Enable

Microsoft Dynamics NAV Upgrade Toolkit

Microsoft Dynamics

UPGRADE TOOLKIT

PREFACE

This document contains the information necessary to successfully upgrade from Microsoft Dynamics™ NAV 3.7 or 4.0 to Microsoft Dynamics NAV 5.0 and to migrate to Microsoft Dynamics NAV with Microsoft® SQL Server™ 5.0.

The tools described in this document are located in the Upgrade Toolkit, on PartnerSource. You will find a detailed log of the differences between the earlier versions and Microsoft Dynamics NAV 5.0 in the `Microsoft Dynmaics NAV 50 Changes Doc.doc` file that is included in the documentation folder of ithe Upgrade Toolkit.

You should also be familiar with the symbols and typographical conventions used in the Microsoft Dynamics NAV manuals. In the list below, you can see how various elements of the program are distinguished by special typefaces and symbols:

Appearance	**Element**
CTRL	Keys on the keyboard. They are written in small capitals.
Design	Menu items and buttons in windows. They always start with a capital letter, and the access key is underlined.
Address	Field names. They appear in bold and start with a capital letter.
Department	Names of windows, boxes and tabs. They appear in bold italics and start with a capital letter.
Hansen	Text that you must enter, for example: "...enter *Yes* in this field." It is written in italics.
`fin.flf`	File names. They are written with the Courier font and lowercase letters.

TABLE OF CONTENTS

Chapter 1

The Upgrade Toolkit

The Microsoft Dynamics™ NAV Upgrade Toolkit 5.0 contains several tools which must be used when upgrading from Microsoft Dynamics NAV 3.70 or 4.0 (formerly known as Microsoft® Business Solutions–Navision®) to Microsoft Dynamics NAV 5.0. These tools and the instructions for using them are described in detail in this document.

The chapter contains:

- Description of the Upgrade Toolkit

1.1 Description of the Upgrade Toolkit

The Upgrade Toolkit consists of a set of tools and procedures that are designed to help you upgrade to Microsoft Dynamics NAV 5.0. The tools covers the upgrade for the following products:

- Microsoft Dynamics NAV 3.70
- Microsoft Dynamics NAV 4.00

The tools and procedures that you must use vary depending on the version you are upgrading from.

The data conversion tools can be used on a Microsoft Dynamics NAV database or a SQL Server database.

The Upgrade Toolkit also contains the tools for migrating from Microsoft Dynamics NAV 5.0 to Microsoft Dynamics NAV 5.0 with Microsoft® SQL Server™ and a description of the procedures involved. Upgrading to Microsoft Dynamics NAV 5.0 is the first step in the process of migrating to the SQL Server Option for 5.0. For more information, see Chapter 5.

The Upgrade Toolkit is located in the `Upgtk` folder. The files in the toolkit are grouped in the following folders:

Data Conversion Tools - This folder contains the files used to convert data from a prior version to 5.0. The files to upgrade a specific version can be found in the relevant subfolder. For more information, see Chapter 4.

Documents - this folder contains the subfolders:

Localization Instructions - These documents provide information about the data conversion tools and recommendations on how to localize and customize the upgrade tools.

Upgrade Quick Guides - This folder contains the Upgrade Quick Guides for each prior version. Partners can add to these documents to include their own customized upgrade steps.

Feature Enhancements Documents - These documents provide an overview of the new features in Microsoft Dynamics NAV 5.0 when compared to prior versions of the product. The features are classified by granule.

Object Change Tools - This folder contains tools used to change the existing objects in some of the prior versions so that they can work in the 5.0 version. For more information, see Chapter 3.

SQL Migration - This folder contains the tools needed for migrating a Microsoft Dynamics NAV database to the SQL Server Option for 5.0. For more information, see Chapter 5.

Chapter 2

Preparing to Upgrade

There are some preparations that must be made before you can upgrade the customer's database to Microsoft Dynamics NAV 5.0.

These include testing the database and ensuring that the database you are upgrading does not contain any objects that have the same name.

The chapter contains:

- Preparing to Upgrade
- Changing the Setup Properties for a SQL Server Database
- Testing the Old Database

2.1 Preparing to Upgrade

To prepare a customer's old installation for upgrading:

1. Verify that both your solution developer's license file and the customer's license file have been upgraded to the newest version of Microsoft Dynamics NAV.
2. Identify the user ID and password of a superuser in the system.

Note

Before creating a backup of the database, you must ensure that the inventory cost data in the customer's database is up to date by running the ***Adjust Cost-Item Entries*** batch job.

3. Make a backup of the entire database before you begin the upgrade process. Save the file as, for example, `data.fbk`. Keep the backup in a safe place, for several months after the upgrade is complete.

Note

Note how long it takes to complete each of the steps in the upgrade process. You can use this information to estimate the time and cost involved in future upgrades.

4. Make a copy of the customer's database, and upgrade the copy.
5. Ensure that no other users are connected to the database before you carry out each part of the upgrade process.

Note

Because the customer's license file may be too limited to carry out some of the steps in the following sections, use your solution developer's license from this point on. During the upgrade process, you will use the development environment (C/SIDE®) in both the old and new version.

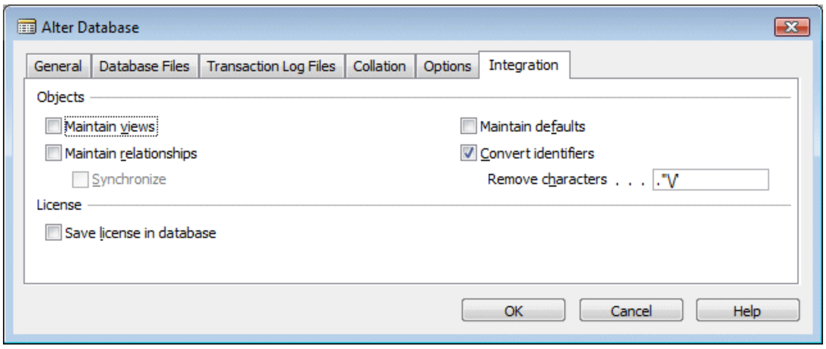
2.2 Changing the Setup Properties for a SQL Server Database

If you are upgrading a SQL Server database, then you must check that the database properties are set correctly. Do this as follows:

1 Open the database in the version of Microsoft Dynamics NAV that the customer is currently using.

2 Click File, Database, Alter Database.

3 Click the ***Integration*** tab and ensure that there are no check marks in the **Maintain Views**, **Maintain Relationships** and **Maintain Defaults** fields.



☐ Maintain views
☐ Maintain relationships
☐ Synchronize
☐ Maintain defaults
☑ Convert identifiers
Remove characters . . . ."\/'
☐ Save license in database

4 Click OK.

After the upgrade is complete, you can reenable the settings if desired.

2.3 Testing the Old Database

To determine the state of the customer's current database and correct any database errors that may exist, follow the procedures described in this section. This will ensure that no errors exist in the database that will be used as the basis for the upgrade.

1. Open the customer's database in the version that the customer is currently using.
2. Run a database test to determine the state of the customer's database.

   Test everything except field relationships between tables. If the test fails, you must follow the workflow for repairing damaged databases.
3. Run the remaining part of the database test, that is, test field relationships between tables.

   This will allow you to determine the extent of any data inconsistency that exists in the database. If any error messages appear during the test, note their content and number. You must then decide whether or not these errors will affect the upgrade.
4. Compile all the objects in the database.

   Make a list of the objects that cannot be compiled. At some point, you must decide what to do with the objects that cannot be compiled. They will create problems if you ignore them.

You have tested the customer's old database and are now ready to begin the process of upgrading the database.

Chapter 3

Customizing the New Standard Objects

Any customizations that have been implemented in the customer's old database must be implemented in the new database. You must identify and correct any illegal locktable calls that are made in the customer's database. We also recommend implementing some changes that will facilitate migrating to the SQL Server Option.

The chapter contains:

- Customizing the New Standard Objects

3.1 Customizing the New Standard Objects

The next stage in the upgrade process involves identifying any customizations that have been made in the customer's current database and redesigning a database in Microsoft Dynamics NAV 5.0 so that it can accommodate these changes.

