MODULE 4: DATA UPGRADE

Module Overview

"Data Upgrade" discusses the data upgrade phase of the upgrade project. It shows the general process that developers follow to upgrade a database from Microsoft Dynamics® NAV 2009 to Microsoft Dynamics® NAV 2013 by using the Upgrade Toolkit. "Data Upgrade" also discusses ways to resolve typical issues that may arise during data upgrade.

Objectives

- Review the data upgrade workflow.
- Explain how to perform trial data upgrade by using the Microsoft Dynamics NAV Upgrade Toolkit.
- Explain testing of data upgrade results.
- Examine the typical issues that may arise during data upgrade and ways to troubleshoot them.

Data Upgrade Workflow

Data upgrade, also known as data conversion or data migration, converts information in the customer's database that uses the old version table and field structure so that it works with the new table and field structure in the solution's target version.

To upgrade your database to Microsoft Dynamics NAV 2013 consists of the following stages:

- Upgrade company-specific data
- Upgrade data that is common to all companies
- Delete unused tables
- Test the new database

You perform data upgrade in several runs by using different environments, as follows.

Data Upgrade Run	Environment
Trial Data Upgrade	Development environment
Pre-production Data Upgrade	Test or staging pre-production environment at customer site
Live Data Upgrade	Production environment at customer site

During trial data upgrade, you may have to customize Upgrade Toolkit objects according to the specific needs of the source solution and data. You also create the step-by-step script document that covers all actions that you must perform when you run the pre-production and live data upgrade at customer site.

 ***Note:*** *You can use the Upgrade Quick Guide document as a basis for the step-by-step script. It also helps you track the time that is required to perform each step. This is important for planning the live data upgrade. Find this document in the Upgrade Toolkit subfolder of the Microsoft Dynamics NAV 2013 installation folder.*

For database upgrade, use the data conversion tools that are provided with Microsoft Dynamics NAV 2013. These tools convert the existing data with the old version's table and field structure so that it functions together with the new version's table and field structure. The data conversion tool modifies only table objects and table data. All other objects are upgraded as part of the customization merge process during code upgrade and object transformation. New customized pages, reports, codeunits, and XMLports are imported during data conversion. This is step 2 in the following example.

This "Data Conversion Workflow" figure outlines the data conversion process.

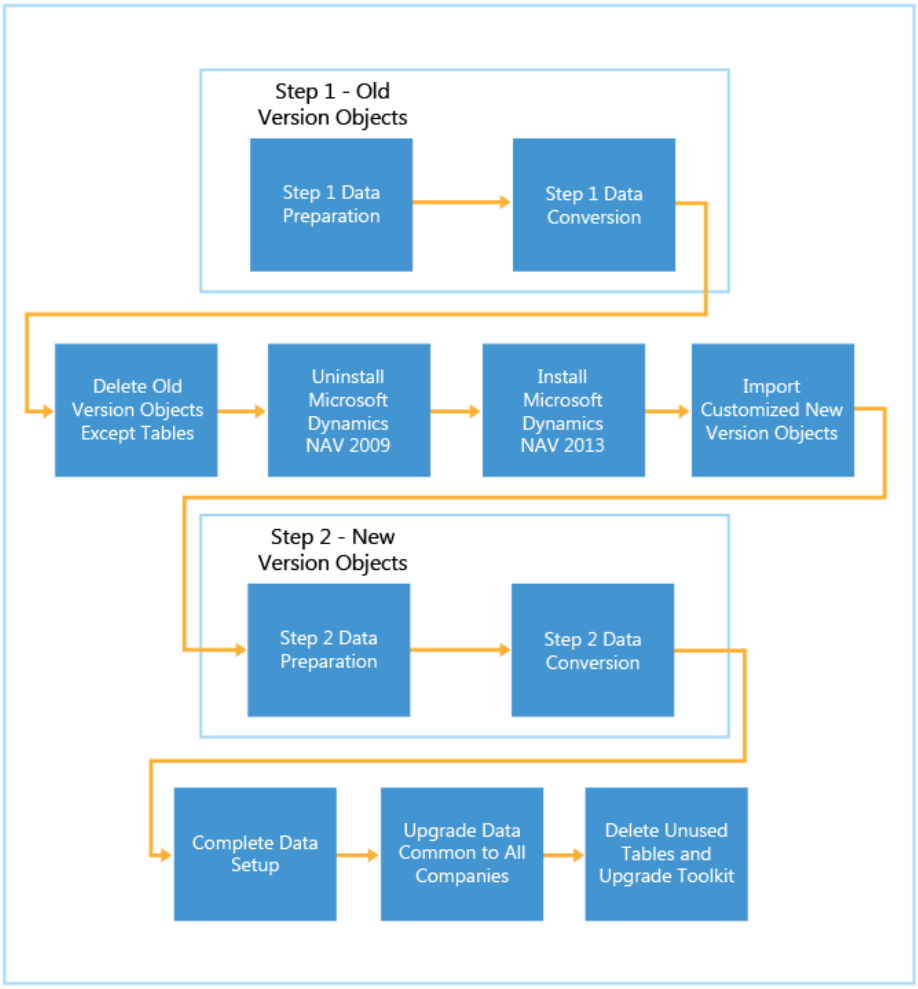

FIGURE 4.1:DATA CONVERSION WORKFLOW

Data Upgrade Step 1

During upgrade step 1, you use the Upgrade Toolkit to make the information in the source database compatible with Microsoft Dynamics NAV 2013.

Task 1 – Open the Customer's Existing Database By Using the Microsoft Dynamics NAV 2009 Classic Client

Open the customer's old database by using Microsoft Dynamics NAV 2009 Classic Client. Make a copy of the database or create a backup, and then open or restore it with the Classic client.

Note: *In Microsoft Dynamics NAV 2013, the component that was previously called the Classic client is now called the Microsoft Dynamics NAV Development Environment, and is no longer a user client. Because tasks 1 through 5 use the Microsoft Dynamics NAV 2009 version of this component, the Microsoft Dynamics NAV 2009 name is used, which is Classic client.*

At this point, test the database by using the standard database tests in Classic client, as follows:

1. Click **File > Database > Test**. The **Test Database** window appears.
2. Select all check boxes except **Test field relationships between tables**.
3. Click **OK** to start the test.
4. After the check is completed, test the database for table and field relations.

If any errors occur, you should resolve them before continuing. Errors in data can interfere with the upgrade process and may even cause the process to fail. It is best to first resolve the errors, and then continue.

Task 2 – Import Upgrade Toolkit Objects for Step 1

The Upgrade Toolkit is a set of Microsoft Dynamics NAV objects. To start using the toolkit, you must import this set of objects into the database. You use the Upgrade Toolkit objects to perform a data upgrade from Microsoft Dynamics NAV 2009 to Microsoft Dynamics NAV 2013.

To import the Upgrade Toolkit in the Classic client, follow these steps:

1. Click **Tools > Object Designer** to start the Object Designer.
2. Click **File > Import** to start the import process.
3. Browse to the UpgradeToolKit\Data Conversion Tools\601\Upgrade601700.1.fob file, and then click **Open**. You receive a message that the client cannot import the objects because of version conflicts. Click **OK** to open the **Import Worksheet** window.

Note: *When you upgrade a local version, make sure that you import the .fob file from the UpgradeToolKit\Local Objects folder on the Microsoft Dynamics NAV media. Do not import the .fob file from the Data Conversion Tools folder.*

4. In **Import Worksheet**, click **Replace All**.
5. Click **Yes** to complete importing the objects. After the import is complete, the **Import Objects** window shows the summary.
6. Click **OK** to close the **Import Objects** window.

The following table shows the standard application objects that are included in the Upgrade601700.1.fob file.

Table ID	Table name
Table 239	BOM Register
Table 355	Ledger Entry Dimension
Table 356	Journal Line Dimension
Table 357	Document Dimension
Table 358	Production Document Dimension
Table 359	Posted Document Dimension
Table 361	G/L Budget Dimension
Table 389	Service Contract Dimension
Table 827	DO Payment Credit Card
Table 5106	Document Dimension Archive
Table 5648	FA Allocation Dimension
Table 7135	Item Budget Dimension

Note: *These objects are stripped of all code and variables before they are added to the Upgrade601700.1.fob file. This guarantees that you do not receive compilation errors when you compile these objects in step 2. This enables you to focus on resolving compilation errors that are caused by customization or changes in the platform.*

If you have added your own fields to any of these tables, you have to replicate these changes after importing the objects.

Task 3 – Data/Object Changes Before Step 1

Before you run the actual data conversion in step 1, you must introduce certain changes in data and objects. Follow these steps for all companies in the database.

1. Run the **Adjust Cost-Item Entries** batch job to make sure that the inventory cost data in the customer's database is up to date.
2. If the customer's organization posts inventory costs to the general ledger, run the Post Inventory Cost to G/L batch job.

Note: *You must perform these two steps for all companies in the database.*

3. If the customer company is using any combination of production orders, transfer orders, and service orders, you must make sure that data in Table 32 Item Ledger Entry, Table 83 Item Journal Line, Table 89 BOM Journal Line, and Table 5841 Standard Cost Worksheet match the following prerequisites.

Table	Prerequisite
32 Item Ledger Entry	For each record, only one of the following fields can have a value. Verify that only one value is present for each of the following records: • Prod. Order No. • Transfer Order No. • Service Order No.
83 Item Journal Line	For each record, only one of the following fields can have a value. Verify that only one value is present for each of the following records: • Prod. Order No. • Transfer Order No. • Service Order No.
89 BOM Journal Line	If there are lines in this table, you must delete or post them before the upgrade.
5841 Standard Cost Worksheet	If there are lines in this table, you must delete or implement them before the upgrade.

