rules.

26.1.2. Rules

Rules are used to make suggestions for a translation and look if there are any deviating translations in your objects.
Press Functions > Rules…

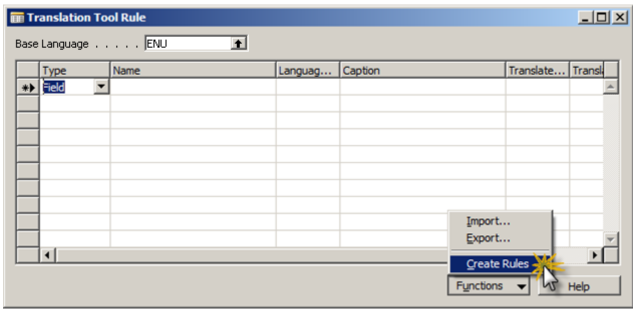

Translation Tool
Rules…
Show Existing Translations Ctrl+T
Validate Caption
Save Translation as Rule Ctrl+S
Save Selection as Rules Shift+Ctrl+S
Refresh F5
Refresh Selection Ctrl+F5
Import
Export
Mark Changed Objects Ctrl+M
Write all Captions to Database
Write Changes to Database F11
Object
Functions
Code
Help

Language . . . . ENU

Object T…	Object No.	Object Name	Type	Id	Name	Captions	Missing C…
Form	30	Item Card	Option	133	Purchase,Prod. Order		
Form	30	Item Card	Control	229		✓	
Form	30	Item Card	Control	230		✓	
Form	30	Item Card	Control	234			
Form	30	Item Card	Control	79			
Form	30	Item Card	MenuItem	82			
Form	30	Item Card	MenuItem	80			
Form	30	Item Card	MenuItem	300			
Form	30	Item Card	MenuItem	83			
Form	30	Item Card	MenuItem	163			

Name. . . . . Item Card :: Purchase,Prod. Order
ENU Caption . . . . . . Purchase,Prod. Order
Caption. . . . .
Calculated Caption . . .

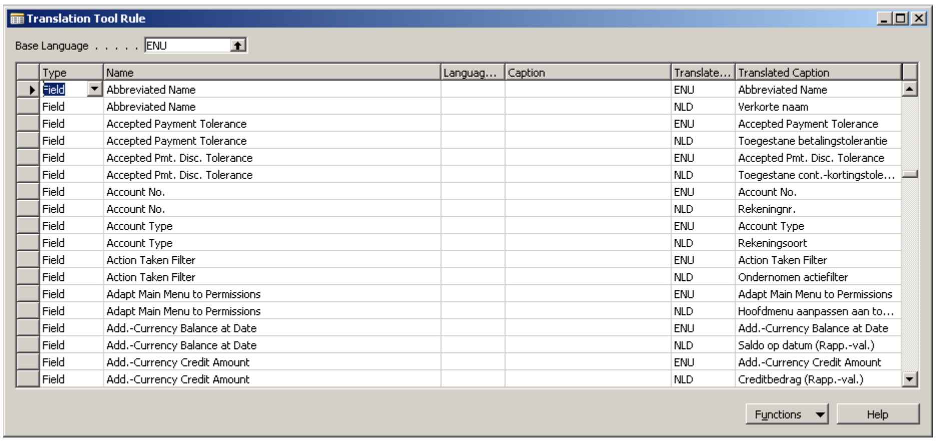
And Functions > Create Rules.



Base Language . . . . . ENU

Type	Name	Languag…	Caption	Translate…	Transl
Field					

The Translation Tool will now scan all the existing translations and create all possible rules. It is possible to manually add your own rules and import and export them to a text file using the import and export options under the functions buttons.



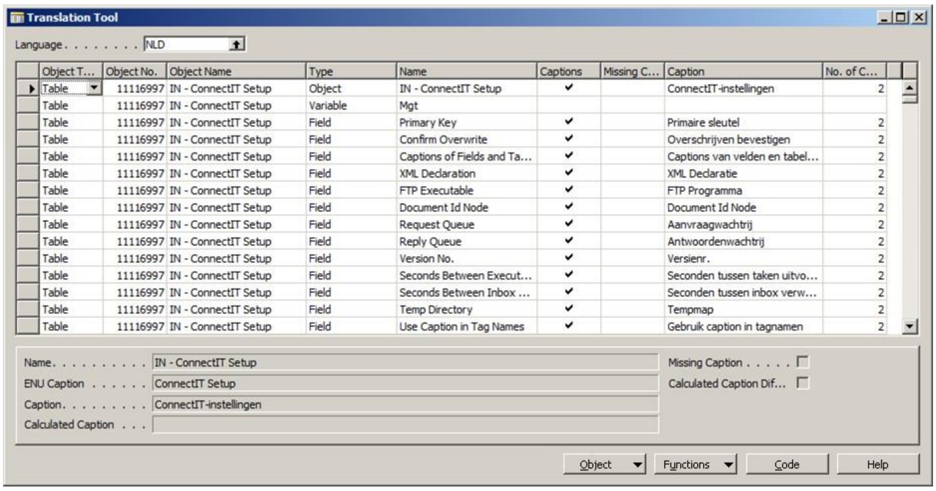
Base Language . . . . . ENU

Type	Name	Languag…	Caption	Translate…	Translated Caption
Field	Abbreviated Name			ENU	Abbreviated Name
Field	Abbreviated Name			NLD	Verkorte naam
Field	Accepted Payment Tolerance			ENU	Accepted Payment Tolerance
Field	Accepted Payment Tolerance			NLD	Toegestane betalingstolerantie
Field	Accepted Pmt. Disc. Tolerance			ENU	Accepted Pmt. Disc. Tolerance
Field	Accepted Pmt. Disc. Tolerance			NLD	Toegestane cont.-kortingstole…
Field	Account No.			ENU	Account No.
Field	Account No.			NLD	Rekeningnr.
Field	Account Type			ENU	Account Type
Field	Account Type			NLD	Rekeningsoort
Field	Action Taken Filter			ENU	Action Taken Filter
Field	Action Taken Filter			NLD	Ondernomen actiefilter
Field	Adapt Main Menu to Permissions			ENU	Adapt Main Menu to Permissions
Field	Adapt Main Menu to Permissions			NLD	Hoofdmenu aanpassen aan to…
Field	Add.-Currency Balance at Date			ENU	Add.-Currency Balance at Date
Field	Add.-Currency Balance at Date			NLD	Saldo op datum (Rapp.-val.)
Field	Add.-Currency Credit Amount			ENU	Add.-Currency Credit Amount
Field	Add.-Currency Credit Amount			NLD	Creditbedrag (Rapp.-val.)

26.2. Translate Captions

Now the translation tool is setup to translate the captions used in your objects.

And you will see an overview of all the translations that are present in you objects.

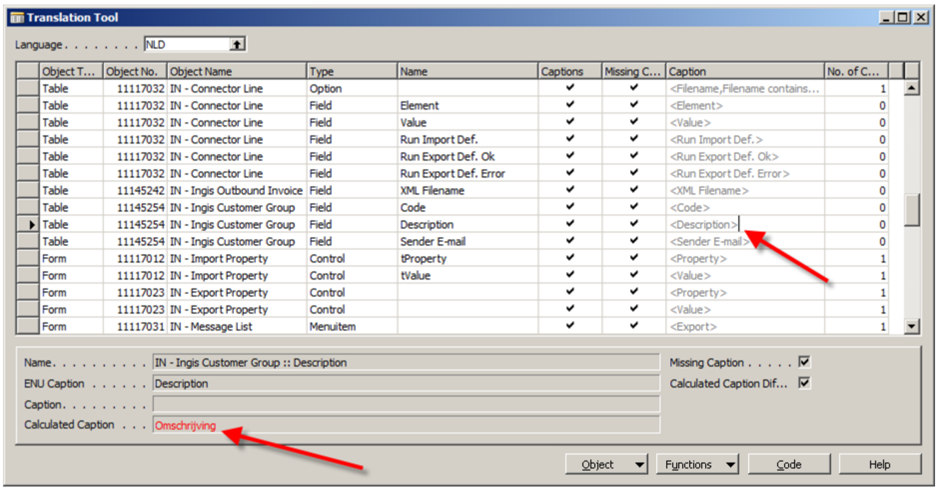


Language . . . . NLD

Object T…	Object No.	Object Name	Type	Name	Captions	Missing C…	Caption	No. of C…
Table	11116997	IN - ConnectIT Setup	Object	IN - ConnectIT Setup	✓		ConnectIT-instellingen	2
Table	11116997	IN - ConnectIT Setup	Variable	Mgt				
Table	11116997	IN - ConnectIT Setup	Field	Primary Key	✓		Primaire sleutel	2
Table	11116997	IN - ConnectIT Setup	Field	Confirm Overwrite	✓		Overschrijven bevestigen	2
Table	11116997	IN - ConnectIT Setup	Field	Captions of Fields and Ta…	✓		Captions van velden en tabel…	2
Table	11116997	IN - ConnectIT Setup	Field	XML Declaration	✓		XML Declaratie	2
Table	11116997	IN - ConnectIT Setup	Field	FTP Executable	✓		FTP Programma	2
Table	11116997	IN - ConnectIT Setup	Field	Document Id Node	✓		Document Id Node	2
Table	11116997	IN - ConnectIT Setup	Field	Request Queue	✓		Aanvraagwachtrij	2
Table	11116997	IN - ConnectIT Setup	Field	Reply Queue	✓		Antwoordenwachtrij	2
Table	11116997	IN - ConnectIT Setup	Field	Version No.	✓		Versienr.	2
Table	11116997	IN - ConnectIT Setup	Field	Seconds Between Execut…	✓		Seconden tussen taken uitvo…	2
Table	11116997	IN - ConnectIT Setup	Field	Seconds Between Inbox …	✓		Seconden tussen inbox verw…	2
Table	11116997	IN - ConnectIT Setup	Field	Temp Directory	✓		Tempmap	2
Table	11116997	IN - ConnectIT Setup	Field	Use Caption in Tag Names	✓		Gebruik caption in tagnamen	2

Name. . . . . IN - ConnectIT Setup
ENU Caption . . . . . . ConnectIT Setup
Caption. . . . . ConnectIT-instellingen
Calculated Caption . . .
Missing Caption . . . . . ☐
Calculated Caption Dif… ☐

To see what captions are missing you can set a filter on the field “Missing Caption”. In this example you see that the field description does not have a Dutch (NLD) translation. You also see that the Translation Tool has found a rule for the value Description.



Language . . . . NLD

Object T...	Object No.	Object Name	Type	Name	Captions	Missing C...	Caption	No. of C...
Table	11117032	IN - Connector Line	Option		✓	✓	<Filename,Filename contains...	1
Table	11117032	IN - Connector Line	Field	Element	✓	✓	<Element>	0
Table	11117032	IN - Connector Line	Field	Value	✓	✓	<Value>	0
Table	11117032	IN - Connector Line	Field	Run Import Def.	✓	✓	<Run Import Def.>	0
Table	11117032	IN - Connector Line	Field	Run Export Def. Ok	✓	✓	<Run Export Def. Ok>	0
Table	11117032	IN - Connector Line	Field	Run Export Def. Error	✓	✓	<Run Export Def. Error>	0
Table	11145242	IN - Ingis Outbound Invoice	Field	XML Filename	✓	✓	<XML Filename>	0
Table	11145254	IN - Ingis Customer Group	Field	Code	✓	✓	<Code>	0
Table	11145254	IN - Ingis Customer Group	Field	Description	✓	✓	<Description>	0
Table	11145254	IN - Ingis Customer Group	Field	Sender E-mail	✓	✓	<Sender E-mail>	0
Form	11117012	IN - Import Property	Control	tProperty	✓	✓	<Property>	1
Form	11117012	IN - Import Property	Control	tValue	✓	✓	<Value>	1
Form	11117023	IN - Export Property	Control		✓	✓	<Property>	1
Form	11117023	IN - Export Property	Control		✓	✓	<Value>	1
Form	11117031	IN - Message List	Menuitem		✓	✓	<Export>	1

Name. . . . . IN - Ingis Customer Group :: Description
ENU Caption . . . . . . Description
Caption. . . . .
Calculated Caption . . . Omschrijving
Missing Caption . . . . . ☑
Calculated Caption Dif... ☑

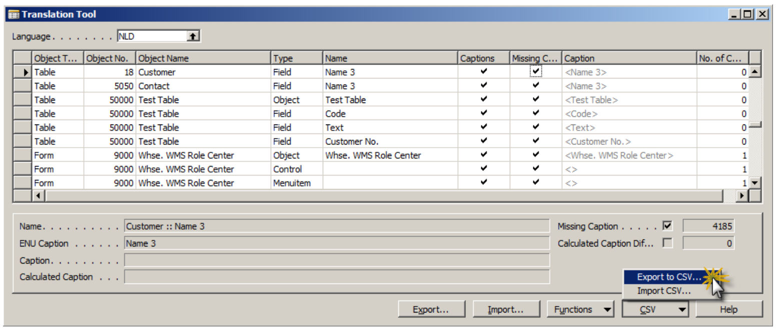
Either you translate a caption manually or you let the translation tool help you by pressing Functions – Write – Validate Captions and then choosing one of the following options:

- With Calculated Caption
  this will populate the caption with the value as suggested in the Calculated Caption field
- With Base Language Caption
  this will populate the caption with the value available for the base language
- With Captions of Selected Language
  this will let you populate the caption with the value available for any other language
- With Name
  this will populate the caption with the value of the Name property

When you are ready you effectuate the translations by pressing Functions – Write Changes to Database or F11.