The procedures described in this chapter are the fastest and safest way to upgrade a customer's installation. There are other ways of upgrading, but we do not recommend them.

Identifying Customized Objects

Start by identifying any customer-specific changes that have been made to the customer's old standard application. You then implement those changes in a new standard Microsoft Dynamics NAV 5.0 application.

In order to identify the customizations that the customer has made, you must have an old standard database to compare with the user's current database. This old base version must be the one that you received from your local Microsoft country or region office in English.

To compare the objects in the customer's old database with the standard database that came with the old version that the customer is using, and to implement the necessary changes in a new standard Microsoft Dynamics NAV 5.0 database, follow this procedure:

1 Create a backup of the customer's current version of Microsoft Dynamics NAV from a client computer and then uninstall the old version of Microsoft Dynamics NAV on that computer.

2 Install the newest version of Microsoft Dynamics NAV on the client computer.

3 Restore the objects from the backup of the old customized database into the new version of Microsoft Dynamics NAV. Do not restore the data now.

4 Restore the backup of the old standard database that you have just created into the new version of Microsoft Dynamics NAV.

Remove IDs

Variables and text constants have unique IDs in Microsoft Dynamics NAV.

If you have created custom objects in an old version, you would have assigned them IDs that were not used in the base version at the time. However, new objects may have been added to the base version that now overlap with your custom objects.

To eliminate this problem, you can remove the IDs from all objects in the new customized database and from the English version of the old base version, import them again and let C/SIDE assign correct IDs.

To remove IDs, follow this procedure:

1 In the version of the customer's current database that you created above, open the *Object Designer* and export all objects to a text file by clicking File, Export.

Save the file as, for example, `ObjectsID_cust.txt`.

2 In Windows Explorer, copy the *remid* tool from the `Upgtk` folder to your working folder, for example `C:\W1\NF\400`. The files that you need are:

```
remid.bat
```

```
remid2.exe
```

3 Go to your Command Prompt and change directory to the working folder, for example `C:\W1\NF\400`.

4 Enter the following command line, but exchange the file names for the names of the files that you have created:

```
REMID ObjectsID_cust.txt ObjectsNoIDs_cust.txt
```

where `REMID` is the name of the tool, `ObjectsID_cust.txt` is the name of the source file that you created in Step 1 above and `ObjectsNoID_cust.txt` is the name of the target file.

5 When the tool has finished its work, you can use the target file during the merge process.

You can only use this tool to remove IDs on text files that contain IDs.

You must now repeat this procedure for the English version of the old base version. When you have completed the procedure, you must have two files with the objects from the customer's current database and the English version of the old standard database, respectively. You must generate these two files before the merging process, because removing the IDs makes the merge much easier.

Attention

Do not remove IDs from the new base version.

When you have completed the merge process described below and you import the merged objects into the new Microsoft Dynamics NAV 5.0 database, C/SIDE will assign correct unique IDs to the customized objects.

Comparing Versions

You must now use a file comparison tool such as the Developer's Toolkit for Microsoft Dynamics NAV to compare the files you created in the previous procedure and find and view the object changes in detail. Make a log of these changes.

Note

For more information about comparing and merging objects, see the separate documentation for the Developer's Toolkit for Microsoft Dynamics NAV. This documentation is located with the Developer's Toolkit on PartnerSource.

When you have compared the old base version and the customer's current database, you must evaluate the changes that have been made.

You must decide which changes you want to duplicate in the standard database that comes with Microsoft Dynamics NAV. Some changes may already be part of this new standard Microsoft Dynamics NAV database and can be skipped during the upgrade. To make the evaluation easier, you should refer to any documentation that describes the changes that have been made to the customer's database.

Be careful to note any duplicated object names and duplicated field names within an object because these must be renamed. You must also note any duplicated function numbers and control numbers because these must be renumbered.

Merging

You must now merge the customized objects into the new Microsoft Dynamics NAV database. For this process, you should use the Developer's Toolkit for Microsoft Dynamics NAV.

In the Developer's Toolkit for Microsoft Dynamics NAV, you must import object files from the old standard version, the customer's current version and the new Microsoft Dynamics NAV database. For this process you need the two files that you created in the section called *Remove IDs* on page 8. You also need an object file with all objects from the new Microsoft Dynamics NAV database.

Merge the customizations into the new Microsoft Dynamics NAV database, which in the Developer's Toolkit is called *the New Custom version*, and correct any errors that the tool finds.

When you have completed the merge, export the New Custom version to a text file, import it into the new Microsoft Dynamics NAV database and compile all objects.

You now have a new customized database. This database contains all the objects that have been customized to include the changes that were made in the old database.

These customized objects and all the other Microsoft Dynamics NAV 5.0 objects must be exported and saved so that you can import them again after you have restored the backup of the customer's old database into a new database in Microsoft Dynamics NAV.

To export the objects:

1. Open the new customized Microsoft Dynamics NAV database. Open the *Object Designer* and compile all the objects.

   It should now be possible to compile all the objects without encountering any errors.

2. Export all the objects in the new customized Microsoft Dynamics NAV database, and name the object file, for example, `objects.fob`.

3. Make a backup file of the demonstration company and name it, for example, `demo.fbk`.

Chapter 4

Upgrading the Old Database

This chapter describes the procedures needed to convert the existing data to the 5.0 database structure.

- Overview
- Upgrading Company-Specific Data
- Upgrading Data Common to All Companies
- Deleting Unused Tables and Upgrade Toolkit
- Testing the Database
- Installing the Clients

4.1 Overview

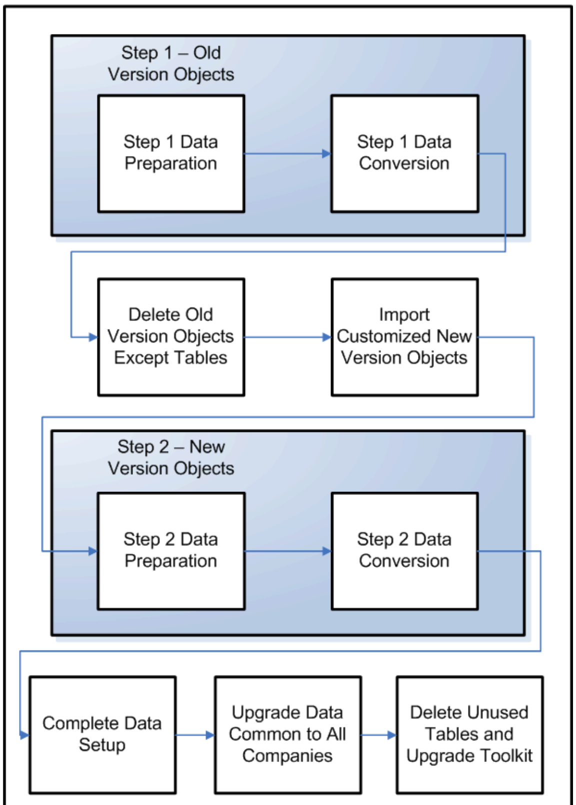
Converting the Old Database

To upgrade the database, you need to:

- Convert the data to work with the 5.0 version
- Test the data in the 5.0 version
- Install the new 5.0 clients.

Data conversion tools are provided to convert the existing data with the old version table and field structure so it can function with the new table and field structure in the new version. Only the table objects and table data are modified by the data conversion tools. All other objects such as forms, reports, codeunits and dataports are "upgraded" as part of the customization merge processes. The new customized forms, reports, dataports and codeunits are exported after the merge and imported during the data conversion. For more information on this process, see Chapter 3, *Customizing the New Standard Objects* and the documentation for the Developers Toolkit.

The process involved in performing the data conversion is outlined in the following diagram:



Data Conversion Workflow

The Step 1 and Step 2 terminology is used as follows:

- *Step 1* - All tasks that involve changes to the existing data with the old version objects.
- *Step 2* - All tasks that involve changes to the existing data with the new version objects.

As you can see from the diagram, Step 1 and Step 2 are made of smaller sub-steps. There are also sub-steps that must be performed between Step 1 and Step 2 and after Step 2. The sub-steps in the diagram above can be further broken down into tasks.