Note: *Instead of manually changing the data to meet these prerequisites, you can run the codeunit 104044 Upg. Nav Old Ver.-Autocorrect. This codeunit automatically fixes any of these data issues. Analyze the code in this codeunit to make sure that the priority that is assigned to production orders, transfer orders, and service orders, matches your customer's priorities.*

The Upgrade Toolkit converts data only for the standard tables and functionality. If the table structure of the customer's solution was changed during the code upgrade, follow these steps:

1. Analyze all changes in table structure that you have made during code upgrade.
2. Modify the code of the Upgrade Toolkit objects that are based on any changes in structure between any of the tables that are included in the Upgrade Toolkit, or any other tables where there are conflicts in structure.

3. For more information about how to introduce modifications in the Upgrade Toolkit objects, refer to the Troubleshooting section of this module.
4. Save these changed objects in a separate .fob file for future use during the pre-production and live data upgrades.

If table structure changes occurred in the new version of an ISV solution that your customer uses, you must retrieve the Upgrade Toolkit objects from the vendor and use them together with your modified objects during the data upgrade.

Task 4 – Step 1 Data Conversion

Use Microsoft Dynamics NAV 2009 Classic client to perform the following steps for each company in the database:

1. In the Classic client, click **Tools > Object Designer** to open the Object Designer.
2. Select the form 104001, **Upgrade – Old Version**, and then click **Run**.
3. Click **Transfer Data**. This runs codeunit 104045, Upgrade NAV 7 Step 1.
4. To correct any errors that may occur, select the error in the **Upgrade Error Log** window, and then click **Show**. This opens the relevant form and, if applicable, shows the specific record with the error. Microsoft Dynamics NAV stores the last successfully processed table in the **State Indicator** table.
5. To see status information on what is upgraded, on the **Upgrade - Old Version** form, click **Status > Status Indicator**.

Note: *If the upgrade process fails, the **State Indicator** table contains the information about the tables that were not upgraded successfully. If the upgrade is completed successfully, the **State Indicator** table is empty.*

The program runs codeunit 104045, Upgrade Nav Old Version that performs the following tasks:

- Checks that the preconditions were met (that is, changes in data can be introduced correctly).
- Puts the information from the tables to be updated to Microsoft Dynamics NAV 2013 in dedicated buffer tables.
- Cleans up data that is not used in Microsoft Dynamics NAV 2013.

At this step, errors may occur. This can be caused by customizations to tables that are made by the partner or the customer. After encountering an error, the program displays the **Upgrade Error Log** window and stores the information about the latest successfully completed process in the **State Indicator** table. During trial upgrade runs you must make sure that the upgrade procedure

handles these errors. You can do this by customizing the upgrade toolkit to automatically fix those errors, or by fixing the errors in the data manually.

To see the size of your database, run report 104001, **Table Information**. Note the size of the database as indicated in the report, or print this report. You will need this information in Task 6 when you set the size for your new database.

Preparation for Data Upgrade Step 2

Step 1 of the Data Upgrade prepared data for the upgrade process by moving any data in tables that have conflicting structure between Microsoft Dynamics NAV 2009 and Microsoft Dynamics NAV 2013 into buffer tables. Conflicting structure can be caused by any of the following situations:

- A field is removed from the original table.
- Type of the field is changed.
- Length of the field is changed.

You must complete several more tasks before you begin step 2 of the Data Upgrade that moves the data from buffer tables into final Microsoft Dynamics NAV 2013 tables.

Task 5 – Delete Source Version Objects

You must delete all Microsoft Dynamics NAV 2009 objects that are not tables. You must also delete any tables that are obsolete in Microsoft Dynamics NAV 2013 features. If this data is still needed in Microsoft Dynamics NAV 2013, then the data from these tables is moved to buffer tables in the previous task.

To delete objects, follow these steps:

1. In the Microsoft Dynamics NAV 2009 Classic client, open Object Designer, and then select form 104001, **Upgrade - Old Version**.
2. Click **Delete Objects**. This runs codeunit 104002, Delete Old Objects that deletes all the objects that are not tables. It deletes tables that belong to the features that are not available in Microsoft Dynamics NAV 2013. This makes sure that no conflicts or compilation errors occur during the next tasks.

There are more than a hundred tables from Microsoft Dynamics NAV 2009 that are not used in Microsoft Dynamics NAV 2013.

To review the complete list of these tables, see the following website.

Eliminated Tables

http://go.microsoft.com/fwlink/?LinkId=277033

Task 6 – Create the New Microsoft Dynamics NAV Database

You performed tasks 1 to 5 in the Microsoft Dynamics NAV 2009 Classic client. Now you must create a new database for the customer's new Microsoft Dynamics NAV 2013 solution.

To create the new database, follow these steps:

1. Make a copy of the current database.
2. Close the Microsoft Dynamics NAV 2009 Classic client.

Note: *When you close the Classic client after you delete objects, you might see errors about objects that no longer exist. You can safely ignore the messages that appear because of references to deleted objects still in memory (in the* ***CompanyClose*** *function in Codeunit 1 Application Management). These messages will not appear again.*

---

3. Uninstall Microsoft Dynamics NAV 2009.

Note: *You can install Microsoft Dynamics NAV 2009 on a computer where Microsoft Dynamics NAV 2013 is already installed. If you have to open the old database by using the Microsoft Dynamics NAV 2009 Classic client after you have installed Microsoft Dynamics NAV 2013 on the same computer, you can do so. But for now you must uninstall Microsoft Dynamics NAV 2009. You have the option of reinstalling it later.*

---

4. Use the following list to verify that the user who is performing the upgrade has the necessary roles and permissions in SQL Server Management Studio.
    - Verify that the user has the db_datareader, db_datawriter, and db_ddladmin database roles for the database. Or, the user may have the db_owner role.

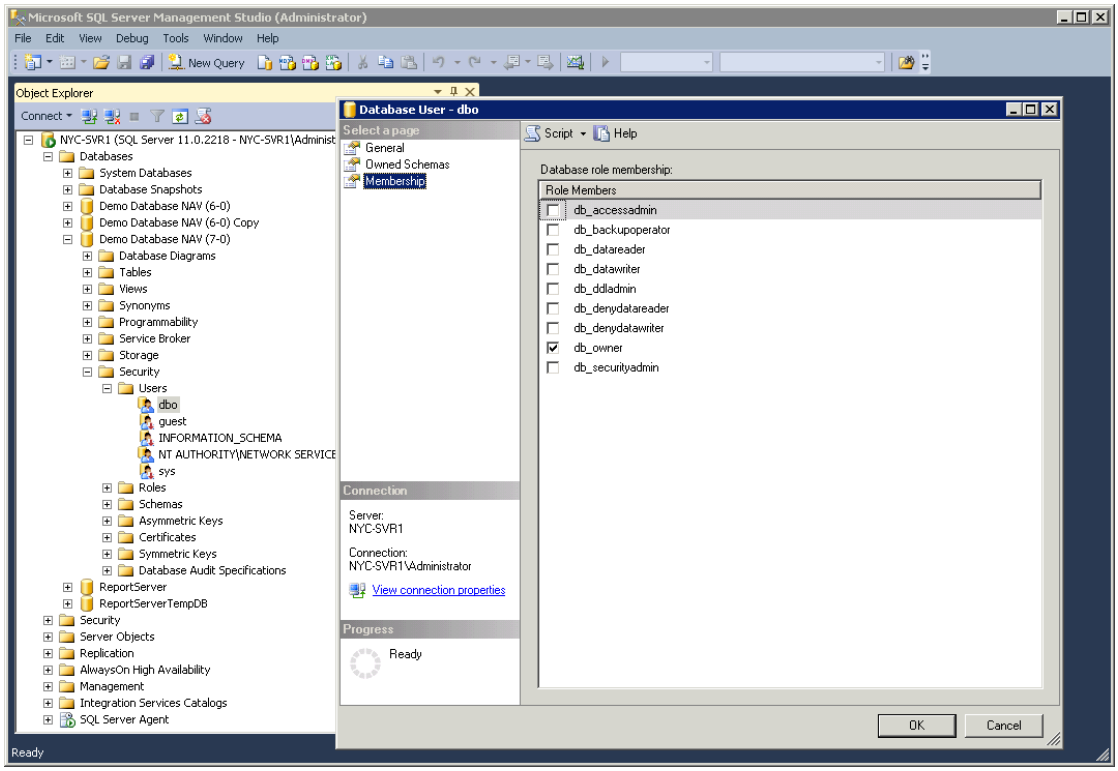

FIGURE 4.2:USER WITH THE DB_OWNER DATABASE ROLE

- Verify that the user has the sysadmin Server Role for the instance of SQL Server.