26.3. Export and Import to CSV

You can use this function to export and import translations to a CSV file. You can send this for example to a translator. And import the translated CSV file back into the Translation Tool.



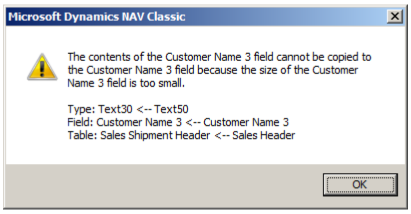
Language . . . . NLD

Object T...	Object No.	Object Name	Type	Name	Captions	Missing C...	Caption	No. of C...
Table	18	Customer	Field	Name 3	✓	✓	<Name 3>	0
Table	5050	Contact	Field	Name 3	✓	✓	<Name 3>	0
Table	50000	Test Table	Object	Test Table	✓	✓	<Test Table>	0
Table	50000	Test Table	Field	Code	✓	✓	<Code>	0
Table	50000	Test Table	Field	Text	✓	✓	<Text>	0
Table	50000	Test Table	Field	Customer No.	✓	✓	<Customer No.>	0
Form	9000	Whse. WMS Role Center	Object	Whse. WMS Role Center	✓	✓	<Whse. WMS Role Center>	1
Form	9000	Whse. WMS Role Center	Control		✓	✓	<>	1
Form	9000	Whse. WMS Role Center	Menuitem		✓	✓	<>	1

Name. . . . . Customer :: Name 3
ENU Caption . . . . . . Name 3
Caption. . . . .
Calculated Caption . . .
Missing Caption . . . . . ☑ 4185
Calculated Caption Dif... ☐ 0

27. Check Transferfields

NAV uses the TRANSFERFIELDS command to copy data from one table to another. When there are conflicting fieldtypes in this tables you will get an error like this.



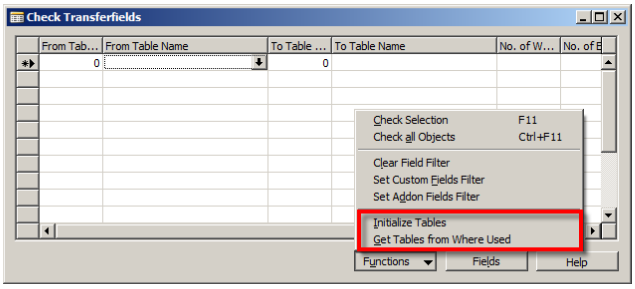
The contents of the Customer Name 3 field cannot be copied to the Customer Name 3 field because the size of the Customer Name 3 field is too small.

Type: Text30 <-- Text50
Field: Customer Name 3 <-- Customer Name 3
Table: Sales Shipment Header <-- Sales Header

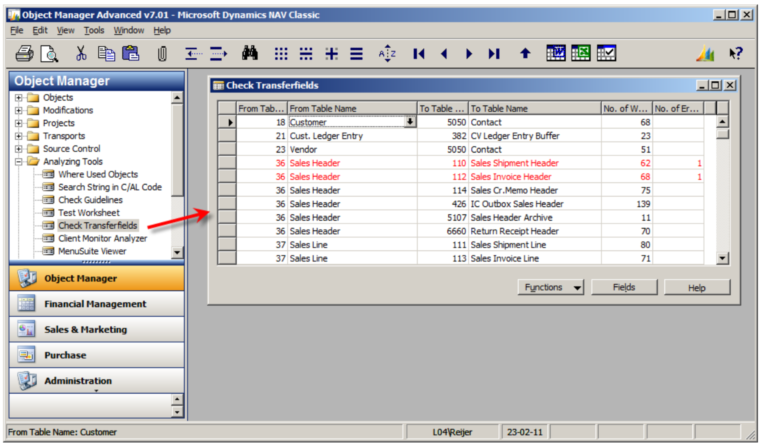
To check if you have any of these conflicting fieldtypes in your database you can use the Check Transferfields form.

27.1. Initialize Transferfields Tables

There are two methods to populate the tables to check. The first one is to use the option "Initialize Tables". This option fills in the tables that uses the transferfields command in a default Cronus database. If you have customizations in your database it is better to use the option "Get Tables from Where Used". This option search your where used base for the transferfields command.



27.2. Transferfields Conflicts



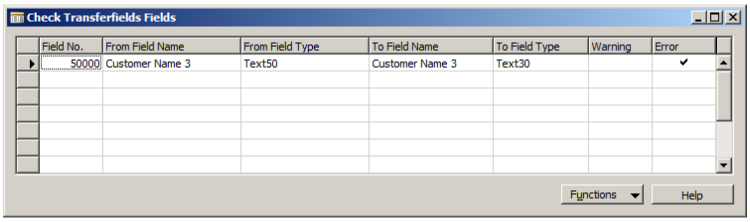
From Tab...	From Table Name	To Table ...	To Table Name	No. of W...	No. of Er...
18	Customer	5050	Contact	68	
21	Cust. Ledger Entry	382	CV Ledger Entry Buffer	23	
23	Vendor	5050	Contact	51	
36	Sales Header	110	Sales Shipment Header	62	1
36	Sales Header	112	Sales Invoice Header	68	1
36	Sales Header	114	Sales Cr.Memo Header	75	
36	Sales Header	426	IC Outbox Sales Header	139	
36	Sales Header	5107	Sales Header Archive	11	
36	Sales Header	6660	Return Receipt Header	70	
37	Sales Line	111	Sales Shipment Line	80	
37	Sales Line	113	Sales Invoice Line	71	

From Table Name: Customer

L04\Reijer

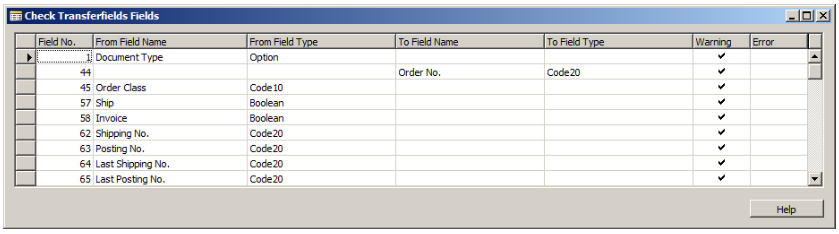
23-02-11

In the above example you can see that there is a conflict between table Sales Header and table Sales Shipment Header. When you drilldown the “No. of Errors” column you will see which fields cause this conflict.



Field No.	From Field Name	From Field Type	To Field Name	To Field Type	Warning	Error
50000	Customer Name 3	Text50	Customer Name 3	Text30		✓

There are two types of conflicts. Warnings are fields that exist in one of the tables but misses in the other. This can be done with a reason but sometimes it is simply forgotten to add the field to one of the tables.

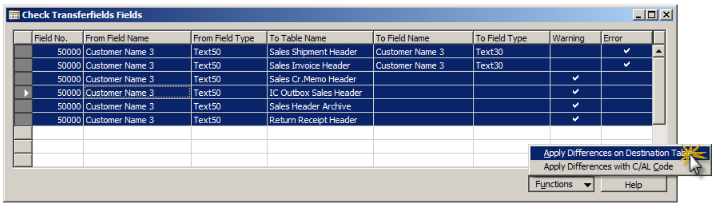


Field No.	From Field Name	From Field Type	To Field Name	To Field Type	Warning	Error
1	Document Type	Option			✓	
44			Order No.	Code20	✓	
45	Order Class	Code10			✓	
57	Ship	Boolean			✓	
58	Invoice	Boolean			✓	
62	Shipping No.	Code20			✓	
63	Posting No.	Code20			✓	
64	Last Shipping No.	Code20			✓	
65	Last Posting No.	Code20			✓	

Conflicts of the type error have to be fixed always.

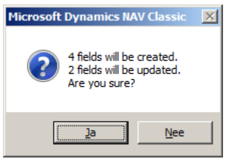
27.3. Apply Differences on Dest. Table

With this function you can create missing fields in the destination table or update existing fields if they have conflicts.


Check Transferfields Fields
Apply Differences on Destination Table
Apply Differences with C/AL Code
Functions
Help

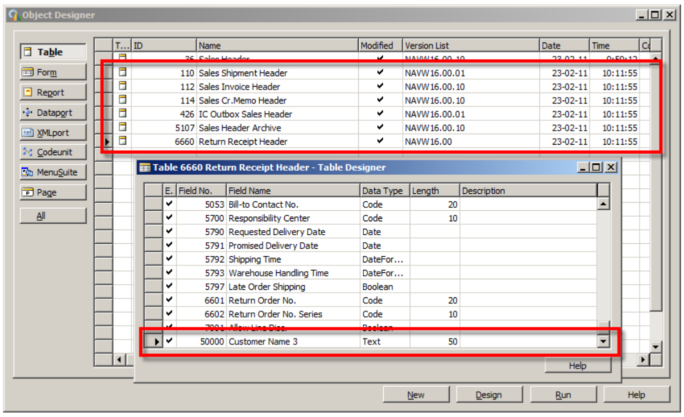
Field No.	From Field Name	From Field Type	To Table Name	To Field Name	To Field Type	Warning	Error
50000	Customer Name 3	Text50	Sales Shipment Header	Customer Name 3	Text30		✓
50000	Customer Name 3	Text50	Sales Invoice Header	Customer Name 3	Text30		✓
50000	Customer Name 3	Text50	Sales Cr.Memo Header			✓	
50000	Customer Name 3	Text50	IC Outbox Sales Header			✓	
50000	Customer Name 3	Text50	Sales Header Archive			✓	
50000	Customer Name 3	Text50	Return Receipt Header			✓	

Select the lines of the fields which you want to transfer to the destination table. In our example the new field “Customer Name 3” that is missing in four tables and has conflicts in two tables.



4 fields will be created.
2 fields will be updated.
Are you sure?

All conflicts are solved in your database.



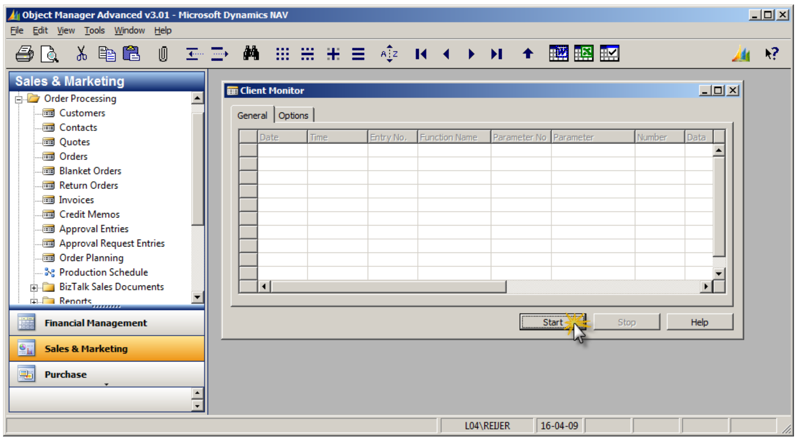
T…	ID	Name	Modified	Version List	Date	Time
	36	Sales Header	✓	NAVW16.00.10	23-02-11	9:50:12
	110	Sales Shipment Header	✓	NAVW16.00.01	23-02-11	10:11:55
	112	Sales Invoice Header	✓	NAVW16.00.10	23-02-11	10:11:55
	114	Sales Cr.Memo Header	✓	NAVW16.00.10	23-02-11	10:11:55
	426	IC Outbox Sales Header	✓	NAVW16.00.01	23-02-11	10:11:55
	5107	Sales Header Archive	✓	NAVW16.00.10	23-02-11	10:11:55
	6660	Return Receipt Header	✓	NAVW16.00	23-02-11	10:11:55

E.	Field No.	Field Name	Data Type	Length	Description
✓	5053	Bill-to Contact No.	Code	20	
✓	5700	Responsibility Center	Code	10	
✓	5790	Requested Delivery Date	Date		
✓	5791	Promised Delivery Date	Date		
✓	5792	Shipping Time	DateFor…		
✓	5793	Warehouse Handling Time	DateFor…		
✓	5797	Late Order Shipping	Boolean		
✓	6601	Return Order No.	Code	20	
✓	6602	Return Order No. Series	Code	10	
✓	7001	Allow Line Disc.	Boolean		
✓	50000	Customer Name 3	Text	50	

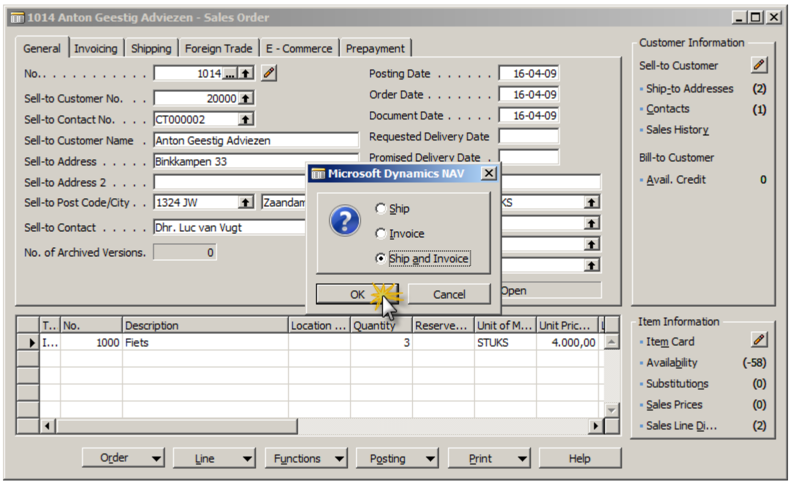
The “Apply Differences with C/AL Code” function also copies the C/AL code from the OnValidate and OnLookup triggers of the source fields to the destination fields.