Here is a list of tasks that must be performed during and after the data conversion process, with references to the relevant sections in this chapter:

Step1 Data Preparation

- Task 1 - Create the New 5.0 Database
- Task 2 - Import Upgrade Step 1 Objects
- Task 3 - Data/Object Changes Prior to Step 1

Step 1 Data Conversion

- Task 4 - Step 1

Step 1 Cleanup

- Task 5- Delete Objects
- Task 6- Import All Customized 5.0 Objects
- Task 7- Compile Imported Objects

Step 2 Data Preparation

- Task 8- Import Upgrade Step 2 Objects
- Task 9- Data/Object Changes Prior to Step 2

Step 2 Data Conversion

- Task 10 - Step 2

Setup Completion

- Task 11 - Complete Data Changes After Step 2
- Task 12 - Initialize the Company

4.3 Upgrading Data Common to All Companies

- Roles and Permissions
- Database Key Groups
- Security Changes Between SQL Server Option 3.7 and 5.0

4.4 Deleting Unused Tables and Upgrade Toolkit

4.5 Testing the Database

4.6 Installing the Clients

Quick Guides

This manual contains the procedures to upgrade the database for versions 3.70 and 4.0. The Upgrade Toolkit also contains quick guides related to each of these versions. Each quick guide lists the actions you must perform when upgrading from a specific version. The following table lists the versions and the related quick guides:

Version	Quick Guide
3.70	`Upgrading Microsoft Dynamics NAV 3.70`
4.0	`Upgrading Microsoft Dynamics NAV 4.0`

Prior Version Information

Where possible, references to prior versions are made in this chapter and the appendixes providing further information on the data conversion process.

User Portal and Commerce Portal

The User Portal and Commerce Portal functionality is not included in the 5.0 version. During the data conversion process the existing data is removed from the old tables, and is no longer available in version 5.0.

4.2 Upgrading Company-Specific Data

This part of the upgrade procedure involves upgrading the customer's database to Microsoft Dynamics NAV 5.0. The first task is to restore the backup of the customer's database that was made earlier.

Step1 Data Preparation

Task 1 - Create the New 5.0 Database

To restore the database:

1. Ensure that you have made the preparations and testing described in Chapter 2.
2. Create a new database in Microsoft Dynamics NAV 5.0.

   We recommend that the new database is large enough to contain the objects in the `object.fob` plus twice the size of the existing data in the old customer database. You can see an example of the size of an old and a new database below:

   Old Database

   (old objects = 60 MB) + (existing data = 500 MB) = 560 MB

   New Database

   (new objects = 80 MB) + (2 x (existing data = 500 MB)) = 1080 MB

   To see the size of data in the old database, run report 104001, ***Table Information***.
3. Restore the backup of the customer's old database that was saved as `data.fbk` into this new database. This is the backup that you made in step 3 on page 4.

Note

You do not have to use the backup and restore method. You can simply make a copy of the old database and then open this copy with the new 5.0 client. You should convert the database when prompted to do so and then expand the size of the database as recommended in the backup and restore procedure. For large databases, this alternative method will normally be quicker than the backup and restore method.

Task 2 - Import Upgrade Step 1 Objects

- Open the *Object Designer* and import the relevant fob file for the version from the table below. This file is in the Upgrade Toolkit on PartnerSource.

Version	Fob file
3.70	`Upgrade370500.1.fob`
4.0	`Upgrade400500.1.fob`

Task 3 - Data/Object Changes Prior to Step 1

Rounding Precisions for Additional Reporting Currency

All Versions

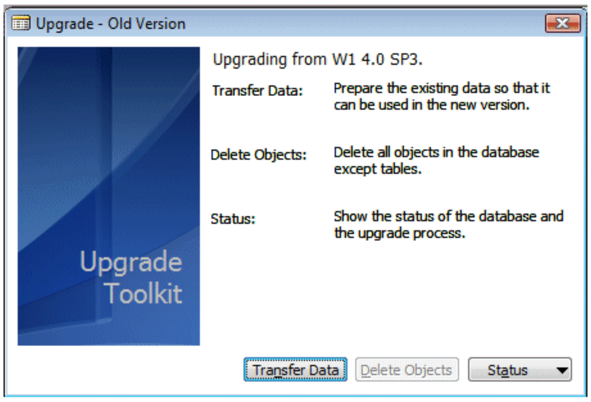
If you have selected an additional reporting currency, you must ensure that that ***Currency*** table contains a record for that currency. You must also ensure that the additional reporting currency record does not have 0 in either the **Unit-Amount Rounding Precision** or **Amount Rounding Precision** fields.

Step 1 Data Conversion

Task 4 - Step 1

When you are upgrading from any version:

1. Open the *Object Designer*, select form 104001, ***Upgrade - Old Version***, and click Run (ALT+R).



Upgrading from W1 4.0 SP3.

Transfer Data: Prepare the existing data so that it can be used in the new version.

Delete Objects: Delete all objects in the database except tables.

Status: Show the status of the database and the upgrade process.

2. Click Transfer Data.

   This runs codeunit 104045, ***Upgrade 5.00 Step 1***. You must run this codeunit for each company. For more information on what the program does when you run codeunit 104045, see *Codeunit 104045, Upgrade 5.00 Step 1* on page 47.

If you have not completed the data preparation in *Task 3 - Data/Object Changes Prior to Step 1*, the ***Upgrade Error Log*** window will appear. You will not be able to proceed with the upgrade until you correct all errors in this window.

3. To correct an error, select the error and click Show. This will open the relevant form and, where possible, show the specific record with the error.

If an error occurs when you are running the codeunit 104045, ***Upgrade 5.00 Step 1***, the program will store the last successfully completed process in the ***State Indicator*** table.

4. To see a status of what has been upgraded and what remains, click Status, Status Indicator.

Note

If the upgrade process fails, the ***State Indicator*** table contains information on which tables have been upgraded and which have not.

Step 1 Cleanup

Task 5- Delete Objects

The next stage in the upgrade procedure involves importing the customer's customized objects into the new Microsoft Dynamics NAV 5.0 database. Before importing these objects, you must delete most of the objects that already exist in the database.

1 On form 104001, ***Upgrade - Old Version***, click Delete Objects.

2 This runs codeunit 104002, ***Delete Objects Excl. Tables***. The codeunit deletes all the objects that are not tables. This will ensure that no conflicts occur during the following steps.

Note

Be aware that when you close the company after having imported the new objects, you might receive the error message "Codeunit 5072 does not exist" or "Tabledata 5121 does not exist." You can ignore this message; it appears because there are references to these deleted objects still lying in memory. The message will not appear again.

Import Customized Objects

Task 6- Import All Customized 5.0 Objects

1 Using the *Object Designer*, import the new customized Microsoft Dynamics NAV 5.0 objects into the database.

These are the objects that were exported to the `objects.fob` file in step 2 on page 10.

When the import starts, a warning appears informing you that some objects with conflicting versions already exist in the database.

2 Click OK and the ***Import Worksheet*** window appears.

3 Click *REPLACE ALL*, and then click OK to import the objects.

If any errors occur while the program is importing the objects, the process will be canceled. Correct the problem in the new customized database and export the objects.fob file again (see step 2 on page 10). In the database you are upgrading, import the objects again.

Task 7- Compile Imported Objects

- Compile all the objects in the customer's new database.

  The database has now been restored into Microsoft Dynamics NAV 5.0 and the new customized objects have been imported into the database.

Compilation Warning:

Some of the objects you deleted in *Task 5- Delete Objects* on page 17 no longer exist in the 5.0 version. However, the old version tables that were not deleted may have included references to these deleted objects. As a result, you can expect compilation errors concerning some tables. A list of these tables is provided after this note. If there are compilation errors in other objects than those listed, it may be that the customized 5.0 objects you imported contain compilation errors. These errors must be fixed before you proceed with the upgrade.