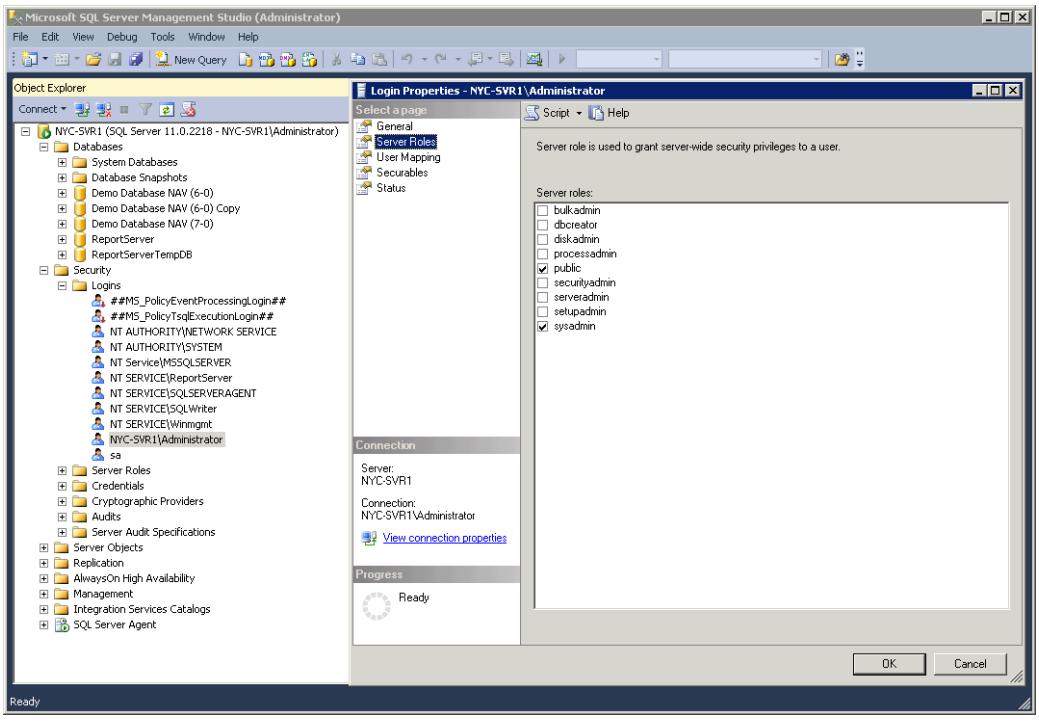

FIGURE 4.3:USER WITH THE SYSADMIN SERVER ROLE

- Verify that the default schema for the user's account is dbo. Do this by selecting the user account under Security/Users for the database to open the **Database User** dialog box. Then enter "dbo" in the **Default Schema** field.

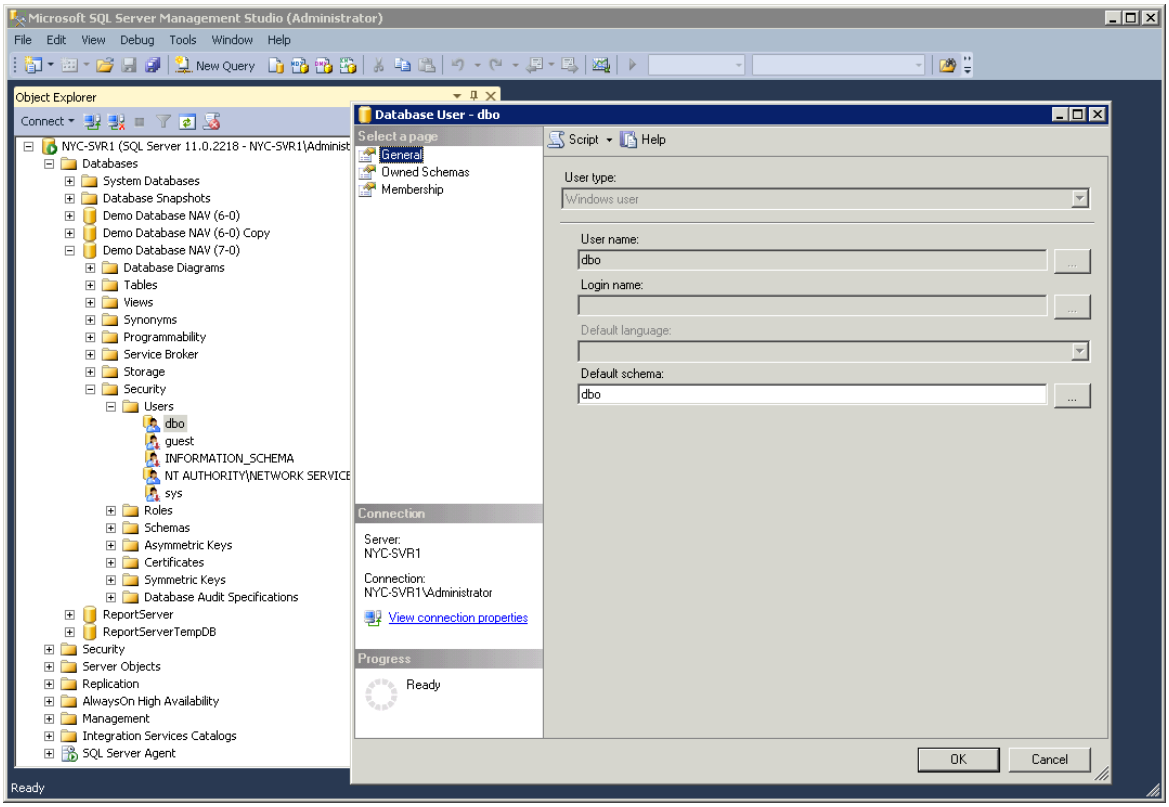

FIGURE 4.4:USER WITH THE DBO DEFAULT SCHEMA

5. Use Microsoft Dynamics NAV 2013 Setup to install the Developer Option. This includes the Microsoft Dynamics NAV Windows client, the development environment, Microsoft Dynamics NAV Server, and SQL Server database components.

Note: *For Microsoft Dynamics NAV 2013, you must install Microsoft Dynamics NAV Server on a 64-bit operating system. Microsoft Dynamics NAV also requires a 64-bit edition of SQL Server. If no usable edition of SQL Server is found on the destination computer, Setup installs a 64-bit edition of SQL Server 2012 Express.*

6. For SQL Server 2008 and SQL Server 2008 R2 editions, verify that the compatibility level of the database is set to 100. For SQL Server 2012, set the compatibility level to 110. To do this, start Microsoft SQL Server Management Studio and connect to the instance of SQL Server to which the Microsoft Dynamics NAV database is attached. Right-click the database, and then click **Properties**. In the **Database Properties** window, select the **Options** page, and then verify that the **Compatibility Level** is set correctly.
7. Open the Microsoft Dynamics NAV 2013 Development Environment.
8. Open the database that you copied in step 1. When you do this, you are prompted to convert the database. The conversion performs the following tasks:
    - Updates system tables.
    - Converts the columns for all Text and Code fields from varchar to nvarchar type.
    - Upgrades to the newest Windows collation. Using SQL collations for the database is no longer supported in Microsoft Dynamics NAV 2013. If your database is using a SQL collation, or an out-of-date Windows collation, the development environment offers to perform a collation upgrade alongside the Unicode upgrade. If you are not satisfied with the choice of collation, you can switch to the desired collation after the upgrade is complete. The collation upgrade sets the default database collation and the column-level collation for all columns in tables, except for variant, to the selected collation value. Variant columns have the collation value stamped on each instance's metadata.
    - Marks all objects in the database as "not compiled".

This conversion can be time-consuming, especially for larger databases. After the conversion, make sure that the Unrestricted Growth option is selected for all database and transaction log files for the database. To do this, click **File > Database > Alter** in the Microsoft Dynamics NAV 2013 Development Environment.

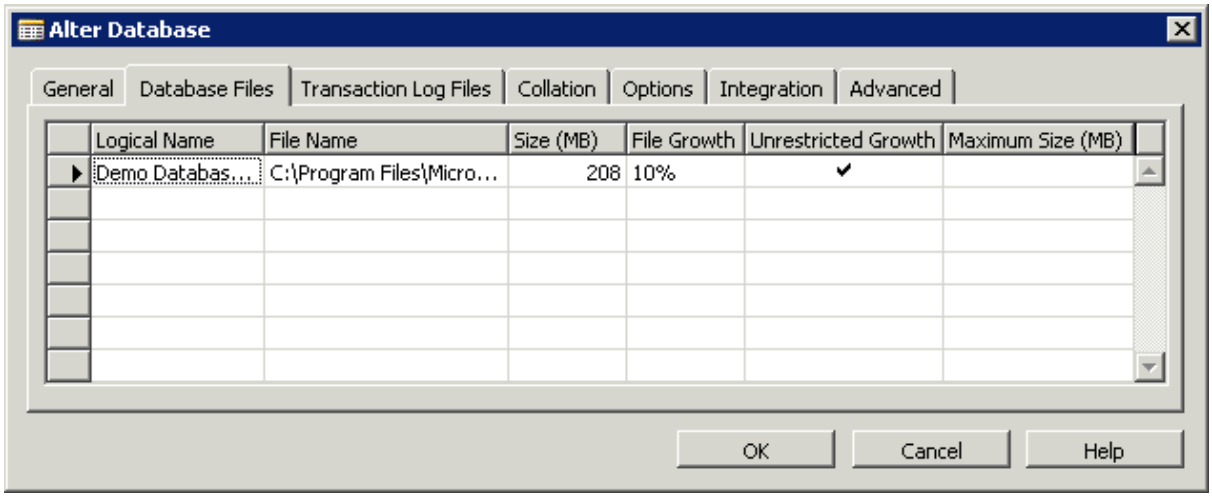

FIGURE 4.5:THE UNRESTRICTED GROWTH OPTION SELECTED FOR THE DATABASE FILE

9. Upload the Microsoft Dynamics NAV 2013 Developer's License to the Microsoft Dynamics NAV database.
10. Configure Microsoft Dynamics NAV Server to connect to the new database. Start the Microsoft Dynamics NAV Server Administration tool, and then select the appropriate Microsoft Dynamics NAV Server instance in the left pane. Click **Edit** at the bottom of the settings list in the center pane and update the Database Instance, Database Name, and Database Server options to point to the converted database. Click **Save** when you are finished.
11. Restart the Microsoft Dynamics NAV Server instance. In the left pane of the Microsoft Dynamics NAV Server Administration tool, select the local Microsoft Dynamics NAV computer. In the center pane, right-click the Microsoft Dynamics NAV Server instance, and then click **Restart**.