28. Client Monitor Analyzer

With the client monitor you can analyze all read and write actions that are done. For example when you want to analyze all data actions that are done in the sales posting process you first start the client monitor.



Create a sales order and post it.


1014 Anton Geestig Adviezen - Sales Order

No. 1014
Sell-to Customer No. 20000
Sell-to Contact No. CT000002
Sell-to Customer Name Anton Geestig Adviezen
Sell-to Address Binkkampen 33
Sell-to Address 2
Sell-to Post Code/City 1324 JW Zaandam
Sell-to Contact Dhr. Luc van Vugt
No. of Archived Versions. 0
Posting Date 16-04-09
Order Date 16-04-09
Document Date 16-04-09
Requested Delivery Date
Promised Delivery Date

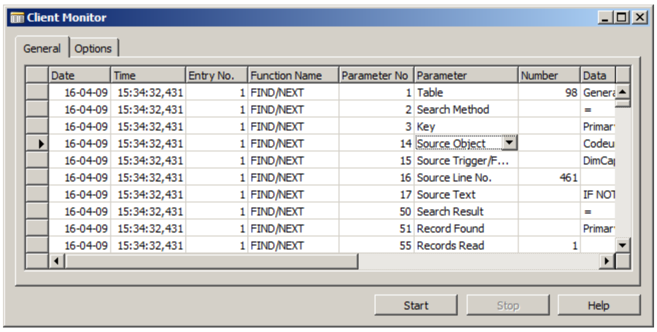
Microsoft Dynamics NAV

○ Ship
○ Invoice
● Ship and Invoice

T..	No.	Description	Location ...	Quantity	Reserve...	Unit of M...	Unit Pric...
I...	1000	Fiets		3		STUKS	4.000,00

Ship-to Addresses (2)
Contacts (1)
Avail. Credit 0
Availability (-58)
Substitutions (0)
Sales Prices (0)
Sales Line Di... (2)

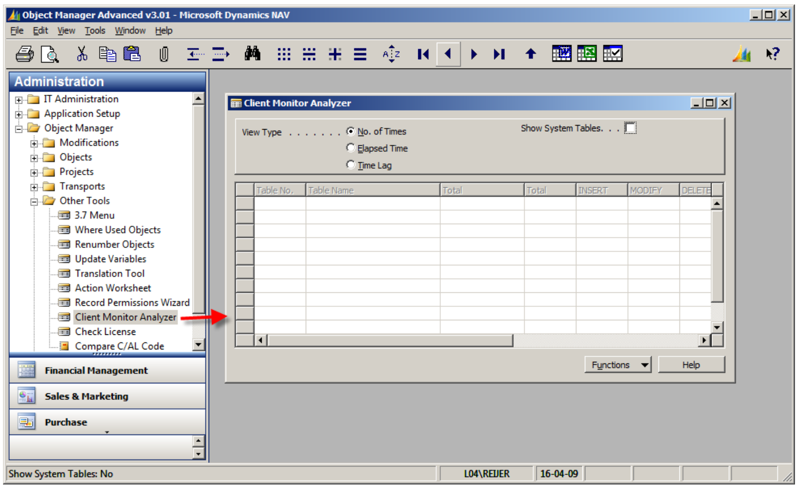
Now stop the client monitor and you get a list with all the data actions.



Date	Time	Entry No.	Function Name	Parameter No	Parameter	Number	Data
16-04-09	15:34:32,431	1	FIND/NEXT	1	Table	98	Genera
16-04-09	15:34:32,431	1	FIND/NEXT	2	Search Method		=
16-04-09	15:34:32,431	1	FIND/NEXT	3	Key		Primar
16-04-09	15:34:32,431	1	FIND/NEXT	14	Source Object		Codeu
16-04-09	15:34:32,431	1	FIND/NEXT	15	Source Trigger/F...		DimCa
16-04-09	15:34:32,431	1	FIND/NEXT	16	Source Line No.	461	
16-04-09	15:34:32,431	1	FIND/NEXT	17	Source Text		IF NOT
16-04-09	15:34:32,431	1	FIND/NEXT	50	Search Result		=
16-04-09	15:34:32,431	1	FIND/NEXT	51	Record Found		Primar
16-04-09	15:34:32,431	1	FIND/NEXT	55	Records Read	1	

Big problem with this form is that it is hard to find which actions are executes many times and which actions did take the most time.

To get a better overview of the client monitor open the Client Monitor Analyzer form.



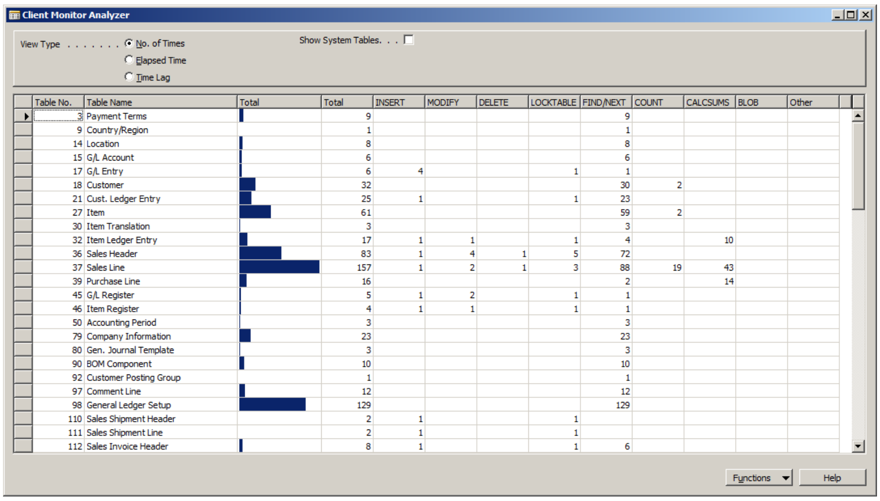
View Type . . . . ● No. of Times ○ Elapsed Time ○ Time Lag

Show System Tables. . . ☐

Table No.	Table Name	Total	Total	INSERT	MODIFY	DELETE

Show System Tables: No | L04\REIJER | 16-04-09

Press F5 and you will see all client monitor lines grouped per table and action type.

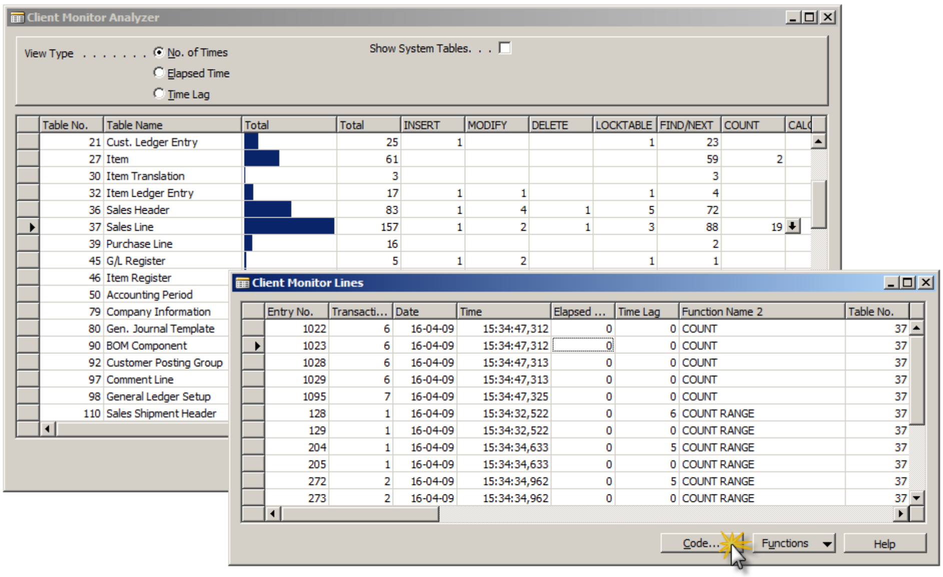


View Type . . . . ● No. of Times ○ Elapsed Time ○ Time Lag

Show System Tables. . . ☐

Table No.	Table Name	Total	Total	INSERT	MODIFY	DELETE	LOCKTABLE	FIND/NEXT	COUNT	CALCSUMS	BLOB	Other
3	Payment Terms		9					9				
9	Country/Region		1					1				
14	Location		8					8				
15	G/L Account		6					6				
17	G/L Entry		6	4			1	1				
18	Customer		32					30	2			
21	Cust. Ledger Entry		25	1			1	23				
27	Item		61					59	2			
30	Item Translation		3					3				
32	Item Ledger Entry		17	1	1		1	4		10		
36	Sales Header		83	1	4	1	5	72				
37	Sales Line		157	1	2	1	3	88	19	43		
39	Purchase Line		16					2		14		
45	G/L Register		5	1	2		1	1				
46	Item Register		4	1	1		1	1				
50	Accounting Period		3					3				
79	Company Information		23					23				
80	Gen. Journal Template		3					3				
90	BOM Component		10					10				
92	Customer Posting Group		1					1				
97	Comment Line		12					12				
98	General Ledger Setup		129					129				
110	Sales Shipment Header		2	1			1					
111	Sales Shipment Line		2	1			1					
112	Sales Invoice Header		8	1			1	6				

Drill down on a particular action to see where the action is executed.



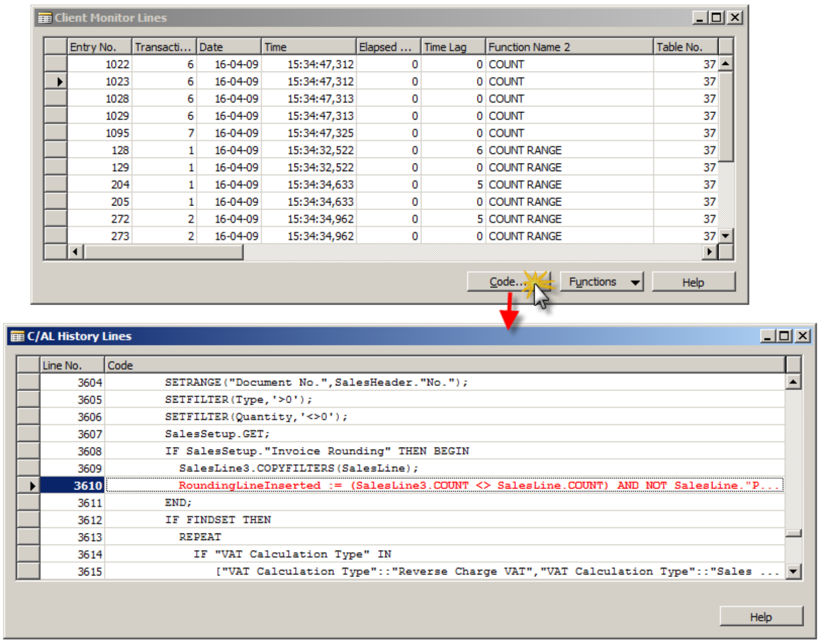
View Type . . . . ● No. of Times ○ Elapsed Time ○ Time Lag

Show System Tables. . . ☐

Table No.	Table Name	Total	Total	INSERT	MODIFY	DELETE	LOCKTABLE	FIND/NEXT	COUNT	CALC
21	Cust. Ledger Entry		25	1			1	23		
27	Item		61					59	2	
30	Item Translation		3					3		
32	Item Ledger Entry		17	1	1		1	4		
36	Sales Header		83	1	4	1	5	72		
37	Sales Line		157	1	2	1	3	88	19	
39	Purchase Line		16					2		
45	G/L Register		5	1	2		1	1		
46	Item Register									
50	Accounting Period									
79	Company Information									
80	Gen. Journal Template									
90	BOM Component									
92	Customer Posting Group									
97	Comment Line									
98	General Ledger Setup									
110	Sales Shipment Header									

Entry No.	Transacti...	Date	Time	Elapsed ...	Time Lag	Function Name 2	Table No.
1022	6	16-04-09	15:34:47,312	0	0	COUNT	37
1023	6	16-04-09	15:34:47,312	0	0	COUNT	37
1028	6	16-04-09	15:34:47,313	0	0	COUNT	37
1029	6	16-04-09	15:34:47,313	0	0	COUNT	37
1095	7	16-04-09	15:34:47,325	0	0	COUNT	37
128	1	16-04-09	15:34:32,522	0	6	COUNT RANGE	37
129	1	16-04-09	15:34:32,522	0	0	COUNT RANGE	37
204	1	16-04-09	15:34:34,633	0	5	COUNT RANGE	37
205	1	16-04-09	15:34:34,633	0	0	COUNT RANGE	37
272	2	16-04-09	15:34:34,962	0	5	COUNT RANGE	37
273	2	16-04-09	15:34:34,962	0	0	COUNT RANGE	37

Press the code button to see which line of code has executed the action.