All versions

When you compile all objects, the following tables will not compile:

Table ID	Table Name
168	***Job Budget Line***
5119	***Outlook Integration Log***
5120	***Outlook Int. Conflict Field***
5121	***Synchronization Entry***
6203	***Web Portal User Role***
6206	***E-Mail Queue***
6209	***Template Property Collection***
6210	***Template Property***
6211	***Page Property Collection***
6212	***Page Property***
6213	***Compiled Property***
6215	***Picture***
6216	***Commerce Portal Setup***
6217	***Web Site***
6221	***Notification Process***
6227	***Web Portal Role***
6228	***Web Portal Permission***
6231	***Web Portal Role Permission***
6240	***Comp. Prop. Multilang.***
6241	***Reverse Auction Header***

6242	***Reverse Auction Line***
6243	***Commerce Portal Comment Line***
6244	***Commerce Portal Login***
7902	***Smart Tag Profile***
7903	***Smart Tag Dataset***
7904	***Smart Tag Dataset Folder***
7905	***Smart Tag Profile Folder***
7950	***Smart Tag Data Setup***
104003	***Upgrade Error Log***

Step 2 Data Preparation

Task 8- Import Upgrade Step 2 Objects

- Open the *Object Designer* and import the relevant fob file for the version from the table below. This file is in the Upgrade Toolkit on PartnerSource.

Version	Fob file
3.70	`Upgrade370500.2.fob`
4.0	`Upgrade400500.2.fob`

Task 9- Data/Object Changes Prior to Step 2

All versions

Set Language

- Click Tools, Language and select the language of the old customer database from the list.

For example, if the old customer database was in German, you must select German as the current language.

Microsoft Dynamics NAV 3.70 and 4.0

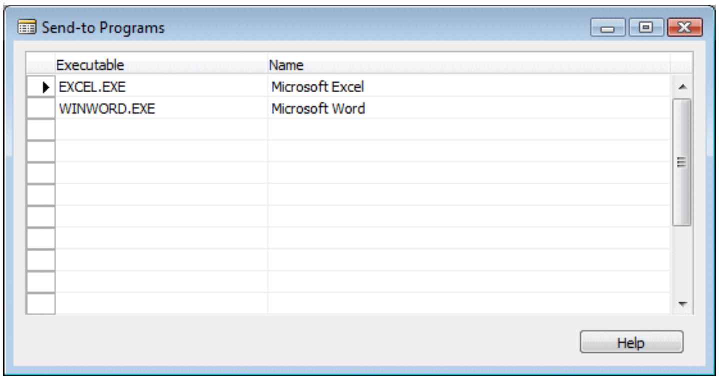
Office Integration

If you want to use the Office Integration features introduced in 5.0, you will need to do some manual setup. You need to set up the Office programs in the *Send-to Programs* window and you need to import the default style sheets, or your own custom style sheets.

To set up Word and Excel in the ***Send-to Programs*** window:

1 Open the *Object Designer*, select form 691, ***Send-to Programs***, and click Run (Alt+R).

2 Fill in the lines like this:



Executable	Name
EXCEL.EXE	Microsoft Excel
WINWORD.EXE	Microsoft Word

To Import the default style sheets that come with the application, do this:

1 Open the *Object Designer*, select form 690, ***Manage Style Sheets***, and click Run (Alt+R).

2 Select **Show - Style sheets common to all forms**.

3 Click Functions, Import.

4 In the **Style Sheet** field, click the AssistButton and browse to the style sheet folder. The default style sheets are installed here: `Program files/Microsoft Dynamics NAV/CSide Client/Stylesheets`.

   Select `NavisionFormToExcel.xslt.`

5 Fill in the **Name** field, for example you can enter *Default*. In the **Send-to Program** field, select *Microsoft Excel*.

6 Click OK.

Repeat the procedure for the default Word template: `NavisionFormToWord.xslt`

Team-Meeting Organizers

Microsoft Dynamics NAV 3.70

When you are upgrading from Microsoft Dynamics NAV 3.70, you must choose meeting organizers for each team that has a Team To-do of the type Meeting assigned to it in the system. To do this:

1 Select form 104090 ***Team - Meeting Organizers***, and click Run (ALT+R).

2 Fill in the code of the salesperson who will be the meeting organizer for these team to-dos of the type Meeting.

Human Resource Units of Measure

Microsoft Dynamics NAV 3.70

When you are upgrading from version 3.70, the upgrade to Microsoft Dynamics NAV 5.0 requires some manual changes concerning Units of Measure in the Human Resources application area.

From Microsoft Dynamics NAV 4.0, a unit of measure can be related to a base unit of measure. In codeunit 104045, ***Upgrade 5.00 Step 1***, all units of measures previously used by a resource are stored in table 5220, ***Temp Human Res. Unit of Measure***. As a default, the **Qty. per Unit of Measure** field is set to the value 1.

To set up the base unit of measure and set up the correct relation between alternative units of measure and the base unit:

1 Open the *Object Designer*, select form 5236, ***Human Res. Units of Measure***, and click Run (Alt+R).

2 Select the base unit of measure and ensure that the value of the **Qty. per Unit of Measure** field is 1.

3 For each other unit of measure, enter the correct number of base units in the **Qty. per Unit of Measure** field. For example, if base unit of measure is set to *Hour*:

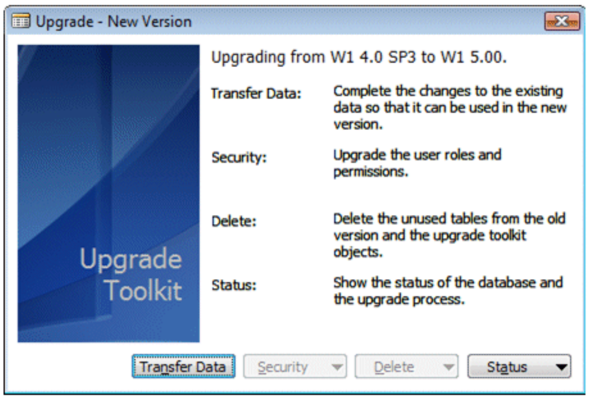
Code	Qty. per Unit of Measure
Day	8
Hour	1

Step 2 Data Conversion

Task 10 - Step 2

When you are upgrading from any version:

1 Open the *Object Designer*, select form 104002, ***Upgrade - New Version***, and click Run (Alt+R).



Upgrading from W1 4.0 SP3 to W1 5.00.

Transfer Data:	Complete the changes to the existing data so that it can be used in the new version.
Security:	Upgrade the user roles and permissions.
Delete:	Delete the unused tables from the old version and the upgrade toolkit objects.
Status:	Show the status of the database and the upgrade process.

2 Click Transfer Data.

This runs codeunit 104048, ***Upgrade 4.00 Step 2***. You must run this codeunit for each company. For more information on what the program does when you run codeunit 104048, see *Codeunit 104048, Upgrade 5.00 Step 2*, on page 49.

If you have not completed the data preparation in *Task 9- Data/Object Changes Prior to Step 2*, the ***Upgrade Error Log*** window will appear. You will not be able to proceed with the upgrade until you correct all errors in this window.

3 To correct an error, select the error and click Show. This will open the relevant form and where possible show the specific record with the error.

If an error occurs when you are running the codeunit 104048, ***Upgrade 4.00 Step 2***, the program will store the last successfully completed process in the ***State Indicator*** table.

4 To see a status of what has been upgraded and what remains, click Status, Status Indicator.

Note

If the upgrade process failed, the ***State Indicator*** table will contain information on which tables have been upgraded and which have not. If the upgrade completed successfully, the ***State Indicator*** table will be empty.

Setup Completion

Task 11 - Complete Data Changes After Step 2

Source Codes and Source Code Setup

All versions

Update the ***Source Code*** table and ***Source Code Setup*** table for any additional source codes in the new version.

Task 12 - Initialize the Company

All versions

1 Run codeunit 2, ***Company-Initialize***.

2 To display the new main menu, close the company and open it again/click ALT+F1.

3 Open form 531, ***Setup Checklist***. This will automatically update the checklist for the 5.0 data structure.

You have now upgraded the company-specific data.

4.3 Upgrading Data Common to All Companies

The data contained in the tables that are not company specific are not upgraded by the upgrade tool. This includes the following tables:

Table ID	Table Name
2000000003	***Member Of***
2000000004	***User Role***
2000000005	***Permission***
2000000053	***Windows Access Control***
2000000203	***Database Key Groups***

You will need to manually update the data in these tables to ensure the data can be used with the 5.0 features.