Task 7: Import All Customized Microsoft Dynamics NAV Objects

Now, you are ready to import the new customized objects into the database. To import the customized objects, follow these steps.

1. Using Object Designer, import all Microsoft Dynamics NAV 2013 objects into the database. These are the objects that you previously created during the Code Upgrade and Object Transformation stages and exported to a .fob file. This is the file that contains all Microsoft Dynamics NAV 2013 objects, merged with your customizations.
2. When the import starts, a warning dialog box appears and states that some objects that have conflicting versions already exist in the database. Click **OK** to open the **Import Worksheet**.
3. Click **Replace All**, and then click **OK** to complete the import.

If any errors occur when Microsoft Dynamics NAV is importing the objects, the process will stop. Resolve the issue in the new customized database and export the objects again. In the database that you are upgrading, import the objects again. It may be required to resolve the issue by adding one more step to the Data Upgrade step 1, for cleaning the existing data or saving data into a buffer table.

Task 8 – Compile Imported Objects

To continue with the data upgrade, compile all objects in the new database. To do this, follow these steps:

1. In Object Designer, click **All**.
2. Press CTRL+A to select all objects.
3. Press F11 to compile all objects.

You may receive compilation errors about the customized Microsoft Dynamics NAV 2013 objects that you imported. You must resolve these issues before continuing with the upgrade.

Data Upgrade Step 2

During Data Upgrade Step 2, you use the Upgrade Toolkit to combine the information from the source database together with the customized Microsoft Dynamics NAV 2013 objects.

Task 9 – Import Upgrade Objects for Step 2

To import upgrade step 2 objects, import the Upgrade601700.2.fob file by using Microsoft Dynamics NAV Development Environment. After you import the .fob file, close the development environment, and then restart the Microsoft Dynamics NAV Server instance again.

 ***Note:*** *When you upgrade a local version, make sure that you import the .fob file from the UpgradeToolKit\Local Objects folder on the Microsoft Dynamics NAV media. Do not import the .fob file from the Data Conversion Tools folder.*

Task 10 – Data/Object Changes Before Step 2

When you upgrade from Microsoft Dynamics NAV 2009 to Microsoft Dynamics NAV 2013, you must set the language. In the Microsoft Dynamics NAV Development Environment, click **Tools > Language**, and then select the language of the old customer database.

If you have modified the Upgrade step 1 to move data into buffer tables, you must modify the Upgrade Toolkit objects for step 2 to move data back from the buffer tables. Save the customized Upgrade step 2 objects to a separate .fob file for future use.

Task 11 – Step 2 Data Conversion

This task migrates the data from the buffer tables that are created based on Microsoft Dynamics NAV 2009 data into new customized Microsoft Dynamics NAV 2013 tables. You must run this for each company in the database.

To perform the step 2 data conversion, follow these steps:

1. Start the Microsoft Dynamics NAV client for Windows and verify the server connection. If the client opens and a Role Center is displayed, the connection is valid.

 ***Note:*** *The first time that you run the Microsoft Dynamics NAV Windows client, you receive an error message about a missing Client Add-In. This message is displayed because in the Role Center page for the default profile there is Microsoft Dynamics Business Chart control add-in. This add-in has not been declared in the* ***Client Add-In*** *table. Upgrade Toolkit inserts client add-ins later. For now, click* ***OK*** *to continue.*

---

2. In the Development Environment, click **File > Company > Open**, and then select the first company that you want to upgrade. You must repeat this procedure for each company in your database.
3. In Object Designer, run page 104002, **Upgrade - New Version**. This opens page 104002, **Upgrade - New Version**, in the Microsoft Dynamics NAV Windows client.
4. Check the value for SQL Server Name. If the value is incorrect, click **Edit** in the ribbon, and then update the value.
5. Click **Test Database Connection** to verify the connection.
6. In the **Upgrade - New Version** page, click **Transfer Data**. This runs codeunit 104048, Upgrade NAV 2009 Step 2.
7. To correct any errors, select the error, and then click **Show**. This opens the relevant page and if applicable, displays the specific record with the error.

If the upgrade process was not completed, the **State Indicator** table contains information about which tables are upgraded and which tables did not upgrade. If the upgrade completed successfully, the **State Indicator** table is empty. To view the **State Indicator** table, click **Status**, and then on the **Upgrade - New Version** page, click **Status Indicator**.

Task 12 – Complete Data Changes After Step 2

After you complete step 2 of the data upgrade, you must provide the final settings to make the upgraded database functional in Microsoft Dynamics NAV 2013.

To complete data changes after step 2, complete these steps:

1. Run page 257, **Source Codes**, in the Microsoft Dynamics NAV 2013 client for Windows.
2. Note any source codes that are added in the Microsoft Dynamics NAV 2013 database.
3. Update the **Source Code** and **Source Code Setup** tables for any additional source codes in the new version.

Task 13 – Initialize the Company

To initialize all companies, follow these steps:

1. Start Object Designer, and then connect to a company.
2. Run codeunit 2, Company-Initialize.
3. Repeat for each additional company in the database.

Upgrade of Data That is Common to All Companies

After you upgrade all company-specific data for each company in the database, you must also upgrade data that belongs to all companies. This part of the process involves the following steps:

- Upgrade roles and permissions
- Upgrade database key groups
- Assign the appropriate license type for Microsoft Dynamics NAV 2013 to Windows groups

Upgrade Roles and Permissions

The following procedure assumes that the customer used the standard roles and permissions that are provided with the CRONUS International Ltd. demonstration database as the basis for the roles and permissions in their database. If you have added, modified, or removed roles and permissions, you may have to upgrade roles and permissions manually.

Note: *In versions of Microsoft Dynamics NAV that are earlier than Microsoft Dynamics NAV 2013, the word* role *was used to describe a set of permissions for a set of objects in the Microsoft Dynamics NAV database that you can assign to one or more users. In Microsoft Dynamics NAV 2013, the new term for this concept is* permission sets*. In this topic, the term* role *is used because the process starts from* roles *and upgrades them to* permission sets*.*

To upgrade roles and permissions in the database, follow these steps:

1. In the Microsoft Dynamics NAV Development Environment, open Object Designer.
2. Connect to the database that you are upgrading.
3. Run XMLport 104001, Import/Export Roles, to export the roles that are currently defined in the database into a file.
4. Run XMLport 104002, Import/Export Permissions, to export the permissions that are currently defined in the database into a file.
5. If the customer has changed the Read/Write/Modify/Delete/Execute rights for any of the standard permissions, or customized roles in any way, you must merge these changes manually with the roles and permissions that are supplied with Microsoft Dynamics NAV 2013.
6. Connect to the CRONUS International Ltd. demonstration database that was provided with your installation set for Microsoft Dynamics NAV 2013.
7. Run XMLports 104001 and 104002 for the CRONUS demonstration database to export new standard permission sets and permissions for Microsoft Dynamics NAV 2013. Make sure that you do not overwrite the files that you created in steps 3 and 4 of this procedure.

Note: *The two text files that contain new standard permission sets and permissions may be provided by your local Microsoft country/region office with the local upgrade toolkit. In that case, skip step 7, and use these two files.*

8. Merge the roles and permissions that were exported from the customer database together with the permissions and permission sets that were exported from the CRONUS demonstration database.
9. Connect again to the database that you are upgrading.
10. Run page 9802, **Permission Sets**, and delete all permission sets except the SUPER permission set.
11. Run page 104002, **Upgrade - New Version**.
12. Click **Import Roles**.
13. Import the user roles from the merged role text file that you created in step 4.
14. Click **Import Permissions**.

15. Import the permissions from the merged permission text file that you created in step 4.
16. When you have finished the roles and permissions upgrade, verify that users can perform all tasks without any permission errors.

Upgrade Database Key Groups

Use key groups to disable or enable multiple table keys at the same time across the whole database. Microsoft Dynamics NAV 2013 provides new key groups that are not included in Microsoft Dynamics NAV 2009. You must upgrade the database key groups manually.

To upgrade the key groups, follow these steps:

1. Connect to the CRONUS demonstration database.
2. Run table 2000000203, **Database Key Groups**, and copy all rows to Microsoft Office Excel.
3. Connect to the database that you are upgrading.
4. Run table 2000000203, **Database Key Groups**, and copy all rows to Microsoft Office Excel.
5. Note those database key groups that are new in the CRONUS demonstration database for Microsoft Dynamics NAV 2013.
6. Create the new database key groups manually in the database that you are upgrading.

Assign the Windows Group License Type

Microsoft Dynamics NAV 2013 implements a new licensing model. One of the new license types is the Windows Group license type. If any users in the Microsoft Dynamics NAV 2009 database are Windows groups, you must assign the Windows Group license type to these users.

