MODULE 2: PROJECT ANALYSIS AND PLANNING

Module Overview

This module covers the main activities of the upgrade analysis and planning process. It describes best practices for the following:

- Collecting requirements for the solution
- Estimating the upgrade
- Defining the test upgrade strategy
- Planning the upgrade sequence based on the analysis that is performed.

In addition, this module explains how to set up environments for the upgrade.

Objectives

- Define the requirements and scope for the upgrade.
- Explain how to estimate the upgrade project and define the test upgrade strategy.
- Review planning of the upgrade sequence and workflow.
- Explain preparing environments for the upgrade.

Scenario

The Cronus International Ltd. company uses Microsoft Dynamics NAV 2009 SP1 to handle their financial affairs. They also use their own custom solution, which is an extension of the Financial Management module. This custom solution requires additional settings and specific workflows. The company plans to upgrade their Microsoft Dynamics NAV 2009 SP1 to Microsoft Dynamics NAV 2013.

Cronus International Ltd. turned to their Microsoft Certified Partner to help them with the upgrade.

Simon, the project manager from the Microsoft Certified Partner side, is selected to be responsible for this upgrade. Simon selects the following team members who will take part in the project:

- Jack, the application consultant, will help Simon define the project scope.
- Victor, the solution architect, will help Simon prepare plans and design the overall solution.
- Adam, the development consultant, will be responsible for developing and implementing the upgrade.
- Linda, the test lead, will plan and execute the test strategy.
- Sean, the technology consultant, will set up the upgrade environments.

Simon then explains the basic principles of upgrading to the Cronus International Ltd team. He summarizes the upgrade workflow, tools, and methods. In addition, he makes a point on the upgrade testing strategy.

After this basic introduction, the upgrade team starts the Diagnostic and Analysis phases of the upgrade project. After they investigate the system that will be upgraded, the upgrade team defines the scope for the upgrade and estimates the work. Simon and Victor plan the upgrade sequence and send the estimation and plan to Cronus International Ltd. for approval.

Project Analysis and Planning Overview

Project Analysis and Planning Overview

According to the Microsoft Dynamics Sure Step methodology, an upgrade project starts with the diagnosis and analysis of the system that is to be upgraded, and the planning of the activities that the project team will perform. It is one of the most important phases in the upgrade process, as correct planning, identification of business processes, and scoping improve the possibility of success for the project.

The project analysis and planning activities are as follows:

- Define requirements for the target solution.
- Estimate all upgrade actions.
- Define the upgrade testing strategy.
- Develop a comprehensive upgrade workflow plan.
- Establish the environments for supporting the upgrade process.

When the Analysis and Design phases are complete, the project team must arrange the following documents:

- Finalized Project Charter
- Detailed Upgrade Project Plan

Collecting Business Requirements

To define the requirements for the upgrade project, you must carefully examine the source application to determine the customizations and localizations that the upgrade project contains such as whether it has any ISV solutions, and so on. Then the project team can develop a thorough understanding of what exactly should be upgraded and in what way.

The two main activities that help define the upgrade requirements are source solution analysis and defining the project scope.

An additional way to better understand the customer solution is to conduct a solution overview. This is a presentation of the functionality to the upgrade team. The overview highlights the differences between the customizations and standard functionality, and then describes the solution's capabilities.

Additionally, if the customer wants to create additional functionality for the existing customization, you can conduct a Fit-Gap Analysis. In this training manual’s scenario, no additional functionality is required.

A good practice is to discuss with the customer what new functionality was added to the Microsoft Dynamics NAV versions since Microsoft Dynamics NAV 2009 SP1. Then you can help the customer create an opportunity to add more users to the license as part of the upgrade project.

During the discussion, you can collect the key requirements for future post-upgrade user training. This training introduces the users at the customer's site to the differences between the old functionality and features of the new solution. You must also include the planned time for these training sessions in project estimates.