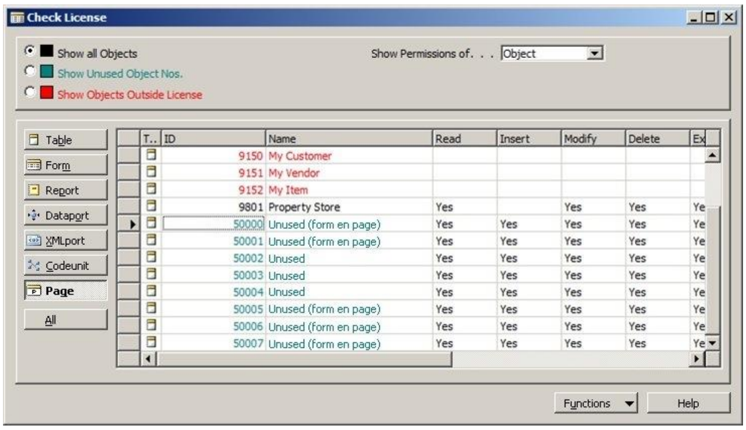


Entry No.	Transacti...	Date	Time	Elapsed ...	Time Lag	Function Name 2	Table No.
1022	6	16-04-09	15:34:47,312	0	0	COUNT	37
1023	6	16-04-09	15:34:47,312	0	0	COUNT	37
1028	6	16-04-09	15:34:47,313	0	0	COUNT	37
1029	6	16-04-09	15:34:47,313	0	0	COUNT	37
1095	7	16-04-09	15:34:47,325	0	0	COUNT	37
128	1	16-04-09	15:34:32,522	0	6	COUNT RANGE	37
129	1	16-04-09	15:34:32,522	0	0	COUNT RANGE	37
204	1	16-04-09	15:34:34,633	0	5	COUNT RANGE	37
205	1	16-04-09	15:34:34,633	0	0	COUNT RANGE	37
272	2	16-04-09	15:34:34,962	0	5	COUNT RANGE	37
273	2	16-04-09	15:34:34,962	0	0	COUNT RANGE	37

```
Line No. Code
    3604         SETRANGE("Document No.",SalesHeader."No.");
    3605         SETFILTER(Type,'>0');
    3606         SETFILTER(Quantity,'<>0');
    3607         SalesSetup.GET;
    3608         IF SalesSetup."Invoice Rounding" THEN BEGIN
    3609           SalesLine3.COPYFILTERS(SalesLine);
    3610           RoundingLineInserted := (SalesLine3.COUNT <> SalesLine.COUNT) AND NOT SalesLine."P...
    3611         END;
    3612         IF FINDSET THEN
    3613           REPEAT
    3614             IF "VAT Calculation Type" IN
    3615                ["VAT Calculation Type"::"Reverse Charge VAT","VAT Calculation Type"::"Sales ...
```

29. Check License

With the Check License form you can see which objects are unused in your license (in other words: available object numbers) and which objects are outside your license.



● Show all Objects
○ Show Unused Object Nos.
○ Show Objects Outside License

Show Permissions of. . . Object

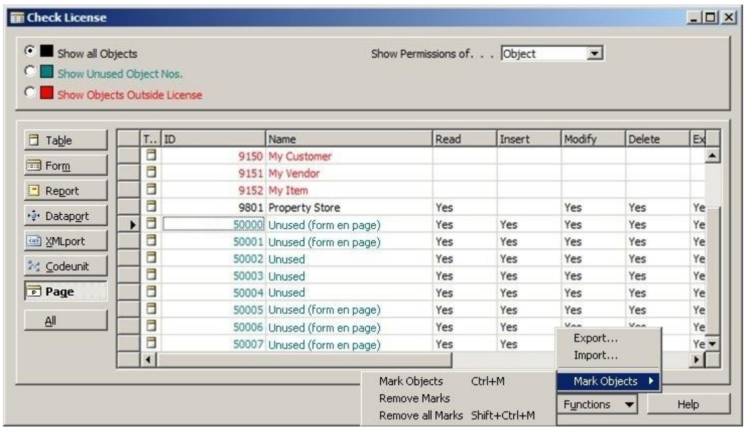
T..	ID	Name	Read	Insert	Modify	Delete	Ex
	9150	My Customer					
	9151	My Vendor					
	9152	My Item					
	9801	Property Store	Yes		Yes	Yes	Ye
	50000	Unused (form en page)	Yes	Yes	Yes	Yes	Ye
	50001	Unused (form en page)	Yes	Yes	Yes	Yes	Ye
	50002	Unused	Yes	Yes	Yes	Yes	Ye
	50003	Unused	Yes	Yes	Yes	Yes	Ye
	50004	Unused	Yes	Yes	Yes	Yes	Ye
	50005	Unused (form en page)	Yes	Yes	Yes	Yes	Ye
	50006	Unused (form en page)	Yes	Yes	Yes	Yes	Ye
	50007	Unused (form en page)	Yes	Yes	Yes	Yes	Ye

29.1. Import and Export License Files

Pressing Functions – Export you can export license information of the active license to a .lic file. At any time you can import a .lic file into the “Check License” window to verify the license against the object set in OMA.

29.2. Mark Objects

To mark objects in the “Check License” window press Functions – Mark – Mark Objects.



● Show all Objects
○ Show Unused Object Nos.
○ Show Objects Outside License

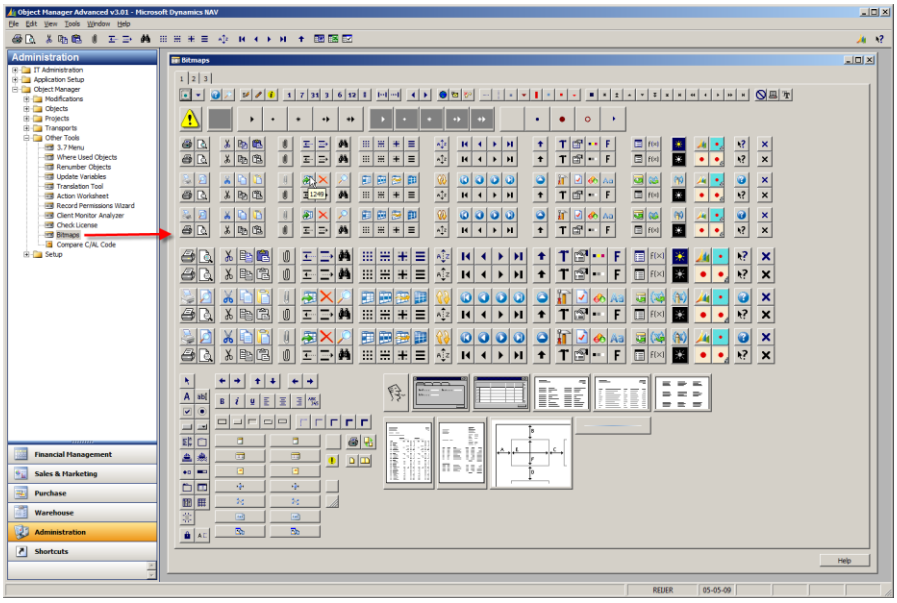
Show Permissions of. . . Object

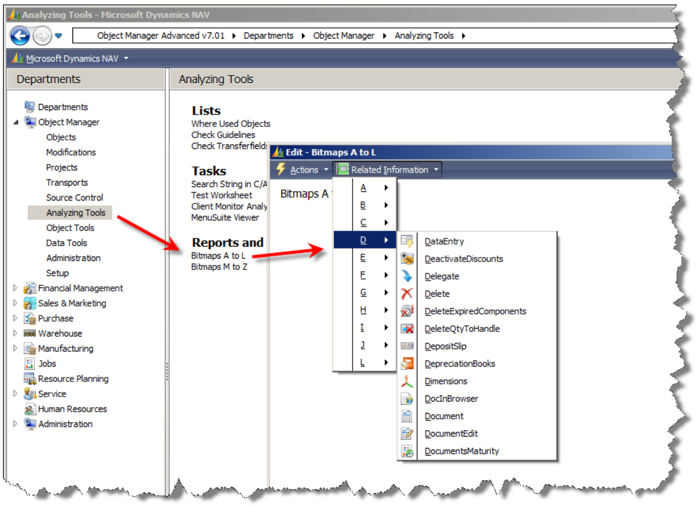
T..	ID	Name	Read	Insert	Modify	Delete	Ex
	9150	My Customer					
	9151	My Vendor					
	9152	My Item					
	9801	Property Store	Yes		Yes	Yes	Ye
	50000	Unused (form en page)	Yes	Yes	Yes	Yes	Ye
	50001	Unused (form en page)	Yes	Yes	Yes	Yes	Ye
	50002	Unused	Yes	Yes	Yes	Yes	Ye
	50003	Unused	Yes	Yes	Yes	Yes	Ye
	50004	Unused	Yes	Yes	Yes	Yes	Ye
	50005	Unused (form en page)	Yes	Yes	Yes	Yes	Ye
	50006	Unused (form en page)	Yes	Yes			Ye
	50007	Unused (form en page)	Yes	Yes			Ye

Objects will be marked by means of a # sign added as a prefix to the Version List value.

30. Bitmaps

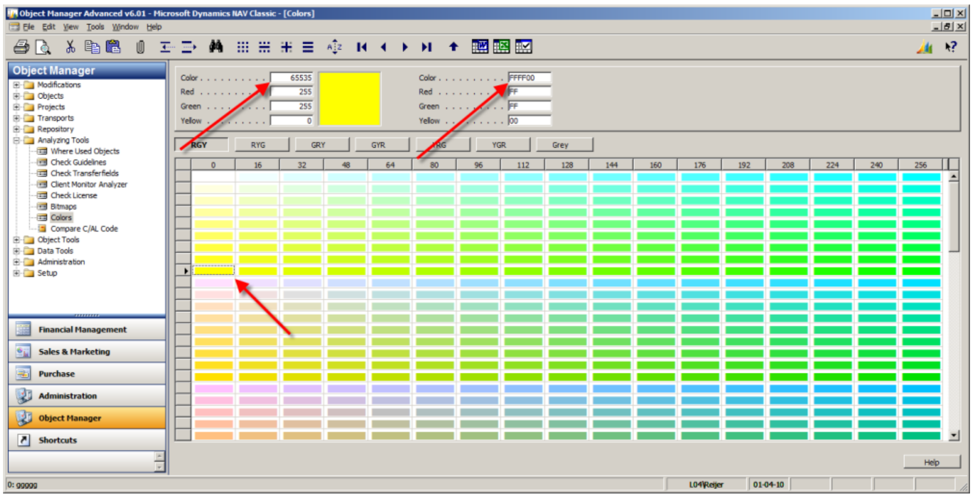
With the Bitmaps form you can see which bitmaps are available in NAV.





31. Colors

With the Colors form you can easily pick a color that can be used in your forms.



Color 65535
Red 255
Green 255
Yellow 0

Color FFFF00
Red . . . . FF
Green . . . . FF
Yellow 00

If you activate a color you see the NAV color number in the left column in the header. The HEX color code will be shown in the right column. Any of the text boxes in the header can be changed. The other values will be automatically calculated.

32. File Functions

Different file functions that can be executed from various windows are to be found here, like:

- Importing files
  see section 7.8 - Import and Export Files
- Comparing files
  see section 32.1 - Compare File
- Comparing directories
  see section 32.2 - Compare Directory
- Splitting files
  see section 7.10 - Split files
- Combining files
  see section 7.11 - Combine Text Files
- Updating object properties in files
  see section 32.3 - Updating Object Properties in Files
- Updating translation, i.e. captions, in files
  see section 32.4 - Translation Files Functions
- Checking guidelines in object file
- Converting Objects
  see section 7.10 - Downgrade Objects

NOTE: All these functions can also be accessed from the "Object Explorer" window through the "Functions" menu button.