To assign the Windows Group license type to a user, follow this procedure:

1. Start Microsoft Dynamics NAV 2013 client for Windows.
2. In the **Search** field, enter "Users", and press ENTER.
3. For each user that is a Windows group, select that user, and then click **Edit**. This opens the **User Card** page.
4. In the **License Type** field, select Windows Group.
5. Click **OK** to accept changes, and then close the **User Card** page.

Note: *Windows Group licensing is a form of license multiplexing. Multiplexing does not reduce the number of user licenses that are required to access a Microsoft Dynamics NAV solution.*

To learn more about license types in Microsoft Dynamics NAV 2013, see the following website.

License Types

http://go.microsoft.com/fwlink/?LinkId=277034

Deletion of Unused Objects

After you complete the database upgrade to Microsoft Dynamics NAV 2013, you must delete obsolete tables and Upgrade Toolkit objects.

To delete unused tables, follow these steps:

1. In the Microsoft Dynamics NAV Development Environment, open Object Designer and run page 104002, **Upgrade - New Version**.
2. Click **Mark Unused Old Tables**. A message dialog box states that old unused tables are marked for deletion. Click **OK**.

Note: *This step also deletes any object translations and permissions for the unused tables.*

3. In Object Designer, set a filter on the Version List column to select tables that are marked as **Old Unused Table - marked for deletion**.
4. Click **Edit > Delete** to delete the tables.

Note: *Information in the **Upgrade Time Log** table is not deleted. Administrators may want to refer to this table to determine how long it takes to upgrade specific tables.*

To delete the Upgrade Toolkit objects, follow these steps:

1. On page 104002, click **Mark/Delete Upgrade Toolkit**. All Upgrade Toolkit objects, except tables, are removed automatically. The tables are marked for deletion.
2. In Object Designer, set a filter on the Version List column to select tables that are marked as Upgrade Toolkit Table - marked for deletion.
3. On the **Edit** menu, click **Delete** to delete the tables.

Test of the New Database

After the data upgrade is completed by developers, you must verify that the resulting solution functions correctly. To test whether the data upgrade succeeded, the project team executes the following types of testing:

- Database integrity testing
- Data acceptance testing
- Functional testing

Database Testing

Database integrity testing lets you determine whether the data in tables complies with basic data integrity rules. It determines the state of the customer's current database. You must correct any database errors that may exist after the upgrade to Microsoft Dynamics NAV 2013.

Error checking in Microsoft Dynamics NAV 2013 may discover errors that were present in the Microsoft Dynamics NAV 2009 database, but were not discovered by testing the old database.

To test the database, follow these steps:

1. In the Microsoft Dynamics NAV Development Environment, open the database.
2. Click **File > Database > Test**.

Note: *Before you run any test, configure the development environment to write test results to a file so that you do not have to acknowledge error messages for every error. To do this, in the* ***Test Database*** *dialog box, open the* ***Options*** *tab, click the* ***File*** *field, and then type or browse to a path and file name.*

3. To test everything except field relationships between tables, in the **Test Database** dialog box, open the **General** tab, and then click **Normal**. If the test fails, you must correct any errors first.
4. To test field relationships between tables, select **Test field relationships between tables**, and then click **Custom**. This lets you determine the extent of any data inconsistencies that may exist in the database. If you receive an error message during the test, note the content and number.
5. Verify that the customer's license file includes all the necessary permissions for the upgraded solution.

Data Acceptance Testing

Data acceptance testing enables the project team to make sure that no data is adversely affected by the data upgrade. This testing also verifies that the resulting solution lets the user work with the upgraded data to retrieve records, generate reports, and perform transactions.

During data acceptance testing, testers, development consultants, or technology consultants perform the following actions:

- Determine whether there are any errors or issues that are caused by the data upgrade processes.
- Test the flow of the process in a controlled environment.

The main ways of performing data acceptance testing are as follows:

- Compare the data in the target database against the information in the source database by using a comparison tool. You can create custom XMLports that export the data into comparison tool. Or, you can export the data from both the old and new database into Excel, and then run the comparison in Excel.

 ***Note:*** *As part of the data upgrade procedure, some data is deleted from the database. Therefore, there is no one-to-one comparison.*

- Run standard and customer reports in the source and target databases and compare the generated data by using a comparison tool. The standard reports that you can run are as follows:
  - G/L Register
  - Trial Balance by Period
  - Fiscal Year Balance
  - VAT Statement
  - Customer – Balance to Date
  - Vendor – Balance to Date
  - Aged Accounts Payable
  - Receivables-Payables
  - Customer – Summary Aging
  - Vendor – Summary Aging
  - Inventory Availability
  - Inventory Valuation
  - Purchase - Quote
  - Order

- Purchase – Invoice
- Purchase – Credit Memo
- Purchase – Receipt
- Purchase Statistics
- Sales – Quote
- Order Confirmation
- Sales – Invoice
- Sales – Credit Memo
- Sales – Shipment
- Sales Statistics
- Prod. Order – List
- Production Order Statistics
- Resource Statistics
- Item – List
- Customer – List
- Vendor – List
- Resource – List
- Contact – List
- Employee – List
- Fixed Asset – List
- Chart of Accounts
- Salesperson – Commission
- Job Register
- Job – Transaction Detail
- Jobs per Customer
- Jobs per Item

Run test cases to validate the data for each application module and area.

Functional Testing

After data acceptance testing is completed, and all the reported bugs and defects are fixed, verified, and closed, the project team must verify that the upgraded solution functions as expected against basic processes and upgraded data. This process is known as functional testing.

To perform functional testing the project team, that consists of testers and development consultants, must compare the actual behavior of the upgraded solution against the following aspects of the functionality:

- Standard features
- Standard features that were changed by the customer or partner
- New features that were developed by the customer, ISVs, or partners

The functional testing activities for data upgrade resemble those for code upgrade and object transformation.

At the testing stage of data upgrade, it is useful to collect the test cases that were created by users and any technical and functional documentation that is available for the source solution. Contact the development consultants who are familiar with the customized functionality.

Moreover, you must run the automated tests that were developed by using the Application Test Toolset. You can use the toolset that is provided by Microsoft. However, you must develop new tests to validate any customized functionality or new features.

In case a defect is found, you must fix it, and verify the solution.

Troubleshooting

Data upgrade is a complex process that must ensure data consistency between the source and target databases. For this purpose, the code of Upgrade Toolkit objects contains many standard data conversion algorithms. In some cases, these algorithms do not suit the custom data that is contained in the source database. Therefore, various issues may occur during the data upgrade process.

The Troubleshooting lesson describes typical issues and ways to resolve them.

All changes that were introduced when resolving these issues, such as new buffer tables and modifications in the code, must be saved to the Upgrade Toolkit objects. These objects then are used to perform pre-production and live data upgrade. This guarantees that no such issues arise.

Field to be Modified or Deleted Contains Data

During the data upgrade process, some fields are modified or removed in new custom objects. When you import these objects into the database, you may receive the message that is shown in the “Non-empty Field Error” figure.

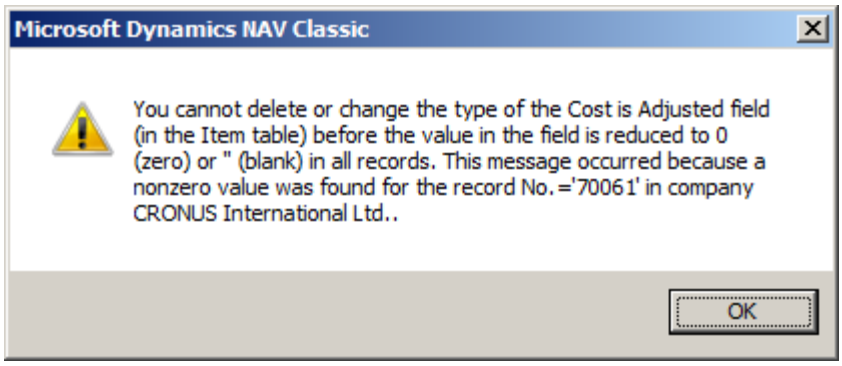

FIGURE 4.6:NON-EMPTY FIELD ERROR

Depending on the circumstances, you can resolve this issue in one of the following ways:

- If the field ID or type is changed in the target solution, move the data to a similar field, as follows:
  a. Create a buffer table that includes key fields as in the source table, and the field that causes the issue.
  b. During the first step of the data upgrade process, copy the data to the created buffer table and clear the field in the initial table.
  c. During the second step of the data upgrade process, move the data back to the appropriate field of the initial table with modifications (according to key fields).
  d. Clean up and delete the record from the buffer table.

- If the field data is not necessary, clear the values.
- If the field data is necessary, and you cannot move data to another field, keep the field in the target solution.

Field Length Is Reduced

If the field length is reduced in Microsoft Dynamics NAV 2013, you can make the new length of the field as large as it was before, or reduce the data by using a temporary table. Do this by using an algorithm to move the data to another field. On the fourth step, reduce the length of the field.

Record Does Not Exist

The code in the Upgrade Toolkit frequently retrieves a record that is based on the primary key of the table, without checking the return value by using an IF statement. Therefore, if the program cannot find a record while transferring data, the "Record does not exist" error occurs.