Roles and Permissions

Note

The following procedure assumes the customer has used the standard roles and permissions provided with the demo company as the basis for the roles and permissions in their database. If this is not the case, the upgrade of roles and permissions may have to be performed manually.

To upgrade user roles and permissions:

1 Open form 104002, ***Upgrade - New Version***, and click Security, Import Roles.

2 Import the standard user roles for the new version from a text file. See note below.

3 Open form 104002, ***Upgrade - New Version***, and click Security, Import Permissions.

4 Import the standard permissions for the new version from a text file. See note below.

Note

Two text files may be provided by your local Microsoft country or region office with the local upgrade toolkit. If these two text files are not included, you can use dataport 104001, ***Import/Export Roles***, and 104002, ***Import/Export Permissions*** to export the new 5.0 standard roles and permissions from the demo company on the 5.0 Product CD.

5 To update the roles and permissions:

If the customer has changed the Read/Write/Modify/Delete/Execute rights for some of the standard permissions, you will need to redo these changes manually. Also if the customer has customized the roles and permissions, you will need to redo these changes manually.

6 When you have finished the roles and permissions upgrade, you should thoroughly test that the existing logins can perform all their tasks without any permissions errors.

Database Key Groups

Any changes to database key groups must be updated manually.

Note

You must perform the update of database key groups *after* you have performed the upgrade of company data. This is because any changes in the keys assigned to key groups will be imported with the 5.0 tables included in the 5.0 objects fob file. (See *Task 2 - Import Upgrade Step 1 Objects* on page 15).

To upgrade database key groups:

1 For version 3.70, add a new database key group, *ConvLoc*.

2 For all versions, add the following new database keygroups:.

New Key Groups
Acc(Cons)
Acc(D/C)
Acc(IC)
Cust(Age)
Cust(Comm)
Cust(Int)
Dim(Setup)
Intrastat
Item(Adj)
Item(Post)
Item(Trkg)
PrePmt
SalesTax
SOP(Blank)
SOP(CRM)
VAT
VATUnreal
Vend(Age)
VIES_EC

Security Changes Between SQL Server Option 3.7 and 5.0

If you are upgrading from SQL Server Option version 4.0 (before SP1) or earlier, you must synchronize the security system after the upgrade.

SQL Server Option 3.70 or 4.0 before SP1

The security functionality for the SQL Server Option was enhanced in version 4.0 SP1. Microsoft Dynamics NAV allows you to specify the level of security that you want to implement in each database. You can choose between two different security models:

- Standard Security
- Enhanced Security

The main difference between these two security models is the way in which they synchronize the Microsoft Dynamics NAV security system with SQL Server and the way that they integrate the Microsoft Dynamics NAV security system with Windows authentication.

To change the security model used in the database or to synchronize the security system, you must have permission to access the Microsoft Dynamics NAV security system. To access the security system, you must be:

- A member of the sysadmin server role on SQL Server or be a member of the db_owner database role for the database in question.
- Assigned the SUPER role in Dynamics NAV.

Furthermore, if you want to change security models, you must ensure that both of the extended stored procedures that come with Dynamics NAV have been added to the instance of SQL Server that you are using. These extended stored procedures are called:

- `xp_ndo_enumusergroups`
- `xp_ndo_enumusersids`

These extended stored procedures are part of the `xp_ndo.dll` that comes on the Dynamics NAV product DVD.

Note

You can read more about the changes to the security system in *Installation and System Management: SQL Server Option for the C/SIDE Client*, which you can find on the product DVD.

Synchronizing the Security System

It is important to remember that the synchronization process is *not* automatic and that the Dynamics NAV security system must be synchronized with SQL Server after certain events, including when you have converted a database.

Converting the Database

When you convert a database to Microsoft Dynamics NAV, it is automatically assigned a security model.

The following table lists the default values:

Conversion Path	Default Security Model
3.70 - 5.0	Standard Security Model
4.0 before SP1	Enhanced Security Model
4.0 SP1 or later	Keeps the currently selected security model

After you have converted the database, you can change the security model that the database uses.

If you convert a Microsoft Dynamics NAV 3.70 database to a Microsoft Dynamics NAV 5.0 database and choose enhanced security, you must be aware of the following changes that were made to the security system:

- New objects have been added to the ***Permission*** table and you must update it.
- The implementation of permissions is more restrictive.

Updating the Permission Table

After you upgrade an old database to Microsoft Dynamics NAV 5.0, you must manually update the ***Permission*** table to include the new MenuSuite and XMLport objects.

To update the ***Permission*** table:

1 Open the Object Designer, select table 2000000005, ***Permission*** and click Design.

2 In the Table Designer window, select field number 3, **Object Type** and open the ***Properties*** window (Shift+F4).

3 In the ***Properties*** window, ensure that XMLport and MenuSuite are added to the comma separated lists in the OptionString, OptionCaption and OptionCaptionML fields.

4 Close, save and compile the table.

Permissions are More Restrictive

In Microsoft Dynamics NAV 5.0, you must ensure that each user has explicit permission (either direct or indirect) to all the tables that they need to perform their tasks. This includes both system and application tables.

For example, in Microsoft Dynamics NAV a table can contain a FlowField that generates sums based on values that are stored in another table. In this case, the user must have permission to read both tables or they will not be allowed to read the first table.

This means that it might be necessary to change the permissions for some of the existing roles.

Important

Every time you make any changes to the Microsoft Dynamics NAV security system, you must synchronize the security system with SQL Server.

Warning

Never use SQL Server tools to add or delete information stored in the Microsoft Dynamics NAV Windows ***Login*** table or ***User*** table because this information is used during the synchronization process.

4.4 Deleting Unused Tables and Upgrade Toolkit

Delete Old Tables No Longer Used in 5.0

You may want to delete the tables from the old version that are only used during the upgrade process.

To delete unused tables that have been carried over from the old version:

On form 104002, ***Upgrade - New Version***, click Delete, Unused Old Tables.

Delete Upgrade Toolkit Objects

You may want to delete the upgrade toolkit objects after the upgrade.

To delete the upgrade toolkit objects:

1 On form 104002, ***Upgrade - New Version***, click Delete, Upgrade Toolkit.

This will delete all upgrade toolkit objects except table 104002. To delete the table:

2 Open the *Object Designer*, select tabel 104002, ***Status log***, and press F4.

4.5 Testing the Database

To determine the state of the customer's current database and correct any database errors that may exist, follow the procedures described in this section. This will ensure that no errors exist in the database after the upgrade.

1. Run a database test to determine the state of the database you have just upgraded.

   Test everything except field relationships between tables. If the test fails, you must follow the workflow for repairing damaged databases.

2. Run the remaining part of the database test, that is, test field relationships between tables.

   This will allow you to determine the extent of any data inconsistency that exists in the database. If any error messages appear during the test, note their content and number.

3. To verify that the customer's license file includes all the necessary permissions in the upgraded solution, use the customer's license file to test the functionality in the database.

4. Test the upgraded item ledger entry invoiced quantities to ensure that any date compressed item ledger entries have been upgraded correctly.

4.6 Installing the Clients

Now you are ready to install the new version of the program on the client computers. You must perform the following tasks:

1. Make a new backup of the new upgraded database, thereby ensuring that you have a new backup of your updated database.
2. Remove all the earlier versions of the application from the client computers.
3. Install the new version of Microsoft Dynamics NAV on all the client computers and on the server.

Chapter 5

Migrating to Microsoft Dynamics NAV 5.0 with Microsoft SQL Server

This chapter contains instructions for migrating from Microsoft Dynamics NAV 5.0 to Microsoft Dynamics NAV 5.0 with Microsoft SQL Server.

The chapter contains:

- Preparing to Migrate
- Checking the Old Database
- Migrating the Old Database

5.1 Preparing to Migrate

The data conversion tools can be used to upgrade data from one version of Microsoft Dynamics NAV with Microsoft SQL Server to another. This chapter is only relevant if you are migrating from a Microsoft Dynamics NAV database to the SQL Server Option.

Note

If you want to migrate from an earlier version of Microsoft Dynamics NAV to the SQL Server Option for 5.0, you must first upgrade to Microsoft Dynamics NAV 5.0.