What's New: Developer and IT Pro Changes for Microsoft Dynamics NAV 2013

http://go.microsoft.com/fwlink/?LinkId=277013

What's New: Application Changes for Microsoft Dynamics NAV 2013

http://go.microsoft.com/fwlink/?prd=12036&pver=2013&plcid=0x409&clcid=0x409&ar=Upgrade&sar=1.2

Source Solution Analysis

The key to developing accurate upgrade project requirements is to perform a thorough and accurate analysis of the source solution. This includes the following tasks that define the strategy for the code and data upgrade with later testing:

- Review of all existing technical and user documentation for the solution. This documentation helps define the scope of objects for the upgrade, resolve code conflicts, and test the upgraded solution. The following are examples of documentation:
  - Functional or technical specification designs
  - Test cases
  - Business process descriptions
  - User’s guides
  - Installation guides
  - Release notes
- Investigate the business processes of the company.
- Determine the parts of the solution that are still required and the parts that must be removed.
- Determine whether the functionality of the old customization is covered by the new version's standard functionality.

- Determine the data that can be removed or archived during the data upgrade.
- Consider postponing the whole upgrade if the latest version of the ISV add-on is not available.
- Decide whether the solution must be optimized for better performance.
- Benchmark the whole database or specific areas to define the list of objects and areas that must be optimized. For Dynamics NAV 2009 and earlier versions, you can use Application Benchmark Tool for Microsoft Dynamics NAV. For Dynamics NAV 2013, you can use Application Test Toolset.

Define the Project Scope

Project scope is a set of activities that is agreed upon with the customer according to the contract. The project scope outlines the project deliverables and describes how the project will be conducted. If there is an upgrade project, its scope includes the type of upgrade, amount of modifications, and so on.

The best way to determine how the level of customizations in the customer's source solution affects the type of upgrade is to use the decision tree diagram. The Define the Project Scope figure shows the different decisions that are involved in upgrading from Microsoft Dynamics NAV 2009 SP1 to Microsoft Dynamics NAV 2013.