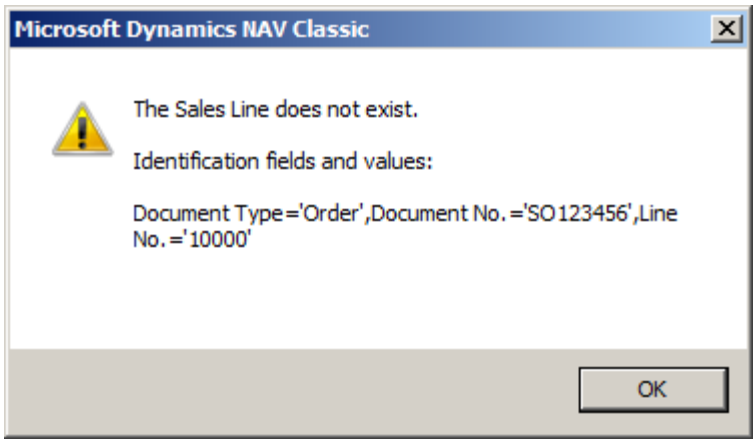

FIGURE 4.7:THE RECORD DOES NOT EXIST ERROR

If this error occurs, identify the location in the appropriate codeunit of the Upgrade Toolkit, and then rewrite the **Record.GET** function by using the IF statement, as shown in the following table.

Old Function	New Function
Customer.GET(”No.”);	IF Customer.GET(”No.”) THEN;
ItemLedgerEntry.GET(”Entry No.”);	IF ItemLedgerEntry.GET(”Entry No.”) THEN;

Table Common to All Companies Becomes Company-Specific

The customer may require that some information that is used by all companies in the database should be distributed into separate company-specific tables. Resolve this as follows:

1. Create a buffer table with fields that are identical to the source table.
2. During the first step of the data upgrade process, move the data to the buffer table.
3. Delete all data in the source table.
4. During the second step of the data upgrade process, copy the data from the buffer table to the original table, that is now company-specific.
5. Clear and delete the buffer table.

Customer Deletes a Standard Field

The Upgrade Toolkit contains data upgrade rules for many standard tables and fields. If the customer removed a standard field in the source solution, an error may occur.

Depending on the situation, you can resolve this error in either of the following ways:

- Find references to this field in the code of Upgrade Toolkit objects and delete (comment) them.
- Add the field to the initial database.

Lab: Data Upgrade to Microsoft Dynamics NAV 2013

Scenario

In this lab, you practice migrating data from a Microsoft Dynamics NAV 2009 database to Microsoft Dynamics NAV 2013 by using the Upgrade Toolkit.

Simon, the development consultant who is responsible for upgrading Microsoft Dynamics NAV 2009 for CRONUS International Ltd., finishes upgrading code and transforming objects. Now he must move data from the customer's database to the Microsoft Dynamics NAV 2013 database by using the Microsoft Dynamics NAV Upgrade Toolkit.

He must also save any modifications to the Upgrade Toolkit objects for future use during the live data upgrade.

Lab Setup

This lab does not depend on labs in previous modules.

Exercise 1: Prepare the Database for Data Upgrade

Exercise Scenario

Simon starts by preparing the database for the upgrade. He uses Microsoft Dynamics NAV 2009 Classic client to verify that data in the database is consistent and that there are no data integrity issues. Then he imports the objects for step 1 of the data upgrade process.

During the upgrade process, Simon identified a conflicting change in table 14, **Location**.

In the custom Microsoft Dynamics NAV 2009 database, there is field 50001, **Default Bin Code**, that is used by CRONUS to specify the default bin code for each location. Microsoft Dynamics NAV 2013 includes this field as a standard field 138, **Default Bin Code**.

To resolve the conflict, Simon customized the data upgrade objects to include migrating the data from the original customized field to the target standard field in Microsoft Dynamics NAV 2013.

Task 1: Open the Database in Microsoft Dynamics NAV 2009 Classic client

High Level Steps

1. Start Microsoft Dynamics NAV 2009 Classic client.
2. Open the existing CRONUS database.

Detailed Steps

1. Start Microsoft Dynamics NAV 2009 Classic client.
   a. Click **Start > All Programs > Microsoft Dynamics NAV 2009 Classic with Microsoft SQL Server**. Or, double-click Microsoft Dynamics NAV 2009 Classic with Microsoft SQL Server.

2. Open the existing CRONUS database.
   a. Click **File > Database > Open**.
   b. In the **Server Name** field, enter "NYC-SVR1".
   c. In the **Database Name** field, enter "Demo Database NAV (6-0)", or look up the database.
   d. Make sure that **Authentication** is set to Windows Authentication.
   e. Click **OK**.
   f. Press F12 to open a company.
   g. In the **Open Company** window, select CRONUS International Ltd., and then click **OK**.
   h. Confirm the message by clicking **OK**.

Task 2: Verify Data Consistency

High Level Steps

1. Test the database.
2. Test field relationships.

Detailed Steps

1. Test the database.
   a. Click **File > Database > Test**.
   b. Select the **Custom** option.
   c. Select all enabled check boxes except **Test field relationships between tables**.
   d. Click **OK**. The test progress is displayed. When the test is complete, a success dialog box appears.
   e. Click **OK**.

2. Test field relationships.
    a. Click **File > Database > Test**.
    b. Select the **Custom** option.
    c. Clear all check boxes, and then select **Test field relationships between tables**.
    d. Click **OK** to start the test.
    e. Click **OK** to confirm the message dialog box.

Task 3: Import Upgrade Step 1 Objects

High Level Steps

1. Import the Upgrade601700.1.fob file.

Detailed Steps

1. Import the Upgrade601700.1.fob file.
    a. Click **Tools > Object Designer**.
    b. Click **File > Import**.
    c. Browse to the X:\NAV 2013 Setup files\DVD_BUILD33781\UpgradeToolKit\Data Conversion Tools\601\Upgrade601700.1.fob file and then click Open. A warning dialog box informs you that there are conflicts.
    d. Click **OK** to open the Import Worksheet.
    e. Click **Replace All**, and then click **OK** to complete the import. When the import is complete, the **Import Objects** window displays the import statistics.
    f. Click **OK** to close the **Import Objects** window.

Note: *Use the same steps to import the X:\Allfiles\Mod04\Labfiles\Lab 4.A - Starter.fob file. This file contains the Microsoft Dynamics NAV 2009 customized objects.*

Task 4: Customize Upgrade Step 1 Objects

High Level Steps

1. Verify the customization in table 14, **Location**.
2. Create a buffer table to store information about default bin codes for locations.
3. Change the code in the codeunit 104045, **Upgrade Nav Old Version** to copy relevant data to the buffer table.
4. Save, close, and mark the object.
5. Export the customized objects for step 1.

Detailed Steps

1. Verify the customization in table 14, **Location**.
   a. In Object Designer, click **Table**.
   b. Select table 14, **Location**.
   c. Click **Run**.
   d. Make sure that field **Default Bin Code** is present. The field can be found at the end of the window, scroll to the right to see it.

2. Create a buffer table to store information about default bin codes for locations.
   a. Click **New** to create a new table.
   b. In the **Table Designer** window, enter the following information.

Field No.	Field Name	Data Type	Length
1	Code	Code	10
50001	Default Bin Code	Code	20

   c. Click **File > Save**.
   d. In the **Save As** dialog box, enter "123456701" in the **ID** field, and then enter "Temp Location" in the **Name** field.
   e. Make sure that the **Compiled** check box is selected, and then click **OK**.
   f. Close Table Designer.

3. Change the code in the codeunit 104045, **Upgrade Nav Old Version** to copy relevant data to the buffer table.
   a. In Object Designer, click **Codeunit**.
   b. Select codeunit 104045, **Upgrade Nav Old Version**.
   c. Click **Design**.
   d. Find the function trigger for the **UpdateLocation** function.
   e. Click **View > Locals**.
   f. On the **Variables** tab, define a new variable, name it TempLocation, set its type to Record, and its subtype to Temp Location.
   g. Close the **C/AL Locals** window.
   h. Replace the UpdateLocation function trigger code with the following C/AL code.

```
WITH Location DO
  IF StateIndicator.UpdateTable(TABLENAME) THEN BEGIN
    TimeLog.TimeLogInsert(TimeLog,TABLENAME,TRUE);
// CIUPG2013 >
//    IF NOT ISEMPTY THEN BEGIN
//      StateIndicator.Update;
//      MODIFYALL("Inbound BOM Bin Code",'');
//      MODIFYALL("Outbound BOM Bin Code",'');
    IF FINDSET(TRUE) THEN BEGIN
      REPEAT
        StateIndicator.Update;
        TempLocation.INIT;
        TempLocation.TRANSFERFIELDS(Location);
        TempLocation.INSERT;
      UNTIL Location.NEXT = 0;
      MODIFYALL("Inbound BOM Bin Code",'');
      MODIFYALL("Outbound BOM Bin Code",'');
      MODIFYALL("Default Bin Code",'');
// CIUPG2013 <
    END;
    TimeLog.TimeLogInsert(TimeLog,TABLENAME,FALSE);
    StateIndicator.EndUpdateTable(TABLENAME);
  END;
```


Note: *Do not delete and retype the complete trigger code. Only enter the changes to the code, that are marked with the CIUPG2013 comment.*

4. Save, close, and mark the object.
    a. Click **File > Save**.
    b. In the **Save** dialog box, make sure that the **Compiled** check box is selected.
    c. Click **OK**.
    d. Close the **C/AL Editor** window.
    e. In the Object Designer, set the **Version List** property for the codeunit 104045, Upgrade Nav Old Version to "UPGW17.00.00,CIUPG2013".