If you have successfully upgraded the customer's Microsoft Dynamics NAV application to Microsoft Dynamics NAV 5.0, you will already have implemented the necessary changes to the objects in the customer's database. You will have also implemented the necessary locking changes in any customer-specific objects that existed in the earlier version of the application.

To prepare the customer's Microsoft Dynamics NAV 5.0 installation for migrating:

1 Verify that both your solution developer's license file and the customer's license file have been upgraded to SQL Server Option for Microsoft Dynamics NAV 5.0.

2 Identify the user ID and password of a superuser in the system.

3 Make a backup of the entire database before you begin the migration process. Keep the backup in a safe place, and keep it for a long time.

Note

Note how long it takes to complete each of the steps in the migration process. You can use this information to estimate the time and cost involved in future migrations.

4 Make a copy of the customer's database, and migrate the copy.

5 Ensure no other users are connected to the system before you carry out each part of the migration process.

6 Install Microsoft SQL Server 2000 or 2005 on the server computer.

Note

Because the customer's license file may be too limited to carry out some of the steps in the following sections, use your solution developer's license from this point on.

5.2 Checking the Old Database

To determine the state of the customer's current database and correct any database errors that might exist, follow the procedure described in this section. This will ensure that no errors exist in the database that will be used as the basis for the migration.

1 Open the customer's Microsoft Dynamics NAV 5.0 database, and check that no other users are currently using the system.

2 Run a database test to determine the state of the customer's Microsoft Dynamics NAV 5.0 database.

Test everything except field relationships between tables. If the test fails, you must follow the workflow for repairing damaged databases.

3 Run the remaining part of the database test, that is, test field relationships between tables.

This will allow you to determine the extent of any data inconsistency that exists in the database. If error messages appear during the test, note their content and number. Decide whether or not these messages will affect the migration process.

4 Compile all the objects in the database.

Make a list of the objects that cannot be compiled. At some point, you must decide what to do with the objects that cannot be compiled. They will create problems if you ignore them.

5 Open the *Object Designer* and import the `Migrate.fob` file.

This file is located in the Upgrade Toolkit on PartnerSource. The import begins and a message appears.

Click Yes to import the objects.

The following objects are imported:

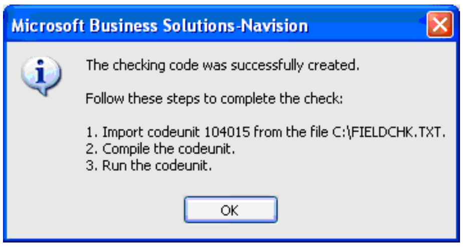
Type	**No.**	**Name**
Table	104010	***Incorrect Data Value***
Table	104011	***Code Field Information***
Form	104010	***Incorrect Data Values***
Form	104013	***Code Field Information***
Codeunit	104010	***Create Field Checking Code***
Codeunit	104011	***Date Check Management***
Codeunit	104012	***Code Check Management***
Codeunit	104013	***Date Check Indicator Mgt.***
Codeunit	104014	***Date Check Indicator Mgt. 2***
Codeunit	104015	***Field Check***

Note

If you are using the `migrate.fob` file included in the 5.0 Upgrade Toolkit to migrate to the SQL Server Option for Microsoft Dynamics NAV 5.0, you will get import/compilation errors if you are not using the 5.0 client and executables. This is because the codeunit 104011, ***Data Check Management***, contains text variables that are 1024 characters in length. This increased text variable length was only introduced with the 4.0 C/SIDE and executables.

6 In the *Object Designer*, select codeunit 104010, ***Create Field Checking Code***, and click Run (ALT+R).

Codeunit 104010 generates some new code in a text file with this path and name: `c:\fieldchk.txt`. You can change the file name and path by modifying the codeunit.



The checking code was successfully created.

Follow these steps to complete the check:

1. Import codeunit 104015 from the file C:\FIELDCHK.TXT.
2. Compile the codeunit.
3. Run the codeunit.

Import the `c:\fieldchk.txt` file.

The following objects are imported:

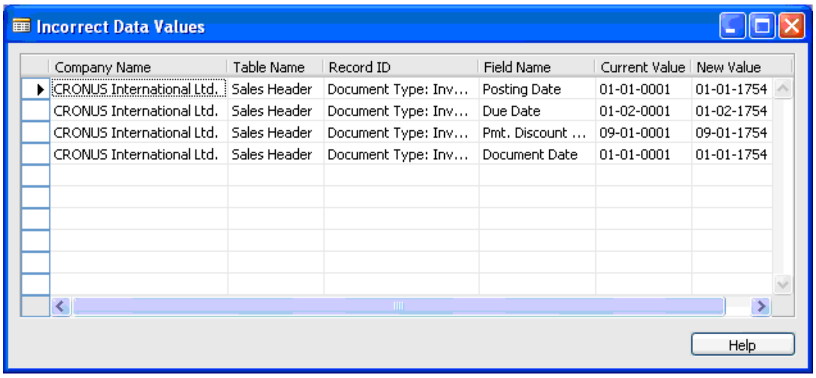
Type	No.	Name
Codeunit	104015	***Field Check***

7 Select codeunit 104015, ***Field Check***, and compile it.

8 Run codeunit 104015, ***Field Check***.

Codeunit 104015 checks that every date, text, code and decimal value in the database can be stored in a SQL Server database. It also checks that nonzero decimal values will not be rounded to zero when they are stored in the SQL Server database. A progress indicator will help you monitor this process. This involves the program reading most of the records in the database and may therefore take some time.

If any values need to be changed, the program shows a list of the incorrect values and the suggested new values in the ***Incorrect Data Values*** window:



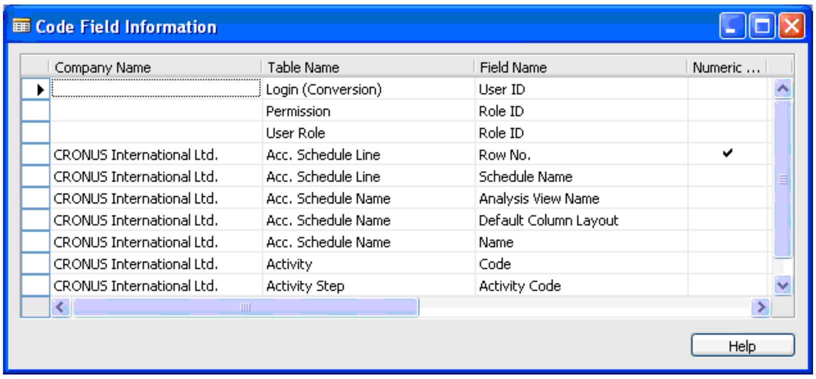
Company Name	Table Name	Record ID	Field Name	Current Value	New Value
CRONUS International Ltd.	Sales Header	Document Type: Inv…	Posting Date	01-01-0001	01-01-1754
CRONUS International Ltd.	Sales Header	Document Type: Inv…	Due Date	01-02-0001	01-02-1754
CRONUS International Ltd.	Sales Header	Document Type: Inv…	Pmt. Discount …	09-01-0001	09-01-1754
CRONUS International Ltd.	Sales Header	Document Type: Inv…	Document Date	01-01-0001	01-01-1754

You can modify these suggestions if you want. When you close the window, a message will appear asking you whether or not you want to implement the changes. If you click Yes, the program will implement the changes. The program will read and modify a small number of records in the database during this step.

Codeunit 104015 also checks the code fields in your data. If your code fields contain numeric values of varying lengths, it will list them for you. If you sort by these fields the resulting sorting will be incorrect. Furthermore, any filters using these fields and containing numeric ranges will give unexpected results. One way of overcoming this sorting problem is to represent these code fields as integers. This can be done if the code fields only contain numbers and these numbers do not start with zeros.

If there are any inconsistencies, they will be listed in the form 104013, ***Code Field Information***. To open this form:

- In the *Object Designer*, run form 104013, ***Code Field Information***.