32.1. Compare File

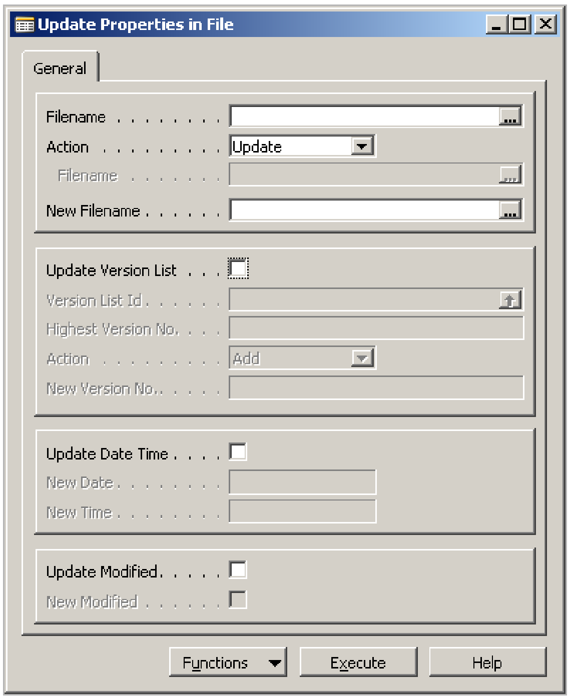
Select File Functions > Compare File to open the windows file dialog to choose an object text file. This file will be compared with the same object residing in your database, using the compare executable as defined in the setup (see section 9.1 Setup).

32.2. Compare Directory

Select File Functions > Compare Directory to open the "Select a Folder" window to choose a windows directory containing object text files. These files will be compared with the same objects residing in your database, using the compare executable as defined in the setup (see section 9.1 Setup).

32.3. Updating Object Properties in Files

Select File Functions > Properties to open the "Update Properties in File" window.



Filename
Action: Update
Filename
New Filename
Update Version List ☐
Version List Id.
Highest Version No.
Action: Add
New Version No.
Update Date Time ☐
New Date
New Time
Update Modified ☐
New Modified ☐

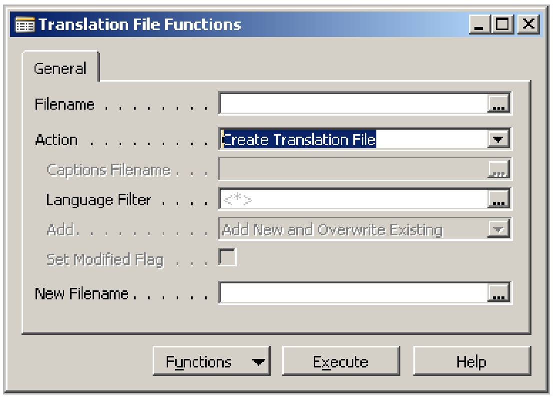
In “Filename” you enter the file for which you want to update the object properties. This can either be a objects text file or a OBP file. The result will be written to the file defined in “New Filename”.
You can update Version List, Date/ Time and Modified object properties by marking the corresponding check boxes and defining how you what to update these properties.

You can perform the following actions:

- Update
  given an object text file all object properties will be updated as described above
- Add
  given an object text file all object properties will be added from the file as defined in “Filename”, which can either be an object text file or an object properties (.obp) file
- Remove
  given an object text file all object properties will be removed
- Create OBP File
  given an object text file all object properties will be extracted and placed into an object properties (.obp) file

32.4. Translation Files Functions

Select File Functions > Translations to open the “Translation File Functions” window.



Translation File Functions

General

Filename . . . .

Action . . . . Create Translation File

Captions Filename . . .

Language Filter . . . . <*>

Add. . . . . Add New and Overwrite Existing

Set Modified Flag . . . ☐

New Filename . . . . . .

In “Filename” you enter the file for which you want to update the translations. The result will be written to the file defined in “New Filename”.

You can perform the following actions:

- Create Translation File
  given an object text file all strings will be extracted and placed into a translation (.txt) file
- Delete Captions
  given an object text file all captions will be deleted from the object txt file
- Add Captions
  given an object text file captions will be added from the file as defined in “Captions Filename”, which can either be an object text file or an translation file

A. Change List

OMA2.17

See the following chapters:

OMA3.01

See the following chapters:

OMA3.01.02

- Bug fix: An Empty TableRelation raised an error in the Function AnalyseTableRelation in Codeunit “OM - Where Used Management”. This is fixed

OMA3.01.03

See the following chapters:

- Shortcuts for design MenuSuite and Pages added to the Object Explorer.

OMA3.01.04

- When you run a Table in the Object Explorer the “OM - Show Table Data” form opens. Check added that a developer license is active.
- Compile (F11) option added to the Object Explorer.
- Functionality added to add date expression like <Month,2> in the ‘filename-fields’ that are available at the Tab “Transport Files” from the setup.
- Bug fix: Currform.SetSelectionFilter didn’t always got the right selection in the Object Explorer. Function SetSelectionFilter2 added to select the selected records.

OMA3.03

- Bug fix: Cope with text constant that have a linefeed in them.

OMA3.04

See the following chapters:

- Deleting objects in Object Explorer is now possible.
- Exporting and importing of BLOB fields added in action management
- Exporting and importing a FAB with different language settings went wrong. Boolean and date fields are now always ex- and imported in a fixed format.
- Trigger FIND added to the where used functionality
- Bug fix: In some clients disappeared all objects from the object designer when filtergroup(1) was used on the object table. Now filtergroup(2) is used instead.
- Bug fix: Where used functionality: Cope with functions without an id.
- Bug fix: Where used functionality: Cope with dataitems in an XmlPort without VarName
- Bug fix: Where used functionality: Relations to tables with the character '/' like "Country/Region" were skipped in updating where used in functionality. This is fixed.
- Bug fix: Where used functionality: Analyzing where used in with Fields with a quote went sometimes wrong. This is fixed.
- Bug fix: Translation Tool: Language Id is used when there is no record in Table 8 – Language.
- Bug fix: Open Table Form: The fieldname of the field "Name" of table Company is now used instead of the string "Name". So in old Dutch databases the string 'Naam' is used.
- Bug fix: Open Table Form: A new form with a shortcut option 'Ctrl' was used. This raised an error when using a German client where "Strg" is used. The shortcut is removed.

OMA3.05

See the following chapters:

- 5 Separated fobs for NAV versions 3.7, 4.0, 4.0 SP2, 5.0 and 2009
- Review of all progressbars.

- All FIND('-') and FIND('+') replaced by 4.0 SP2 functions (FINDSET, FINDFIRST, …)
- All open and save Command Dialog's reviewed.
- All references to standard NAV objects are removed. The Object Manager will now compile without errors in an empty database.
- The BLOB References of the saved compiled objects are moved to a separated table. This to prevent that the blob is unnecessarily fetched from the database.
- Renumber objects/ Renumber fields/Update variables. Before updating the objects you will be confirmed is there are any object locks present on involved objects.
- Bug fix: Renumber objects/ Renumber fields/Update variables. The C/AL of all involved objects will be compared to the last C/AL History Object before the objects will be updated. This to be sure that the Object Manager will not use an old version.
- Bug fix: Renumber fields: Renumbering of fields of type TableFilter added
- Bug fix: Renumber fields: Renumbering of disabled fields added
- Bug fix: Where used functionality: When a function or field is renamed the C/AL of all reference objects will be updated.
- Bug fix: Where used functionality: Error handling added (IF CU.RUN()). When an object has unexpected C/AL which results in an error the object will be skipped.
- Bug fix: Update Variables: Sometimes the end of a function was not recognized so only the first function of a codeunit was updated. This is fixed.
- Bug fix: Edit and run table: If you tried to open a form with a disabled field you got an error. This is fixed. Only enabled fields are added to the form.
- Bug fix: Backup and restore: Field 14 removed from "Action Field" table. This field was disabled and was sometimes crashing NAV without an error.
- Bug fix: Object Compare Sheet: Version List Fields extended to length 80

OMA3.06

- Bug fix - Where used functionality: Relations were not deleted if an object was deleted. Relations will now be deleted.
- Bug fix - Assign modifications: If you had a project with 100 characters you got an error when opening the "Assign Modifications" form. This is fixed.
- Bug fix - Action Management: Action Management is now always executed in ENU Language. Language is set with CurrReport.Language.
- Bug fix - Object Explorer: Caption of objects was not calculated. This is fixed.
- Bug fix - Object Explorer: Design of object was not possible with a German client. A different shortcut key is now used. (the same for run and new)

- Bug fix - Where used functionality: If a long textconstant was present in the global variables sometimes not all variables were read. This is fixed.

OMA3.07

- Send Keys Management: Added shortcuts for “&Find first” for all languages.
- Field “Update Version List” added to the Setup and to the Transport table. When transporting, the version list of the objects are only updated if this Boolean is enabled.
- Bug fix - Action Management: In some clients Booleans were exported as ‘true’ and ‘false’. This is fixed to ‘1’ and ‘0’.
- Bug fix - Where used functionality: SourceTableView property of Form was not good analyzed. This is fixed.
- Bug fix – Translation tool: You got an error when you tried to import a menu item with a name longer than 30 characters. This is fixed.

OMA4.01

- A new trace method: SQL Trigger.
- A new assign method: Active Project.
- Two new fields added to the project card at tab planning. Url and Path
- A new option at the Project Card to transport the project. This makes it easier if you only have to transport 1 project.
- At the transport card the “Version List Number” will be automatic increased if you choose a “Version List Id”.
- It is possible to edit the C/AL code in the code forms. So e.g. when you zoom in from the Where-Used functionality to the code you can slash out the code and save the object.

OMA4.02

- Bug fix – SQL-trigger: Adding the SQL-trigger resulted in an error in some environments. [Id] changed to [ID] and [OM – SETUP] to [OM – Setup]

OMA4.03

- Bug fix – Transporting Files: Better error message when you try to transport an object that cannot compile.
- Bug fix – C/AL History: The C/AL was not written to the history when you assigned a modification to a project. This is fixed.
- Bug fix – Renumber Fields: Renumbering failed when there was a line in an object with more than 1000 characters. This is fixed.
- Bug fix – Renumber Fields: Better error message when you try to renumber a field to a number outside your license.
- Bug fix – Where used functionality: Determining “where used in” with a report with more than 100 data items resulted in an error. This is fixed.
- Bug fix – Edit Table Data: Opening a table with an “-sign in the caption resulted in an error. This is fixed.

OMA4.04

- Bug fix – Where used functionality: Variables with names like ‘next’, ‘find’ etc. were seen as commands instead of variables. This is fixed.
- Bug fix – Edit Table Data: Filters on the form were saved. So if you opened the form with another table you got an error that the filters could not be set. Filters will not be saved anymore.
- Bug fix – Edit Table Data: If you tried to open a table with too many text fields. (like 200 or so) you got an error that the form was too big. This is fixed.
- Bug fix – Edit Table Data: Opening a form with “Trace Method” SQL-trigger resulted in two object changes. This is fixed.

OMA4.05

- Set and remove breakpoints in Where Used results.
- New hidden field in “OM – Setup”: “Keep Objects Maximized”. If this field is enabled then objects keep maximized if you press design in the “Object Explorer”
- Bug fix: All code with datetime fields converted to separated date and time fields. This because of wrong calculation with daylight savings.

OMA6.01

- Commits are now shown in the Client Monitor Analyzer.
- A new field is added to the transport table for assigning a status. This is customer specific functionality so this field is not added to the transport card.
- If you assign a modification to a project you will get a warning if the object is not compiled.
- A new field “Default Role” is added to the User table.
- A new option “Remove Modify Flag at Transport” is added to the setup table. If this field is checked (default) the modify flag

will be removed from all transported objects. This is customer specific functionality so this field is not added to the setup card.

- A new option “Check for Untransported Projects at Importing Transport” is added to the setup table. This is customer specific functionality so this field is not added to the setup card.
- Bug fix – If you transported objects with the SQL-trigger enabled all objects were seen as modifications. This is fixed.
- Bug fix – Objects were sometimes keep popping up in the object compare sheet. This is fixed.
- Bug fix – With some clients the keystrokes were sent to fast. This is fixed.
- Bug fix – Importing of indirect permissions caused an error if an end user did not have the right permissions. These permissions are added to the import management codeunit. If importing of the objects still not work a message will appear that you will have to import a fob.
- Bug fix – If a key was deleted or inserted all the other keys were pointing to the wrong where used in base. This is fixed
- Bug fix – Sometimes the "Run Table" and "Edit Table Data" functions failed because the existing form could not be exported. This is fixed
- Bug fix – Where-Used Functionality: When there was C/AL code in a report DataItem the SourceExpr were not recognized
- Bug fix – Where-Used Functionality: 'EVALUATE(Field, '1');' was not seen as a fill of the field. This is fixed
- Bug fix – Where-Used Functionality: 'Field += 1;' was not seen as a fill of the field. This is fixed
- Bug fix – Check Guidelines: Sometimes the ELSE of an IF was seen as the ELSE of a case statement. This is fixed
- Bug fix – The check if the SQL-trigger is added was executed with the "OM - Setup" table even if you did not had the right to modify it. Now an object is used that can be changed.
- Bug fix – Translation Tool: Rules on translations of textconstants were not always recognized. This is fixed
- Bug fix – Translation Tool: If a rule had a translation that was longer than 50 characters the text was saved wrong. This is fixed.