5. Export the customized objects for step 1.
    a. In Object Designer, click **All**.
    b. Select the **Version List** field for any of the rows, and then click **View > Field Filter**, or press F7.
    c. In the **Version List - Field Filter** window, enter "UPGW17.00.00*" in the text box, and then click **OK**.

Note: *This selects all objects that belong to the Upgrade Toolkit step 1.*

    d. Press CTRL+A to select all objects.
    e. Click **File > Export**.
    f. Browse to the original Data Upgrade step 1 object file path (X:\NAV 2013 Setup files\DVD_BUILD33781\UpgradeToolKit\Data Conversion Tools\601), enter "Upgrade601700.1.Custom.fob", and then click **Save**.

Note: *This saves the custom version of the upgrade objects for step 1. You can use this during pre-production or live data upgrade runs.*

Exercise 2: Run Data Upgrade Step 1

Exercise Scenario

Simon finishes the data upgrade step 1 by transferring the data from Microsoft Dynamics NAV 2009 tables to the buffer tables that are used during the data upgrade process.

Task 1: Transfer Data

High Level Steps

1. Run form 104001, **Upgrade - Old Version**.
2. Transfer data.
3. Review the data transfer status.

Detailed Steps

1. Run form 104001, **Upgrade - Old Version**.
   a. In Object Designer, click **Form**.
   b. Select form 104001, **Upgrade - Old Version**.
   c. Click **Run**.

2. Transfer data.
   a. Click **Transfer Data**. The progress bar indicates that the data upgrade is running.
   b. Wait until the data transfer is completed.

3. Review the data transfer status.
   a. Click **Status > State Indicator** to determine whether there are any errors.
   b. Close the **Upgrade State Indicator** form.
   c. Close the **Upgrade - Old Version** form.

Exercise 3: Prepare for Data Upgrade Step 2

Exercise Scenario

After the first step of the data upgrade successfully finishes, Simon prepares the database for the second step of the upgrade.

Task 1: Delete Objects

High Level Steps

1. Delete all non-table objects.

Detailed Steps

1. Delete all non-table objects.
   a. In Object Designer, click **Form**.
   b. Select form 104001, **Upgrade - Old Version**.
   c. Click **Run**.
   d. Click **Delete Objects**.
   e. In the confirmation dialog box that asks whether you want to delete the old objects, click **Yes**.
   f. When object deletion is completed, click **OK**.

Task 2: Create a New Database

High Level Steps

1. Create a new Microsoft Dynamics NAV 2013 database.
2. Uninstall Microsoft Dynamics NAV 2009.
3. Install Developer Option for Microsoft Dynamics NAV 2013.
4. Set the Database Compatibility Level for SQL Server 2012.
5. Set user permissions.
6. Open the new database in Microsoft Dynamics NAV 2013 Development Environment.
7. Upload the development license to the database.
8. Configure the Microsoft Dynamics NAV 2013 Server to connect to the new database.

Detailed Steps

1. Create a new Microsoft Dynamics NAV 2013 database.
    a. Close Microsoft Dynamics NAV 2009 Classic client.
    b. Click **OK** on any error messages that state that an object is missing.
    c. Click **Start > All Programs > Microsoft SQL Server 2012 > SQL Server Management Studio**.
    d. In the **Connect to Server** dialog box, make sure that Server Name is "NYC-SVR1", and then click **Connect**.
    e. In Object Explorer, expand **Databases**, and then right-click the Demo Database NAV (6-0) database.
    f. Click **Tasks > Backup**.
    g. In the **Backup Database** window, accept all default options, and then click **OK**.
    h. After the database backup finishes, click **OK**.
    i. Right-click the Demo Database NAV (6-0) database, and then click **Tasks > Restore > Database**.
    j. In the **Database** field under the **Destination** group, enter "Demo Database NAV (6-0) Copy".
    k. Click **Files**.
    l. In the **Restore As** field for the data file, enter "C:\Program Files\Microsoft SQL Server\MSSQL11.MSSQLSERVER\MSSQL\DATA\Demo Database NAV (6-0) Copy.mdf".
    m. In the **Restore As** field for the transaction log file, enter "C:\Program Files\Microsoft SQL Server\MSSQL11.MSSQLSERVER\MSSQL\DATA\Demo Database NAV (6-0) Copy_log.ldf".
    n. Click **Options**.

o. Clear the **Take tail-log backup before restore** check box.
p. Click **OK**.

2. Uninstall Microsoft Dynamics NAV 2009.
   a. Click **Start > Control Panel**.
   b. Click **Uninstall a program**.
   c. Select Microsoft Dynamics NAV 2009 SP1, and then click **Uninstall**.
   d. In the confirmation dialog box, click **Yes**.
   e. After the wizard finishes uninstallation, click **Close**.

3. Install Developer Option for Microsoft Dynamics NAV 2013.
   a. These steps are only required in case Microsoft Dynamics NAV 2013 is not installed.
   b. Browse to X:\NAV 2013 Setup files\DVD_BUILD33781.
   c. Double-click setup.exe to start the Microsoft Dynamics NAV 2013 installation wizard.
   d. Click **Next**.
   e. Click **I accept**.
   f. Click **Choose an installation option**.
   g. Under **Developer**, click **Customize**.
   h. Remove the following components:
      i. Portal Framework for SharePoint
      ii. Web Server Components
      iii. ClickOnce Installer Tools
   i. Click **Next**.
   j. Under the **SQL Database Components** group, clear the **SQL Server Instance** field.
   k. Click **Apply**.
   l. Wait until the installation wizard is finished.
   m. Click **Close**.

4. Set the Database Compatibility Level for SQL Server 2012.
   a. In SQL Server Management Studio, right-click the Demo Database (6-0) Copy database, and then click **Properties**. The **Database Properties** window opens.
   b. In the **Database Properties** window, click **Options** to show the **Options** page.
   c. In the **Compatibility level** field, select SQL Server 2012 (110).
   d. Click **OK**.

5. Set user permissions.
    a. In SQL Server Management Studio, under the Demo Database (6-0) Copy database, expand **Security > Users**.
    b. Right-click the NT AUTHORITY\NETWORK SERVICE user, and then click **Properties**. **The Database User** window opens.
    c. Click **Membership**.
    d. Select the db_owner role membership.
    e. Click **OK** to accept changes and close the **Database User** window.

6. Open the new database in Microsoft Dynamics NAV 2013 Development Environment.
    a. Click **Start > All Programs > Microsoft Dynamics NAV 2013 Development Environment**.
    b. Click **File > Database > Open**.
    c. In the **Open Database** dialog box, enter "NYC-SVR1" in the **Server Name** field, and then select Windows Authentication.
    d. Look up the **Database Name** field, then in the **Available Databases** window, select Demo Database (6-0) Copy, and then click **OK**.
    e. Click **OK** to open the database. A warning dialog box informs you that the database must be upgraded.
    f. Click **OK**. Another warning dialog asks you for another confirmation.
    g. Click **OK**.
    h. After the database upgrade process is complete, click **OK**.

7. Upload the development license to the database.
    a. Click **File > Database > Alter**.
    b. On the **Integration** tab, select the **Save license in the database** check box.
    c. Click **OK**.
    d. In the **Upload the License File** dialog box, browse to the X:\NAV 2013 Training License\NAV 2013 training demo_dev license.flf file, and then click **Open**.
8. Configure the Microsoft Dynamics NAV 2013 Server to connect to the new database.
    a. Click **Start > All Programs > Microsoft Dynamics NAV Administration**.
    b. Expand Microsoft Dynamics NAV (local) and select DynamicsNAV70.
    c. Click **Edit**.

d. In the **Database Name** field in the **General** FastTab, enter "Demo Database NAV (6-0) Copy".
e. Clear the Database Instance field in the General FastTab.
f. Click **Save**, and then click **OK** to confirm the message.
g. Select **Microsoft Dynamics NAV (local)**.
h. Right-click DynamicsNAV70 instance in the central pane, and then click **Restart**.
i. Click **OK** to confirm the restart.

Task 3: Import Customized Objects

High Level Steps

1. Import the customized Microsoft Dynamics NAV 2013 objects.
2. Compile all objects.

Detailed Steps

1. Import the customized Microsoft Dynamics NAV 2013 objects.
   a. In Microsoft Dynamics NAV 2013 Development Environment, click **Tools > Object Designer**.
   b. Click **File > Import**.
   c. Browse to the X:\Allfiles\Mod4\Labfiles\Objects.fob file, and then click **Open**, and then click **Yes**.

Note: *This file usually contains your custom Microsoft Dynamics NAV 2013 objects, with merged customizations. This file is the output of the code upgrade process.*

---

   d. Click **OK** to open the Import Worksheet.
   e. Click **Replace All**, and then click **OK** to complete the import.
   f. After the import is complete, click **OK** to close the **Import Objects** window.

2. Compile all objects.
   a. In Object Designer, click **All**.
   b. Click CTRL+A to select all objects
   c. Click **Tools > Compile**.
   d. Click **Yes** to confirm the compile operation.
   e. If the **Error List** window shows after compilation, close it.

Note: *Ignore any compilation errors.*

Exercise 4: Run Data Upgrade Step 2

Exercise Scenario

Simon imports the Data Upgrade objects for step 2. He customizes the objects to copy the data from the **Temp Location** table into the new standard field in the **Location** table. He then exports the customized objects for future use, and finishes the Data Upgrade step 2.