Company Name	Table Name	Field Name	Numeric ...
	Login (Conversion)	User ID	
	Permission	Role ID	
	User Role	Role ID	
CRONUS International Ltd.	Acc. Schedule Line	Row No.	✓
CRONUS International Ltd.	Acc. Schedule Line	Schedule Name	
CRONUS International Ltd.	Acc. Schedule Name	Analysis View Name	
CRONUS International Ltd.	Acc. Schedule Name	Default Column Layout	
CRONUS International Ltd.	Acc. Schedule Name	Name	
CRONUS International Ltd.	Activity	Code	
CRONUS International Ltd.	Activity Step	Activity Code	

This window lists all of the code fields used in the database. It also contains information about whether the code field is numeric only, a compatible integer and if it is zero padded. The window displays the minimum and maximum number of digits that the field contains. The window also displays the SQL Data Type that is used in the field and whether it contains any numbering conflicts and the name of any linked tables.

This multitude of columns means that there are numerous ways of sorting this information.

The **SQL Data Type** field shows how a code field is represented on SQL Server. You can change the **SQL Data Type** property for each code field. If you set the **SQL Data Type** field to *Integer* for a code field, you will be allowed to store only positive numbers in the code field. This will ensure that numeric sorting is done correctly.

For more information about numbering and sorting in the SQL Server Option, see the manual *Application Designer's Guide*.

You can represent a code field that is already in use as an integer, only if the **Numeric Only** field and the **Compatible Integer** field are checked and the **Zero Padded** field is cleared.

You must check whether any of the following fields are listed in the ***Numbering Conflicts*** window.

Table Name	**Table ID**	**Field Name**	**Field No.**
G/L Account	15	**No.**	1
Acc. Schedule Line	85	**Row No.**	3
VAT Statement Line	256	**Row No.**	4

If any of these fields appear in the ***Numbering Conflicts*** window, you should be aware that any totals based on them may be inconsistent. When possible, this situation can be corrected either by changing the data, or by changing the SQL Data Type property for these fields to *Integer*.

5.3 Migrating the Old Database

Now that the old database has been checked and modified to ensure that it is compatible with SQL Server, you can migrate to the SQL Server Option.

1 Open the customer's 5.0 database in Microsoft Dynamics NAV 5.0, and check that no other users are currently using the system.

2 Make a backup of the customer's Microsoft Dynamics NAV 5.0 database and name it, for example, `data.fbk`.

3 Create a new database with the SQL Server Option.

4 Restore everything from the backup saved in `data.fbk`.

Restoring the backup into the SQL Server Option database will take some time and will generate a very large transaction log. We recommend that you make a new SQL Server backup of the database before you start to work with it. This will truncate the transaction log and give you a new SQL Server backup as your starting point.

Note

To verify that the customer's license file includes all the necessary permissions in the migrated solution, you must use the customer's license file in the following step.

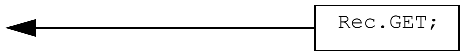
5 Test the newly migrated database to ensure that you do not encounter version control conflicts.

Microsoft Dynamics NAV 3.70

If you receive error messages informing you that a record has been modified by another user, even though you are the only user currently using the system, you have version control conflicts. These problems occur because the SQL Server Option for Microsoft Dynamics NAV 4.0 has stricter version control than previous versions. Any attempt to modify or delete a record will fail if the timestamp on the version of the record that you have read is different from the timestamp on the actual record stored in the database.

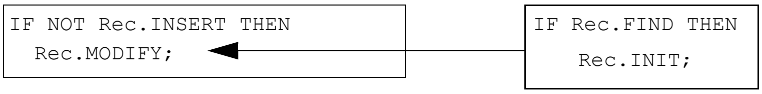
The following tables contain examples of the old incorrect code and the new correct code that must be inserted to avoid version control conflicts:

```
Rec.GET;
NewRec.GET;
NewRec.<Data Field> := 'New';
NewRec.MODIFY;
Rec..<Data Field> := 'Old';
Rec.MODIFY;
```



```
Rec.GET;
```

```
Rec.<Key Field> := 1;
Rec.<Data Field> := 'New';
```



```
IF NOT Rec.INSERT THEN
  Rec.MODIFY;
```

```
IF Rec.FIND THEN
    Rec.INIT;
```

Appendix A
Service Management and Jobs

This appendix describes the upgrade of Service Management and Jobs. It describes tasks that you must complete before upgrading Jobs or Service Management, and it also includes descriptions of what the upgrade tool does.

The appendix contains:

- Upgrading Service Management
- Upgrading Jobs

.

A.1 Upgrading Service Management

Due to major changes in the design of the Service Management functionality, it is necessary to remove unnecessary service-related data stored in the database to avoid data inconsistency and business logic gaps when upgrading to Microsoft Dynamics NAV 5.0.

Preconditions for Upgrading Service

The service posting logic has been changed significantly for version 5.0. We have added partial and line-by-line posting functionality to the Service Management module. An additional posting option, *Consume*, has been implemented. In addition, the validation rules on the service documents have been updated.

Attention

For the new functionality to work correctly, the database must not contain any un-posted service documents (quotes, orders, invoices or credit memos). In addition, posted service order documents must be deleted because this document type will no longer exist.

Upgrading Service Management

The Service Management upgrade is divided into two steps. In the first step, the upgrade tool scans the database and deletes irrelevant data.

Version 3.7 and 4.0

Codeunit 104045 blanks the following fields:

Table ID	Table Name	Field No.	Field Name
36	***Sales Header***	5900	**Service Mgt. Document**
37	***Sales Line***	5900	**Service Contract No.**
		5901	**Service Order No.**
		5902	**Service Item No.**
		5903	**Appl.-to Service Entry**
		5904	**Service Item Line No.**
		5907	**Serv. Price Adjmt. Gr. Code**
110	***Sales Shipment Header***	5900	**Service Mgt. Document**
112	***Sales Invoice Header***	5900	**Service Mgt. Document**
113	***Sales Invoice Line***	5900	**Service Contract No.**
		5901	**Service Order No.**
		5902	**Service Item No.**
		5903	**Appl.-to Service Entry**
		5904	**Service Item Line No.**

Table ID	Table Name	Field No.	Field Name
		5907	**Serv. Price Adjmt. Gr. Code**
114	***Sales Cr.Memo Header***	5900	**Service Mgt. Document**
115	***Sales Cr.Memo Line***	5900	**Service Contract No.**
		5901	**Service Order No.**
		5902	**Service Item No.**
		5903	**Appl.-to Service Entry**
5107	***Sales Header Archive***	5900	**Service Mgt. Document**
5907	***Service Ledger Entry***	38	**Chargeable**
5911	***Service Mgt.Setup***	10	**Check Customer Credit Limit**
		53	**Default Invoice Document**
		15	**Contract Inv. Line Text (Code)**
		16	**Contract Line Inv. Text (Code)**
5914	***Loaner Entry***	13	**Posted**
6084	***Service Line Price Adjmt.***	12	**Quantity to Invoice**
		13	**New Quantity to Invoice**
6650	***Return Shipment Header***	5900	**Service Mgt. Document**
7311	***Warehouse Journal Line***	60	**Reference Document**
7312	***Warehouse Entry***	60	**Reference Document**

Codeunit 104048 does the following:

- The tool implements the new structure and updates some fields. In the Sales & Marketing objects, the tool also removes service fields that are no longer used.
- The following fields are set to *TRUE*:

Table ID	Table Name	Field No.	Field Name
279	***Extended Text Header***	5902	**Service Invoice** (set to *TRUE*)
		5903	**Service Credit Memo** (set to *TRUE*)

.

A.2 Upgrading Jobs

The main change in the design of the Jobs functionality is that phases, tasks and steps are no longer used. These concepts have been replaced with job task lines, which have different job task types. The new design of the Jobs functionality is based on the design of the chart of accounts.

The translation of Phase, Task and Step is implemented in the following way:

- For each job, for each combination of phase, task and step that is used, a job task line is created.
- Each job task line is assigned a job task number, for example 1000 and 1001.
- The job task type (which is similar to the **Account Type** field in the chart of accounts) determines at what level totals are created. The job task type is assigned according to the contents of the **Job Task Total Level** field on form 104005, ***Job Upgrade Setup***.
- In all related tables, such as table 169, ***Job Ledger Entry*** or table 111, ***Sales Shipment Line*** the **Phase**, **Task** and **Step** fields are replaced by a **Job Task No.** field, which is filled in with the number of the job task line that corresponds to the combination of job, phase, task and step.