OMA6.02

- New functionality: “Group Modifications Period (sec.)”
- New functionality: Marking functions extended.
- Bug fix – Assigning modification: When an object was assigned in popup mode sometimes the second time resulted in an error. This is fixed.
- Bug fix – Send Keys: Pause between keystrokes raised to 100 ms.

OMA6.03

- Bug fix in the “Object Manager Setup”.
- New functionality: Field id added to the translation tool.
- New functionality: When changing the project filter in the “Object Explorer” and you have trace mode “Active Project” then the active project will be changed.
- New functionality: Renumber objects can be ran over a text file so updating the where used object is not necessary anymore.
- New functionality: In the renumber objects form you can fetch all the objects of a project, a transport or a text file.
- New functionality: Renumber fields can be ran over a text file so updating the where used object is not necessary anymore.
- New functionality: Update variables can be ran over a text file so updating the where used object is not necessary anymore.
- Bug fix – C/AL History: It was not possible to show the C/AL code of an object that was deleted from the database. This is fixed.

OMA7.01

- Pages
- Integration with NAV2009 R2 locking functionality
- Trace modifications with “Integration Management”
- New option in setup added to enable execution of actions with indirect permissions
- New option in setup added to compile objects after transport is imported
- New option in setup for keep forms maximized when designing or running objects
- New setup initialization type “Pre-Prod”
- New functionality: Variables added in Where Used
- New functionality: Virtual tables added in Where Used
  New functionality: Transferfields added in Where Used
- New functionality: “Record Links” of projects and transport are transported
- New functionality: Possibility to e-mail to multiple roles when changing project status
- New functionality: Server and database name in project status change email
- Warning added when you import an old transport
- New functionality: Option to block/unblock projects and transports at importing transport
- New functionality: Structure of Translation Tool changed so that you do not have to wait when changing between languages
- New functionality: Date Filter in Compress C/AL History
- New Pick Folder form
- New functionality: Edit objects in external editor
- You can set trace mode in assign modifications forms
- Possibility to enter T for Today (h for Heute) in the date filter in the Object Explorer
- Fields added to “Lookup Version List” form to see the “No. of Objects” and “No. of Version List Nos.” per “Version List Id”

- New option in Object Explorer (Ctrl + Shift + F5) to check the C/AL History.
- Field “Restored from Entry No.” added to “C/AL History Objects”.
- Bug fix – Where Used: Primary Key added to Secondary Keys where needed
- Bug fix – Where Used: Smarter context mechanism for recognizing statements with conflicting names

OMA7.02

- New functionality: New field in setup “C/AL Editor Executable”. This executable is used when opening. If empty the default text editor is used.
- New functionality: New field in setup “Server Name”. This can be used if the server name cannot be determined with the server table.
- New functionality: Possibility to go to C/AL code from Check Transferfield form.
- New functionality: Menu options to calculate variable usage from “Where Used In” and “Where Used Object List” forms.
- New functionality: Mark objects from “Search String in C/AL Code” form.
- New functionality: CONFIRM instead of ERROR when Version List exceeds 80 characters when transporting objects.
- Bug fixes
    - At importing and modifying objects in text format the regional data format is used instead of the date format from the setup card.
    - Objects were not locked when designing from object explorer.
    - Where Used Functionality – When an object with a quote was used in a report an error occurred.
    - Check Guidelines – The captions in the Action List of pages were not checked.
    - Objects without a date were popping up in Object Compare Sheet every time.
    - Renumber Fields – Objects were compiled halfway. Therefore not all references were updated.
    - TFS – Batch files were deleted while they were locked. This resulted in an error on slow repositories.
    - Create Table Wizard – A description field of 100 was created. A Search Name with length 30. This could result is an error.

OMA8.01

- New Functionality: Possibility to add projects to transport in planning phase
- New Functionality: Improved "Comment Sheet"
- New Functionality: Improved Objects Import Worksheet
- New Functionality: Mailing with SMTP or Outlook

- New Functionality: Possibility to save to C/AL History without “C/AL Code”
- New Functionality: Possibility to disable Integration Management for importing a conflicting fob
- New Functionality: Check Guidelines on project card
- New Functionality: Check Guidelines when project goes to status ready
- New Functionality: Check Guidelines at transporting
- New Functionality: New action type Execute DOS Command
- New Functionality: Reset Transport Status at Import Transport
- New Functionality: A Boolean on the Project Object table to choose if an object must be transported
- New Functionality: Parameters can be used in “Compare Executable” and “C/AL Editor Executable” setup fields
- New Guidelines Check: Check if TextConstant in used in a dialog
- New Guidelines Check: Check if project tag is present in documentation trigger
- New Check Guidelines: Check Wrong SETCURRENTKEY
- New Check Guidelines: Check Usage Wrong Keys
- Check Guidelines: Available access keys are shown when a conflict is found
- Set date time to object files when exporting objects or splitting object text file
- Possibility to exclude FlowFields in actions
- Possibility to select "<Active company>" and "<Same company>" in action worksheet
- Where Used Functionality: See which objects used in an object (on relations tab)
- Possibility to export objects from "Where Used String" form
- Possibility to mark object from "Object List"
- C/AL Editor is used when editing object in "C/AL History Lines" and "Check Guidelines Code"
- Better collapse functions in "MenuSuite Viewer"
- Possibility to choose between "Name" and "Description" field in "Create Table Wizard"
- Better report to create actions in "Action Worksheet"
  If GUIALLOWED added to dialogs to add "Update C/AL History” and “Update Where Used” to NAS (client must be opened)
- Bug fix: Object unlocking - At integration management the 'LOCKED' tag was not removed if it was set by SQL trigger.
- Bug fix: Renumber Fields - Only the original fields of the last table were removed.
- Bug fix: Renumber Objects - When copying to action worksheet object type table was used.
- Bug fix: Where Used - Identifier that Variable Usage was calculated was not always set to false.
- Bug fix: Determining Highest Version Nos.: OMA7.01.01 was higher than OMA7.02.
- Bug fix: Determining Version Nos.: If version list id 'OM' was present, OMA2.01 was also seen as OM with no. A20.1. This is fixed

- Bug fix: Translation Tool – When a caption was changed it was not exported if another language was active. This is fixed
- Bug fix: Check License – The form was sometimes getting in a loop when showing unused object nos. This is fixed
- Bug fix: Send Keys Management – Objects were not opened when German Language was active. This is fixed
- Bug fix: Group Modifications – An overflow error was given when the previous modification was more than 24 days ago. This is fixed
- Bug fix: Renumber Elements – Controls of RequestForm and RequestPage were not seen. This is fixed

OMA8.02

- If the exact same project tag is already present in an object it is not added for a second time.
- Where Used Management: Calculating which objects has to be updated is made faster.
- Bug fix: Calculation the highest transport number was not done right. This is fixed.
- Bug fix: The “Assign modifications” form was not pointing to the right project when opened from the project card. This is fixed.
- Bug fix: Calculating the initials of the active user was done with the field Code instead of the field “User ID”. This is fixed.
- Bug fix: Modify permission were needed to export an object. This is changed to read permissions.
- Bug fix: Check Guidelines. The check if a project tag is present in an object looked at the whole objects instead of the documentation trigger.
- Bug fix. Initializing the database with the option "Block Design with Ctrl+F2" was giving an error.

OMA8.03

- When entering active project in the “Assign Modifications” form also the project to assign will be selected.
- Pressing <enter> in the search box on the “Search String in C/AL Code” form will trigger the search.
- Bug fix: Domain will be removed when “User ID” is validated in “OM - User” table.
- Bug fix: In the compare database orphans of the right database were not shown. This is fixed.
- Bug fix: It was not possible to add a project tag to a report with RDLDATA. This is fixed.
- Bug fix: Menusuites are ignored if you try to add a project tag to a menusuite.
- Bug fix: Where Used - Sometimes a statement like database::”Sales Header” was shown in the run trigger. This is fixed.
- New functionality: Where Used - Fields used in keys added.
- New field in repository setup table to raise the timeout value for TFS

- Bug fix: TFS source control. Illegal temp files were created with English local settings. This is fixed.

OMA8.54

- The exit value of a function added as entity to the Where Used functionality
- In the "Where Used In" list you can see which lines are temporary records.
- It is possible to enable and disable field and keys in the Where Used Card.
- An SQL trigger to check if the where used objects are up to date (100 times faster)
- Setup: veld "Update Where Used" an option added to confirm, always or never update the where used tables
- New guidelines check
- Check same field types in table relation and FlowFields.
- Show if a control name is not used.
- Show if a code line does not have to be broken (i.gesplitst over 2 regels)
- Show if a CALCDATE is used without '<>'.
- Show where the MARK command is used.
- Check if a space is used in a function name.
- Check if an integer is used in commands like FORM.RUN(3)/REPORT.RUN(3)/...
- Error handling in check guidelines so the OMA continues with the next object if an error occurs.
- Captions can easily be changed in the check guidelines tool.
- Known translations can be auto applied if they are present as a rule in the translation tool.
- Possibility added to specify critical checks.
- New option in Project and Transport Flow to block the project/transport.
- Set known comments in project that were already present when the project started.
- Possibility to import a transport without the wizard.
- Edit and add project and transport comments with you text editor.
- Show if an object is in an active project in Compare Database Sheet.
- Possibility to rollback a project or transport.
- Show free numbers that are both available as form type as page type.
- Ex- and import license definition Check License form.
- Mark objects functions added.
- Possibility to choose another base language than ENU in the translation tool.
- Direct interaction between Database and Translation Tool (without translation file).
- New file format *.obp: Object Properties.
- Extracting "Object Property File" and "Object Translation File" from "Text Files".
- Extracting "Object Property File"

- Extracting "Object Translation File"
- New form to modify translations in a text file and/or translation file.
- New form to modify properties in a text file and/or OBP file.
- Possibility to split a FIB file.
- Possibility to split and combine a translation file.
- Existing functionality:
- * Split Text Files (7.4)
- * Combine Text Files (7.5)
- * Compare Directory (COD11102094 - OM - Compare Directory)
- * Compare File (COD11102093 - OM - Compare File)
- * Import File (COD11102043 - OM - Im Export Management)
- * Convert (Downgrade) Objects
- An option that pages has to be included in the Record Permission Wizard.
- Lookup on table relations in the Edit Table Data form.
- It is now possible to add the SQL triggers to a database without windows logon.
- Setting to lock the database. No object modifications will be possible.
- Fob, fib, obj, txt and obp-files can be compared in the Compare Database Sheet.
- Administration task added to remove the OMA from a database.
- The UID Offset can be set on a project type and/or in the setup.
- When an transport is imported all object modifications will be assigned to the included project.
- Edit comments in projects and transports with an external editor.
- After an object restore the object is compiled to rebuild the Meta Data.Field "Remove Modify Flag at Transp." added to transport type card.
- Recognizing if partner or customer license present when upgrading from trial to full version.
- Administration Task added to empty the where used tables.
- Administration Task added to empty deleted fields and tables.
- Administration Task added to upgrade to the full version.
- A hidden column "Project" added to the modifications and project history form.
- A more detailed Project button added to the Assign Modification form.
- Shortcut Ctrl+P added for opening the documents.
- When "Show Last Changes" you are getting a question is you want to update the C/AL History.
- Column for users added to the Planning Board.
- Progress bar added when assigning modifications.
- An option "Confirm Yes" added to the "Remove Locks at Closing Menu".
- On the Modifications form a menu option added to go to the history of an object.
- Marking of objects added to the "Check Guidelines Comments" form.
- Fields added to the project subform on the assign modifications form.
- A new option on the Project Card to activate a project.