Task 1: Import Upgrade Step 2 Objects

High Level Steps

1. Import the Upgrade601700.2.fob file.

Detailed Steps

1. Import the Upgrade601700.2.fob file.
   a. In Object Designer, click **File > Import**.
   b. Browse to the X:\NAV 2013 Setup files\DVD_BUILD33781\UpgradeToolKit\Data Conversion Tools\601\Upgrade601700.2.fob file. A warning dialog box informs you that there are no conflicts.
   c. Click **Yes** to complete the import.
   d. Click **OK** to close the **Import Objects** window.

Task 2: Customize Upgrade Step 2 Objects

High Level Steps

1. Customize codeunit 104048, Upgrade New Version to move the data from the **Temp Location** table to the **Location** table.
2. Save, close, and mark the codeunit.
3. Export the customized objects for Step 2.

Detailed Steps

1. Customize codeunit 104048, Upgrade New Version to move the data from the **Temp Location** table to the **Location** table.
   a. In Object Designer, click **Codeunit**.
   b. Select codeunit 104048, Upgrade New Version, and then click **Design**.
   c. Click **View > C/AL Globals**.
   d. On the **Functions** tab, create a new function, and name it **UpdateLocations**.
   e. Close the **C/AL Globals** window.
   f. Scroll down to the UpdateLocations function trigger, and then click **View > C/AL Locals**.

g. On the **Parameters** tab, define a by-reference parameter named StateIndicator, with type Record, and subtype State Indicator.

h. On the **Variables** tab, define the following local variables.

Name	DataType	Subtype
Location	Record	Location
TempLocation	Record	Temp Location

i. In the function trigger for the **UpdateLocations** function, enter the following code.

```
// CIUPG2009 >

WITH TempLocation DO

 IF StateIndicator.UpdateTable(TABLENAME) THEN BEGIN

  TimeLog.TimeLogInsert(TimeLog,TABLENAME,TRUE);

  IF FINDSET THEN BEGIN

   REPEAT

    IF Location.GET(Code) THEN BEGIN

     Location."Default Bin Code" := "Default Bin Code";

     Location.MODIFY;

    END;

   UNTIL NEXT = 0;

   DELETEALL;

  END;

  TimeLog.TimeLogInsert(TimeLog,TABLENAME,FALSE);

  StateIndicator.EndUpdateTable(TABLENAME);

 END;

// CIUPG2009 <
```

j. At the end of the Upgrade function trigger, above the DeleteRemovedPermissions code line, enter the following C/AL code.

```
// CIUPG2009 >

UpdateLocations(StateIndicator);

// CIUPG2009 <
```

2. Save, close, and mark the codeunit.
   a. Click **File > Save**.
   b. In the **Save** dialog box, make sure that the **Compiled** check box is selected, and then click **OK**.
   c. Close the C/AL Editor.
   d. In the Object Designer, set the **Version List** property for the codeunit 104048, Upgrade New Version to "UPGW17.00.00,CIUPG2013"

3. Export the customized objects for Step 2.
   a. Select the codeunit 104048, Upgrade New Version.
   b. Click **File > Export**.
   c. Browse to the original Data Upgrade step 1 object file path (X:\NAV 2013 Setup files\DVD_BUILD33781\UpgradeToolKit\Data Conversion Tools\601), type "Upgrade601700.2.Additional.fob", and then click **Save**.

 ***Note:*** *This file does not replace the Upgrade601700.2.fob file. It is an addition to it.*

Task 3: Run Data Conversion

High Level Steps

1. Start Microsoft Dynamics NAV 2013 client for Windows.
2. Run data conversion.

Detailed Steps

1. Start Microsoft Dynamics NAV 2013 client for Windows.
   a. Click **Start > All Programs > Microsoft Dynamics NAV 2013**.
   b. Make sure that the client starts and that no errors occur.

 ***Note:*** *If the client starts and shows a Control add-in error in the role center, ignore the error.*

2. Run data conversion.
   a. Switch to Microsoft Dynamics NAV Development Environment.
   b. Run page 104002, **Upgrade - New Version**.

Note: *If the Microsoft Dynamics NAV Development Environment asks to open a company, select CRONUS International Ltd., and then click* ***OK****.*

c. Verify that the **SQL Server Name** field contains NYC-SVR1.
d. Click **Test Database Connection**.
e. Click **OK** to confirm the successful connection testing.
f. Click **Transfer Data**.
g. Click **Yes** to confirm the start of data upgrade step 2.
h. The client asks for permission. Select the **Allow for this client session** option, and then click **OK**.

Task 4: Initialize the Company

High Level Steps

1. Run codeunit 2, Company-Initialize.

Detailed Steps

1. Run codeunit 2, Company-Initialize.
   a. In Object Designer, click **Codeunit**.
   b. Select codeunit 2, Company-Initialize.
   c. Click **Run**.

Exercise 5: Complete Data Upgrade

Exercise Scenario

Simon finishes the data upgrade process by upgrading the data for all companies, deleting unused tables, and then testing the new database.

Task 1: Delete Unused Objects

High Level Steps

1. Delete unused tables from the database.
2. Delete the Upgrade Toolkit objects.

Detailed Steps

1. Delete unused tables from the database.
   a. In Object Designer click **Page**.
   b. Select page 104002, **Upgrade - New Version**, and then click **Run**.
   c. Click **Mark Unused Old Tables**.
   d. In the dialog box that asks you whether you want to mark the old unused tables, click **Yes**. You receive a message that old unused tables are marked for deletion.

e. Click **OK** to close the message box.
f. In Object Designer, set a filter on the **Version List** column to select tables marked as **Old Unused Table - marked for deletion**.
g. Press F4 to delete tables.
h. Click **Yes** to confirm the deletion.

2. Delete the Upgrade Toolkit objects.
   a. On the **Upgrade – New Version** page, click **Mark/Delete Upgrade Toolkit**. All Upgrade Toolkit objects, except tables, are removed automatically. The tables are marked for deletion.
   b. In Object Designer, set a filter on the **Version List** column to select tables that are marked as **Upgrade Toolkit Table - marked for deletion**.
   c. Press F4 to delete tables.
   d. Click **Yes** to confirm the deletion.

Summary

The data upgrade phase of the upgrade project is intended to transfer the information from the customer's database to the target version of the solution. It touches only the table objects and the information that they store.

Data upgrade is performed in three major stages:

- Trial data upgrade that happens on the development environment
- Pre-production data upgrade that is performed at the pre-production environment at the customer's site
- Live data upgrade that is performed at the actual production environment at the customer's site at the final phase of the upgrade project.

During trial data upgrade, you use the Microsoft Dynamics NAV Upgrade Toolkit to achieve the following goals:

- Convert the information about the source version to be compatible with the target's version architecture.
- Resolve issues that arise when transferring the information from source version tables to temporary tables and then to target version tables.
- Delete unused tables.
- Run field checks and introduce necessary changes in data/objects.

After data upgrade is complete, you perform data acceptance testing and functional testing to make sure that the information in the upgraded database corresponds to the initial data, and that the users can access data correctly according to their daily tasks.

Data upgrade troubleshooting is an important part of the data upgrade process. It helps the project team register the complete list of actions that are required to perform smooth live data upgrade at the production environment by using the previously modified Upgrade Toolkit objects.

Trial data upgrade results in a step-by-step description of all actions that are required to upgrade the customer's database to the target version and .fob files that contain the customized Upgrade Toolkit objects.

Module Review

Test Your Knowledge

Test your knowledge with the following questions.

1. Data upgrade is also known as data conversion or data migration.

   ( ) True

   ( ) False

2. Which of the following objects do you export to separate .fob files during the trial data upgrade for use during the pre-production and live upgrades?

   ( ) Upgraded tables

   ( ) Modified Upgrade Toolkit objects for step 1 and step 2

   ( ) Upgraded codeunits

   ( ) Application Test Toolset objects from PartnerSource

3. Which client do you use to perform step 1 of the data upgrade?

   ( ) Microsoft Dynamics NAV 2009 RoleTailored client

   ( ) Microsoft Dynamics NAV 2013 client for Windows

   ( ) Microsoft Dynamics NAV 2009 Classic client

   ( ) Microsoft Dynamics NAV 2013 Development Environment

4. When you perform data acceptance testing, the project team verifies that the upgraded solution functions correctly against the required functionality.

   ( ) True

   ( ) False

5. Name the possible ways to resolve the following issue: a field that is removed in the target solution contains data in the source database.

Test Your Knowledge Solutions

Module Review and Takeaways

1. Data upgrade is also known as data conversion or data migration.

   (√) True

   ( ) False

2. Which of the following objects do you export to separate .fob files during the trial data upgrade for use during the pre-production and live upgrades?

   ( ) Upgraded tables

   (√) Modified Upgrade Toolkit objects for step 1 and step 2

   ( ) Upgraded codeunits

   ( ) Application Test Toolset objects from PartnerSource

3. Which client do you use to perform step 1 of the data upgrade?

   ( ) Microsoft Dynamics NAV 2009 RoleTailored client

   ( ) Microsoft Dynamics NAV 2013 client for Windows

   (√) Microsoft Dynamics NAV 2009 Classic client

   ( ) Microsoft Dynamics NAV 2013 Development Environment

4. When you perform data acceptance testing, the project team verifies that the upgraded solution functions correctly against the required functionality.

   ( ) True

   (√) False

5. Name the possible ways to resolve the following issue: a field that is removed in the target solution contains data in the source database.

   MODEL ANSWER:

   This issue can be resolved in any of the following ways, depending on the situation:

   - By moving the data to a similar field
   - By clearing the values in this field
   - By keeping the field in the target solution.