Preconditions for Upgrading Jobs

Certain conditions must be met in the database in order to upgrade. Before upgrading, do the following:

- To determine to which job task lines the job task type *End-Total* will be assigned, open the Object Designer and run form 104005, ***Job Upgrade Setup***, and then fill in the **Job Task Total Level** field.
- Ensure that the following fields are empty:

Table ID	Table Name	Field No.	Field Name
36	***Sales Header***	56	**Job No.**
37	***Sales Line***	45	**Job No.**
		47	**Phase Code**
		48	**Task Code**
		49	**Step Code**
38	***Purchase Header***	56	**Job No.**
39	***Purchase Line***	45	**Job No.**
		47	**Phase Code**
		48	**Task Code**
		49	**Step Code**
81	***Gen. Journal Line***	42	**Job No.**
207	***Res. Journal Line***	10	**Job No.**
210	***Job Journal Line***		All fields

- In the ***Job Budget Line*** table, fill in the **No.** field on all lines.
- Ensure that all items and resources to which a budget line refers have been assigned a basic unit of measure.
- If there are any budget lines with resource groups, ensure that the unit of measure code is defined on form 104005, ***Job Upgrade Setup***.

Upgrading Jobs

Codeunit 104045 performs the following tasks:

- Job ledger entries are upgraded. Each ledger entry where **Chargeable** is *True* is used to create a line in table 1003, ***Job Planning Line***, and the **Line Type** field is set to *Contract*.
- Job Ledger Entries in 5.0 support currency. For all upgraded entries, the ***Currency Factor*** field is set to *1*.
- Lines in table 168, ***Job Budget Lines***, are translated to lines in table 1003, ***Job Planning Lines*** with the **Line Type** field set to *Schedule*. Lines in table 220, ***Job Budget Entries*** are removed.
- In earlier versions, it was possible to create budget lines for resource groups. In 5.0, a resource is created for each resource group that is used in job budgets. These resources will have numbers derived from the number series on form 104005, ***Job Upgrade Setup***, if it is defined. Otherwise, the resource number series in table 314, ***Resources Setup*** will be used.

  The resources created to substitute for resource groups in budgets will be assigned the unit of measure defined on the ***Job Upgrade Setup*** form.

One Error Message for Multiple Job Journal Lines

As mentioned in *Preconditions for Upgrading Jobs*, you must delete all job journal lines before upgrading. If one journal line exists when the upgrade tool runs, you will receive an error message and the relevant journal will be opened so that you can delete the line. However, if multiple job journal lines exist, you will receive one generic error message. Then you must manually identify and delete the lines.

Appendix B

Known Issues and Upgrading 5.00: Steps 1 and 2

This appendix describes known issues that may affect the upgrade. It also describes the tasks that the Upgrade Toolkit performs when you run codeunit 104045, ***Upgrade 5.00 Step 1*** and codeunit 104048, ***Upgrade 5.0 Step 2***.

This appendix contains the following sections:

- Known Issues
- Codeunit 104045, Upgrade 5.00 Step 1
- Codeunit 104048, Upgrade 5.00 Step 2

.

B.1 Known Issues

This section describes known issues, including tasks that you may need to perform before running the upgrade tool.

RIM Toolkit

The upgrade toolkit does not remove RIM data.

Microsoft Dynamics NAV 3.70

If there is data in any of the following tables, delete the data before running step 1:

Table ID	Table Name
8600	***Implementation Status***
8601	***Mapping Header***
8602	***Mapping Lines***
8603	***Mapping Group***

Upgrading SQL Server while Upgrading Microsoft Dynamics NAV

If you upgrade to SQL Server 2005 at the same time that you upgrade to Microsoft Dynamics NAV 5.0, you could run into problems with the SQL Server and attaching the database. We recommend that you upgrade only one at a time.

Value Entries for Item Charges in Foreign Currency

Value entries for item charges that are not in the local currency will not have the document type and document line number filled in when the database is upgraded. This means that value entries for item charges in foreign currency that existed before the upgrade will lose the link to the source document.

B.2 Codeunit 104045, *Upgrade 5.00 Step 1*

The tasks performed by the codeunit are grouped into function areas.

Item Ledger Entries and Item Tracking

Codeunit 104045 performs the following tasks:

Microsoft Dynamics NAV 3.70

- In the ***Item Application Entry*** table, sets the **Consumption** and **Output is Adjusted** fields to false and sets the **Output Value Entry No.** field to *0*.
- In the ***Item Application Entry*** table, clears the **Output Completely Invoiced Date** field for outbound entries if the corresponding item ledger entry has not yet been invoiced.

Other

Codeunit 104045 clears the following fields:

Table ID	Table Name	Field No.	Field Name
27	***Item***	5406	**Indirect Cost per Unit**

Outlook Integration

Codeunit 104045 performs the following tasks:

- Blanks the following fields:

Table ID	Table Name	Field No.	Field Name
13	***Salesperson/Purchaser***		
		5086	**Outlook Contacts Folder Path**
		5087	**Outlook Tasks Folder Path**
		5088	**Outlook Calendar Folder Path**
		5089	**Navision User ID**
		5090	**Outlook User Name**
		5091	**Enable Synchronization**
		5092	**Outlook Profile Name**
		5093	**Outlook Contacts FolderID**
		5094	**Outlook Tasks FolderID**
		5095	**Outlook Calendar FolderID**
		5096	**Outlook Contacts StoreID**
		5097	**Outlook Tasks StoreID**
		5098	**Outlook Calendar StoreID**
		5100	**Notify about Contact Changes**

.

Table ID	Table Name	Field No.	Field Name
		5101	**Notify about Task Changes**
		5102	**Notify about Appmt. Changes**
		5103	**Synchronize From**
		5104	**Synchronize To**
		5105	**Synchronize To-dos**
		5106	**Version No.**
		5107	**Notify about Team To-do Changes**
		5108	**No. of Errors and Warnings**
		5109	**No. of Conflicts**
		5110	**No. of Unsynchd. Categories**
5050	***Contact***		**Version No.**
5080	***To-do***		**Version No.**
5119	***Outlook Integration Log***	ALL	
5120	***Outlook Int. Conflict Field***	ALL	
5121	***Synchronization Entry***	ALL	
5181	***Unsynchronized Category***	ALL	
5199	***Attendee***		**Recipient ID**

Job Queue

Codeunit 104045 performs the following tasks:

- Blanks the following tables:

Table ID	Table Name
5980	***Job Scheduler Mgt. Setup***
5982	***Job Scheduler Log***

- Transfers the existing job scheduler data to the new job queue entry table. Codeunit 104048 transfers data between tables as shown below. After the transfer of data the codeunit deletes the data in the old table:

Table ID	Name		Table ID	Name
5981	***Job Scheduler Setup***	>	472	***Job Queue Entry***

B.3 Codeunit 104048, *Upgrade 5.00 Step 2*

The tasks performed by the codeunit are grouped into function areas.

Item Ledger Entries, Value Entries and Item Tracking

Codeunit 104048 perform the following tasks:

Microsoft Dynamics NAV 3.70

- If there are any value entries where the valuation date is 12.31.9999, the codeunit resets the valuation date to the posting date of the value entry.
- If the **Expected Cost Posting to G/L** field is set to yes in the inventory setup, the codeunit fills the **Expected Cost Posted to G/L** and **Exp. Cost Posted to G/L (ACY)** fields in the ***Value Entry*** table with the expected cost amount.

Date Formulas and Date-Time Fields

Codeunit 104048 perform the following tasks:

- Copies the date formulas from the temporary tables into the new date formula fields in the base tables, and deletes the records in the temporary tables.
- Calculates and updates the new **Date-Time** fields in several tables based on the information in the existing **Date** and **Time** fields. These new **Date-Time** fields combine the values in the separate date and time fields into one field.

Outlook Integration

Codeunit 104048 performs the following tasks:

- In the ***Contact*** table (5050), the default value of the **E-Mail** field is copied to the new **E-Mail 2** field.
- In the ***Salesperson/Purchaser*** table (13), the default value of the **E-Mail** field is copied to the new **E-Mail 2** field.