- Record Permissions: An option that pages has to be included in the role.
- Renumber elements: When there is nothing to renumber the object does no longer getting a new timestamp.
- Updating Version List: When you have two times the same "Version List Id" one is deleted.
- When you renumber an object data and conflicting data is present you will be warned and the data will be deleted.
- Where Used: If an object is used in a menusuite then show it in the run trigger.
- Possibility to import a transport without the wizard. This can be done with codeunit "OM - Import Transport".
- Show if an objects is in an active project in the "Compare Database" form.
- Codeunit 1: Open Menu can be run with a new Codeunit "OM - Open Menu".
- When you import an object text file first all objects are imported and then compiled.
- Create Table Wizard: An option to add a Drop Down field group.
- Create Table Wizard: Spaces are placed in code only if space is mandatory in the check guidelines setup.
- Button added to the database card to test the connection.
- Bug fixes
    - Create Table Wizard: It was possible to selection a search name field without a name field.
    - Sometimes the OMA could not find the right user is in the OM - User table. This is fixed.
    - Compare Databases: It was not possible to connect to an SQL database with SQL Server Authentication.
    - Renumber Objects: Sometimes if the sorting was on "New Number" records were skipped.
    - Projects and Transports: Project Type was not validated if you choose project type with assist edit on the no. field.
    - Projects and Transports: The field "Update Version List" was not copied from the "Transport Type".
    - Projects and Transports: Subform was disabled is there were no objects is the project.
    - Documentation Tag: When the trace method Timer was used and a tag was added to the documentation trigger a new modification was added.
    - SQL Trigger: When a fob was imported and an object was locked outside you license it was not possible to remove the LOCKED tag from the version list.
    - Colors: RGY was used instead of RGB.
    - Check Guidelines: If a language was not present in the Language table it was skipped.
    - Check Guidelines: Analyzing a SETCURRENTKEY was giving an error sometimes.
    - OBJ file: When an object had a time in another format as your local settings you were getting an error.
    - Check License: When a filter on ID was on the form the shown recordset was not right.

  - Where Used: When a dataitem with a quote was used not all references were found.
  - Where Used: A filter in a tableview on a form without a selected key was not indexed.
  - You were getting an error when you were opening an object in an external editor and the option "Lock Object at Design" was enabled.
  - Renumber Elements: An error was given if the ReturnValue of a function had a conflicting id.
  - Create Table Wizard: It was not possible to enter a name in the nos. series table.
  - If an object was included in the renumber objects and was not renumbered the where used tables of the object were deleted.
  - When you deleted 10 lines in the c/al code form and the object was outdated you were getting 10 warnings.
  - Projects and transport: Calculation of description went wrong if %1<Year,4> was used.
  - Adding project tag: Calculation of tag went wrong if %1<Year,4> was used.
  - Updating objects failed when an object was locked in the Object Designer but not in the Object Manager.
  - Downgrading objects went wrong is GETLASTERRORTEXT() was present in an object. Now the () are deleted.
  - Grouping modifications was going wrong if the prevision modification was more than 24 days newer.
  - Editable table data and run table failed if the local time format was not readable for NAV.

OMA9.01

- Lock Marking: you can lock the marking functionality to prevent that someone else will mark and unmark objects.
- New trigger can be added in the on SQL level to the prevent a database conversion.
- Database compare sheet can now be used with databases that does not have the OMA installed.
- Better number series assignment. Now the last part of a project and transport number will be incremented instead of the first number part.
- New options "Update Referencing Objects" and “Set Modify Flag” added to the object update tools.
- Range fields (From/Till) in "Update Variables Tool" changed to filter fields.
- New option "Left Side Base Line" in report "Compare C/AL Code".
- Better file names are used when comparing objects.
- Comparing a directory now uses the file names used in the directory instead of the Object Manager format.
- A warning is given when downgrading objects and a SourceTableTemporary property is removed.
- A fob is included in the setup that can send mail with the NAV Mail Component 7.3

- Objects can be converted from Microsoft Dynamics NAV 2013 to previous versions.
- Objects belonging to a project can be exported in their current state, before the project or after the project.
- Status can be changed for a number of projects from the project list and plan board.
- OBJ files can be split and combined.
- Compare Fields
    - Field Compare Sheet. You can analyze the differences between two databases in a worksheet.
    - New file type called "Fields File" which contains the definition of table fields. This file type can be used in the following functionality: Import File, Import Directory, Compare File, Compare Directory, Split, Combine and Compare Fields, Mark Objects with Import File, Lock Objects with Import File.
- Where Used
    - If an object is used in the report selection or as a web service the reference is show in the run trigger of the object. Will be refreshed when an object changes or at Ctrl+F5.
    - A sub type is shown in the where used object lines. Is shows the object number of a variable.
    - A new column function name is added to the where used object lines.
    - The function INIT on tables is added as trigger in the where used object line.
    - When deleting a field in the where used only the filled fields are emptied instead of all fields.
    - When the C/AL history objects is not present the C/AL entry number is now automatically repaired.
- Check Guidelines
    - Filters added to the header of the form so you can add e.g. the objects of a particular project or transport.
    - Fields "Object Date" and "Object Time" added so you can easily filter for recent objects.
    - "Check Field Names" check extended. It now checks that a field and the related table cannot have the same name. E.g. the field name Customer must be "Customer No.".
    - New Guideline Checks
        - Check if a table has fields with the same caption.
        - Check if a CalcFormula is present
        - Check if a table relation is present on fields where in the code a primary key is assigned. e.g. "Customer No." := Customer."No.".
        - Check if a bigger field is assigned to a smaller field. 'This assignment can cause an overflow. Code10 := Code20'
        - Check if all relating tables are deleted in the OnDelete trigger of the main table.
        - Check if a string is used in a calc formula or a table relation.

        - Check that no code is written in the OnLookup trigger of a field. Must be transferred to function and called from a form.
        - Show where commenting with '{' and '}' is used.
        - Check that fields matches when a TransferFields command is found. E.g. "Transferfields on field Customer No. can cause an error. Code20 := Code40"
- Translation Tool
    - Two new fields added to the translation tool: "Translate Later" and Comment.
    - If a translation is not found with the same type, all other types are checked.
    - Showing existing translation is now grouped and shown with the number of times used.
    - A warning is given when you are trying to import or export more than 100 objects in the translation tool. It's better to work with a translation file when working with a lot of objects.
    - Bug fix: The field "Calculated Caption Differs" was not always updated when a new caption was validated
- Create Table Wizard
    - Entity types added to the Create Table Wizard: Master Data, Document (Header/Lines), Setup, Ledger Entry.
    - Captions and names can now be added/changed before the objects are created.
- Projects and Transports
    - A hidden column is added to the project subform where you can see the last user that has changed the object.
    - Export projects and transports as OBJ, FAB or Text file.
    - The transport import wizard now starts right after the objects are confirmed in the compare sheet.
- Check Transfer Fields
    - A new field "Known" added to filter out already known warnings.
    - Field class added to see if a field is a flow field.
- Action Worksheet
    - New action type "Add Record" added to the action worksheet.
    - When you validate a table number in the action worksheet the existing field mapping is maintained.
- Objects to Ignore
    - The name "Known Object Export Error" changed to "Object to Ignore".
    - MenuSuites added to "Object to Ignore" list.
    - "Objects to Ignore" are now skipped when adding objects to the check guidelines tool.
- Bug Fixes
    - When you reset a project status not existing users will be emptied from the project card.

  - Removing a tag from the version list was ignoring special characters.
  - The client closed when Quick Find was set to no. This is fixed.
  - The progress bar will now show 0 seconds if it accidently passes 100%.
  - Renumbering object in export mode was not working.
  - It was not possible to open the edit table sheet with a RecordId field.
  - Menusuites containing empty guides were sometimes shown wrong.
  - Integration Management. A object deletion was not added to the modifications table.
  - The hash character was removed from the version list when deleting a variable in the where used functionality.
  - Bug fix: rollback objects were picked incorrectly from history when rolling back with a time value

OMA9.02

- Bug fix - Translation Tool: Name lengths are extended to 128, 132 and 256 in NAV2013 and R2.
- Bug fix - Where Used: Updating sometimes stopped when multiple users were updating at the same time.
- Bug fix - Split Text Files: You could not split a file with unknown object types. E.g. a text file with queries in NAV2009.
- Bug fix - Change Field Options: Array length extended.
- Bug fix - Backup and Restore: Restoring of large backup files 2GB+ was giving an overflow error.
- Bug fix: Updating the version list was updating the wrong part if the new id was part of an existing larger id.

OMA9.03

- A new codeunit “OM - Check New Transport” added to check for a new transport.
- New option in transport new timestamp: Date of Transport.
- New Timestamp added to transport type.
- Assign Modifications page/form is default filtered on active user
- You get a proper message if you are importing a transport with a renumber field action and you don’t have an active license that allows importing a text file.
- Marking added to all object tools.
- 2013: You get a proper message if marking of objects is not possible if the version list is full.
- 2013: Extra test added to the “Check Settings” to see if dotnet is ok.
- Bug fix: Removed quotes from field names in setup table.
- Bug fix: Check guidelines. If a semicolon was missing on the last line it was not seen as missing.

- Bug fix: Check guidelines. If REPEAT BEGIN where on the same line the BEGIN and END where sometimes removed wrongfully.
- Bug fix: Where used. In a statement like FieldRef.SETRANGE(VarName) the VarName was seen as a fieldname instead of a variable.
- Bug fix: If you removed a project from a transport in NAV2013 it still was connected to the transport.
- Bug fix: Paths in setup extended to length 250.
- Bug fix NAV2013 - Get a user name with a domain.
- Bug fix NAV2013 - Marking an object is possible If the version list is shorter than 248 instead of 80.
- Bug fix NAV2013: Where used - Properties on Actions on a CueGroup where not parsed.
- Bug fix NAV2013: Where used - Table data was not analyzed if the object was not present in the object table as table data. Now info from table information is used.
- Bug fix NAV2013: Where used - On overflow was given is a long key was shown on the card page.
- Bug fix NAV2013: Command Line - Timeout of 5 seconds removed from the import object command.
- Bug fix NAV2013: Command Line - A longer pause is added to wait until the log file free to open.
- Bug fix NAV2013: Command Line - Parameter validatetablechanges added to the command line.
- Bug fix NAV2013: Wrong page was used if you did a lookup to a language.
- Bug fix NAV2013: Check Guidelines - If an action was missing a caption is was not seen as missing.
- Bug fix NAV2013: Trace modifications - In some installations it was not possible to modify an object with the trace trigger. This is fixed.

B. Developer vs. Customer License

In the following table you can see which functionality is available in combination with a developer license and which functionality is available with a customer license. Some parts will only be available with a customer license when the C/AL history is updated with a developer license.

	Dev. License	Cust. License	C/AL Update Needed
Objects			
Object Explorer	X		
Import Objects	X	X	
Object Locks	X		
Modifications			
Start Tracing	X		
Modifications	X		
Assign Modifications	X		
Projects			
Projects	X	X	
Planning Board	X	X	
Project History	X	X	
Transports			
Transports	X	X	
Object Compare Sheet	X	X	
Import Transport	X	X	
Source Control			
C/AL History	X	X	X
Branches	X		
Rollback Objects	X		
Repository	X		
Compare C/AL Code	X	X	X
Analyzing Tools			
Where Used Objects	X		X
Compare Databases	X	X	X
Search String in C/AL Code	X	X	X
Check Guidelines	X		X
Test Worksheet	X	X	
Check Transferfields	X	X	
Client Monitor Analyzer	X	X	
MenuSuite Viewer	X	X	X
Check License	X	X	
Bitmaps	X	X	
Colors	X	X	

Object Tools

Create Table Wizard	X		
Renumber Objects	X		
Renumber Fields	X		
Renumber Elements	X		
Update Variables	X		
Change Field Options	X		
Translation Tool	X		
Downgrade Objects	X		

Data Tools

Action Worksheet	X	X	
Record Permissions Wizard	X	X	

Administration

Compress C/AL History	X		
Backup and Restore	X		

C. Differences between Fobs

	3.7	4.0	4.0 SP2	5.0	2009	2009 R2
Tables	x	x	x	x	x	x
Forms	x	x	x	x	x	x
Reports	x	x	x	x	x	x
Dataports	x	x	x	x	x	x
XMLPorts		x	x	x	x	x
Codeunits	x	x	x	x	x	x
Menusuites		x	x	x	x	x
Pages					x	x
FINDSET			x	x	x	x
Integration Mgt.						x
NAV Locking						x

D. Setup Initialization Methods

	Development	Test	Pre-Production	Production
Lock Object at Design	TRUE	FALSE	FALSE	FALSE
Remove Locks at Closing Menu	Confirm	No	No	No
Save C/AL at Modification	If Changed	No	No	No
Save C/AL at Assigning	If Changed	No	No	No
Save C/AL at Locking	If Changed	No	No	No
Save C/AL at Unlocking	If Changed	No	No	No
Save C/AL after Transporting	Yes	No	No	No
Save C/AL before Import Tr.	If Changed	Yes	Yes	Yes
Save C/AL after Import Tr.	If Changed	Yes	Yes	Yes
Reset Project Status at Import	FALSE	TRUE	FALSE	FALSE
Block Project at Imp. Trans.	<empty>	No	Yes	Yes
Block Transport at Imp. Trans.	<empty>	Yes	No	Yes
Transport Nos. Format	T0001	TT0001	T0001	T0001

E. Object Table SQL Trigger

```
IF EXISTS
(
  SELECT name FROM sysobjects
  WHERE name = 'Object_TraceModifications' AND type = 'TR'
)
DROP TRIGGER Object_TraceModifications

GO

CREATE TRIGGER [dbo].[Object_TraceModifications]
   ON  [dbo].[Object]
   AFTER INSERT,DELETE,UPDATE
AS
BEGIN

  SET NOCOUNT ON

  DECLARE @SetupPresent INTEGER;
  DECLARE @SetupTriggerTestStatus INTEGER;
  DECLARE @SetupSQLCheckObjectLockType INTEGER;
  DECLARE @SetupTraceModifications INTEGER;
  DECLARE @SkipSQLTrigger INTEGER;
  DECLARE @Username NVARCHAR(100);
  DECLARE @LockedBy NVARCHAR(100);
  DECLARE @IsLocked INTEGER;
  DECLARE @IsModification INTEGER;
  DECLARE @CheckIsLocked INTEGER;
  DECLARE @CalledFromRepository INTEGER;
  DECLARE @ObjectType INTEGER;
  DECLARE @ObjectId INTEGER;
  DECLARE @ObjectName NVARCHAR(100);
  DECLARE @ObjectDate DATETIME
  DECLARE @ObjectTime DATETIME
  DECLARE @ObjectTypeText NVARCHAR(100);
  DECLARE @ObjectIdText NVARCHAR(100);
  DECLARE @Message NVARCHAR(100);
  DECLARE @NoOfInserts INTEGER;
  DECLARE @NoOfDeletes INTEGER;
  DECLARE @NoOfModifies INTEGER;
  DECLARE @Action INTEGER;
  DECLARE @TokenNo INTEGER;
  DECLARE @LockObjectAtSaving INTEGER;
  DECLARE @OldVersionList NVARCHAR(80);
  DECLARE @NewVersionList NVARCHAR(100);

  SELECT TOP 1
    @SetupPresent = 1,
    @SetupTriggerTestStatus = [Trigger Test Status],
    @SetupSQLCheckObjectLockType = [SQL Check Object Lock Type],
    @SetupTraceModifications = [Trace Modifications],
    @SkipSQLTrigger = [Skip SQL Trigger],
    @LockObjectAtSaving = [Lock Object at Saving]
  FROM [OM - Setup]
  WHERE [Primary Key] = '';

  IF @SetupPresent = 1
  BEGIN

    IF @SetupTriggerTestStatus = 1
    BEGIN
      UPDATE [OM - Setup]
      SET [Trigger Test Status] = 3;
    END ELSE BEGIN

      IF @SkipSQLTrigger = 0
      BEGIN
        SELECT
          @ObjectType = d.[Type],
          @ObjectId = d.[ID],
          @ObjectName = d.[Name],
          @ObjectDate = d.[Date],
          @ObjectTime = d.[Time]
        FROM Deleted d;

        SELECT
          @ObjectType = i.[Type],
          @ObjectId = i.[ID],
          @ObjectName = i.[Name],
          @ObjectDate = i.[Date],
          @ObjectTime = i.[Time]
        FROM Inserted i;

        SET @CalledFromRepository = 0;
        SELECT @CalledFromRepository = 1
        FROM [OM - Repository Log] rl
        WHERE rl.[Object Type] = @ObjectType
        AND rl.[Object No_] = @ObjectId
AND rl.Status = 1;

IF @ObjectType > 0 AND (@ObjectId < 1000000000 OR @ObjectId >= 2000000000) AND @CalledFromRepository =
0
BEGIN
  SET @LockedBy = '';
  SET @TokenNo = 0;
  SET @IsLocked = 0;

  SELECT
    @LockedBy = ol.[Locked By],
    @LockedBy = ol.[Locked By],
    @TokenNo = ol.[Token No_],
    @IsLocked = 1
  FROM [OM - Object Lock] ol
  WHERE ol.[Object Type] = @ObjectType
  AND ol.[Object No_] = @ObjectId;

  SET @Username = UPPER(SYSTEM_USER);
  IF (CHARINDEX('\', @Username) > 0)
    SET @Username = SUBSTRING(@Username, CHARINDEX('\', @Username) + 1, 100);

  SELECT @NoOfInserts = COUNT([Type]) FROM Inserted;
  SELECT @NoOfDeletes = COUNT([Type]) FROM Deleted;
  SELECT @NoOfModifies = COUNT(i.[Type])
  FROM Inserted i
    INNER JOIN Deleted d
      ON i.[Type] = d.[Type] AND i.[ID] = d.[ID];

  SET @IsModification = 0;
  SET @CheckIsLocked = 0;

  -- INSERT
  IF @NoOfModifies = 0 AND @NoOfInserts > 0
  BEGIN
    SET @Action = 1;
    SET @IsModification = 1;
  END;

  -- MODIFY
  IF @NoOfModifies > 0
    SET @Action = 2;

  -- DELETE
  IF @NoOfDeletes > 0 AND @NoOfInserts = 0
  BEGIN
    SET @Action = 3;
    SET @IsModification = 1;
    SET @CheckIsLocked = 1;
  END;

  -- RENAME
  IF @NoOfModifies = 0 AND @NoOfDeletes > 0 AND @NoOfInserts > 0
  BEGIN
    SET @Action = 4
    SET @IsModification = 1;
    SELECT @Action = 0, @IsModification = 0
    FROM Inserted i, [OM - Update Object] uo
    WHERE i.[Type] = uo.[Object Type]
    AND i.[ID] = uo.[New Object No_];
  END;

  -- MODIFY
  IF @Action = 2
    SELECT
      @IsModification = 1,
      @CheckIsLocked = 1
    FROM Inserted i
      INNER JOIN Deleted d
        ON i.[Type] = d.[Type]
        AND i.[ID] = d.[ID]
    WHERE
    (
      i.[Date] <> d.[Date]
      OR CONVERT(VARCHAR(20), i.[Time], 108) <> CONVERT(VARCHAR(20), d.[Time], 108)
      OR i.[Name] <> d.[Name]
      OR REPLACE(REPLACE(REPLACE(i.[Version List], ',', ''), '#', ''), 'LOCKED', '') <>
         REPLACE(REPLACE(REPLACE(d.[Version List], ',', ''), '#', ''), 'LOCKED', '')
    )
    AND i.[Version List] <> '! CHECK OBJECT VALID !'
    AND d.[Version List] <> '! CHECK OBJECT VALID !';

  IF @SetupSQLCheckObjectLockType <> 0 AND @CheckIsLocked = 1 AND
     @LockedBy <> @Username AND @LockedBy <> ''
  BEGIN
    SELECT
      @ObjectTypeText =
        CASE @ObjectType
          WHEN 1 THEN 'Table'
          WHEN 2 THEN 'Form'
          WHEN 3 THEN 'Report'
          WHEN 4 THEN 'Dataport'
        WHEN 5 THEN 'Codeunit'
        WHEN 6 THEN 'XMLport'
        WHEN 7 THEN 'MenuSuite'
        WHEN 8 THEN 'Page'
        ELSE ''
      END,
    @ObjectIdText = @ObjectId;

  SET @Message =
    CHAR(13) + CHAR(10) + CHAR(13) + CHAR(10) +
    'OBJECT MANAGER ERROR:' + CHAR(13) + CHAR(10) +
    'Object %s %s - %s is locked by %s' + CHAR(13) + CHAR(10);
  RAISERROR(@Message, 16, 1, @ObjectTypeText, @ObjectIdText, @ObjectName, @LockedBy);
  ROLLBACK TRANSACTION;
END;

IF @LockObjectAtSaving = 1 AND @IsModification = 1 AND @LockedBy = ''
BEGIN

  IF EXISTS(
    SELECT 1
    FROM [OM - Repository Setup]
    WHERE [Use Repository] = 1)
  BEGIN
    SET @Message =
      CHAR(13) + CHAR(10) + CHAR(13) + CHAR(10) +
      'OBJECT MANAGER ERROR:' + CHAR(13) + CHAR(10) +
      'The option ''Lock Object at Saving'' cannot be ' +
      'used in combination with repository' + CHAR(13) + CHAR(10);
    RAISERROR(@Message, 16, 1);
    ROLLBACK TRANSACTION;
  END;

  INSERT INTO [OM - Object Lock]
    (
      [Object Type], [Object No_], [Locked By],
      [Lock Date],
      [Lock Time],
      [Deleted], [Token No_], [Branch No_]
    )
  SELECT
    @ObjectType, @ObjectId, @Username,
    CAST(CONVERT(VARCHAR(20), GETDATE(), 112) + ' 00:00:00' AS DATETIME),
    CAST('17540101 ' + CONVERT(VARCHAR(20), GETDATE(), 108) AS DATETIME),
    0, 0, '';
  SELECT @OldVersionList = [Version List]
  FROM Inserted;

  IF CHARINDEX('#', @OldVersionList) = 1
    SET @NewVersionList = '#LOCKED,' + SUBSTRING(@OldVersionList, 2, 100)
  ELSE
    SET @NewVersionList = 'LOCKED,' + @OldVersionList;

  IF LEN(@NewVersionList) <= 80
    UPDATE [Object]
    SET [Version List] = @NewVersionList
    WHERE [Object].[Type] = @ObjectType
    AND [Object].[ID] = @ObjectId
    AND CHARINDEX('LOCKED', [Version List]) = 0;

  SET @IsLocked = 1;
  SET @LockedBy = @Username;

END;

IF @SetupSQLCheckObjectLockType = 2 AND @CheckIsLocked = 1 AND @IsLocked = 0
BEGIN
  SELECT
    @ObjectTypeText =
      CASE @ObjectType
        WHEN 1 THEN 'Table'
        WHEN 2 THEN 'Form'
        WHEN 3 THEN 'Report'
        WHEN 4 THEN 'Dataport'
        WHEN 5 THEN 'Codeunit'
        WHEN 6 THEN 'XMLport'
        WHEN 7 THEN 'MenuSuite'
        WHEN 8 THEN 'Page'
        ELSE ''
      END,
    @ObjectIdText = @ObjectId;

  SET @Message =
    CHAR(13) + CHAR(10) + CHAR(13) + CHAR(10) +
    'OBJECT MANAGER ERROR:' + CHAR(13) + CHAR(10) +
    'Object %s %s - %s is not locked' + CHAR(13) + CHAR(10);
  RAISERROR(@Message, 16, 1, @ObjectTypeText, @ObjectIdText, @ObjectName);
  ROLLBACK TRANSACTION;
END;

IF @SetupTraceModifications = 1 AND @IsModification = 1
  INSERT INTO [OM - Modification]
```

```
              (
                [Object Type], [Object No_], [Object Name],
                [Object Date], [Object Time],
                [Status], [Inserted By],
                [Insert Date],
                [Insert Time],
                [Assigned to Project No_], [Assigned By],
                [Assign Date], [Assign Time], [Auto Assigned],
                [Transport No_], [Object Date Time],
                [Locked By], [Token No_], [Traced By SQL], [SQL Trigger], [SQL Status],
                [System User], [Host Name]
              )
            SELECT
              @ObjectType, @ObjectId, @ObjectName,
              @ObjectDate, CAST('17540101 ' + CONVERT(VARCHAR(20), @ObjectTime, 108) AS DATETIME),
              0, @Username,
              CAST(CONVERT(VARCHAR(20), GETDATE(), 112) + ' 00:00:00' AS DATETIME),
              CAST('17540101 ' + CONVERT(VARCHAR(20), GETDATE(), 108) AS DATETIME),
              '', '',
              CAST('17530101 00:00:00' AS DATETIME), CAST('17530101 00:00:00' AS DATETIME), 0,
              '', CAST(CONVERT(VARCHAR(20), @ObjectDate, 112) + ' ' +
              CONVERT(VARCHAR(20), @ObjectTime, 108) AS DATETIME),
              @LockedBy, @TokenNo, 1, @Action, 1,
              SUBSTRING(SYSTEM_USER, 1, 50), SUBSTRING(HOST_NAME(), 1, 50);

        END;

      END;
    END;

  END;

END;
```

F. Update Object Table SQL Trigger

```
IF EXISTS
(
  SELECT name FROM sysobjects
  WHERE name = 'OM_Update_Object' AND type = 'TR'
)
DROP TRIGGER OM_Update_Object

GO

CREATE TRIGGER [dbo].[OM_Update_Object]
   ON  [dbo].[OM - Update Object]
   AFTER INSERT,DELETE,UPDATE
AS
BEGIN
  SET NOCOUNT ON;

  UPDATE [Object]
  SET [Object].[ID] = Inserted.[New Object No_]
  FROM [Object], Inserted
  WHERE [Object].[Type] = Inserted.[Object Type]
  AND [Object].[ID] = Inserted.[Object No_]
  AND Inserted.[Update Properties] = 0;

  UPDATE [Object]
  SET
    [Object].[ID] = Inserted.[New Object No_],
    [Object].[Modified] = Inserted.[Modified],
    [Object].[Version List] = Inserted.[Version List],
    [Object].[Date] = Inserted.[Object Date],
    [Object].[Time] = Inserted.[Object Time]
  FROM [Object], Inserted
  WHERE [Object].[Type] = Inserted.[Object Type]
  AND [Object].[ID] = Inserted.[Object No_]
  AND Inserted.[Update Properties] = 1;

  DELETE FROM [OM - Update Object] WHERE [Object Type] = -1;

END;
```