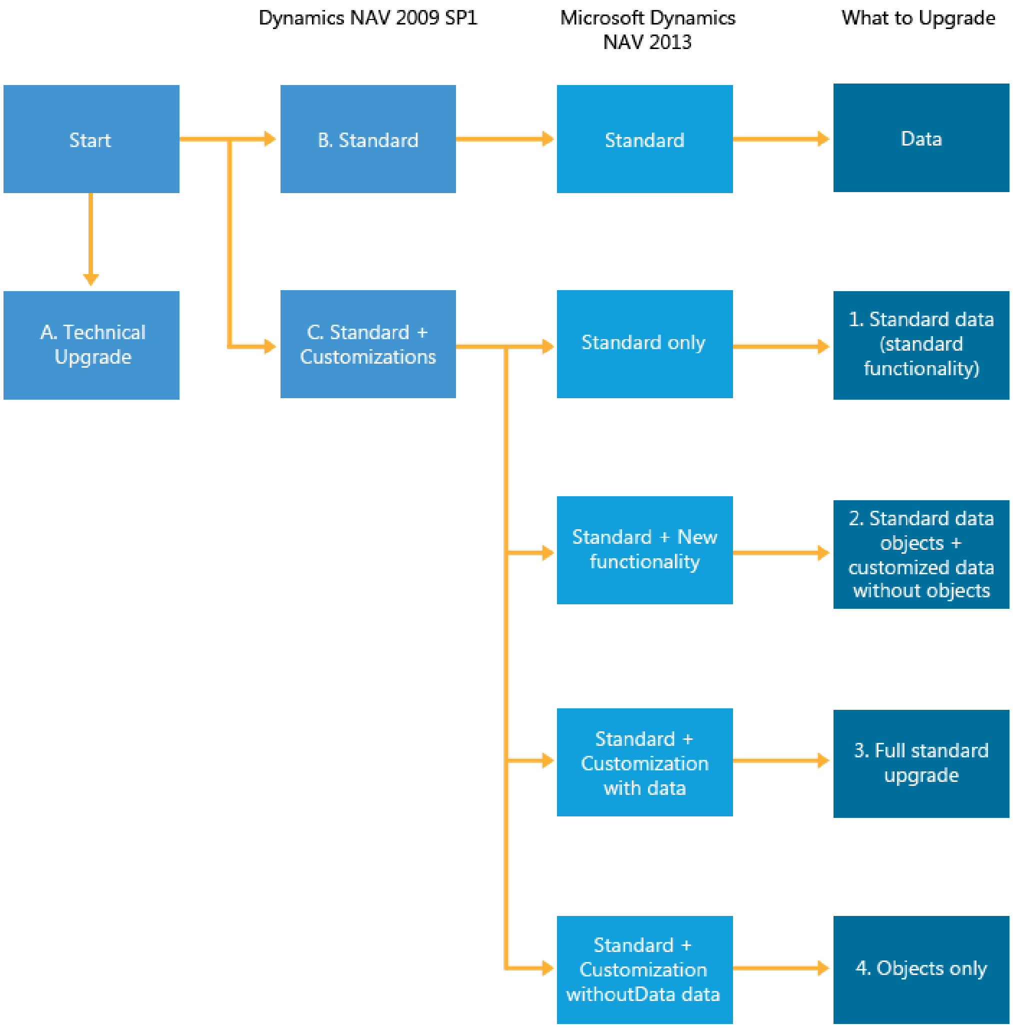

Dynamics NAV 2009 SP1
Microsoft Dynamics NAV 2013
What to Upgrade
Start
B. Standard
Standard
Data
A. Technical Upgrade
C. Standard + Customizations
Standard only
1. Standard data (standard functionality)
Standard + New functionality
2. Standard data objects + customized data without objects
Standard + Customization with data
3. Full standard upgrade
Standard + Customization withoutData data
4. Objects only

FIGURE 2.1:DEFINE THE PROJECT SCOPE

For information about upgrade paths and documentation about upgrades from earlier versions of Microsoft Dynamics NAV, refer to the MSDN website and documentation in the Upgrade Toolkit of the earlier versions.

More information can be found at the following locations.

Install and Upgrade Instructions for Microsoft Dynamics NAV 2009 R2

http://go.microsoft.com/fwlink/?LinkId=277014

Install and Configure Microsoft Dynamics NAV 2009 SP1

http://go.microsoft.com/fwlink/?LinkId=277015

Country/Region-Specific Information

http://go.microsoft.com/fwlink/?LinkId=277016

Transforming Forms to Pages

http://go.microsoft.com/fwlink/?LinkId=277017

Technical Upgrade

An upgrade can involve only upgrading the client in response to a bug or to improve performance. Although this is not generally considered best practice, there are situations where this type of upgrade are used, for example, when you migrate to the newest version to take advantage of improvements in C/SIDE.

Note: *Because of multiple changes between Microsoft Dynamics NAV 2009 and Microsoft Dynamics NAV 2013 (including classic stack removal, redesign of codeunit 1, Application Management, removal of server-side automation, and so on), we do not recommend a technical upgrade. In fact, it is not considered a best practice and such an environment is not supported by Microsoft Support.*

Standard Upgrade

If the customer uses only the standard Microsoft Dynamics NAV functionality with standard objects in the database (no objects are modified), you can perform only data migration by using Microsoft Dynamics NAV Upgrade Toolkit.

Similarly, if a customer uses standard Microsoft Dynamics NAV with an ISV solution that has no modified objects, only data migration is required. You must upgrade the ISV solution by using the respective upgrade toolkit.

Standard + Customizations

If there are customizations involved, the following scenarios are possible:

- The customer may have a standard system with customizations that are no longer relevant, and therefore must not be upgraded. In this case, only an upgrade of standard data is required (similar to a standard upgrade). If some data is based on the old customizations, it is deleted during the upgrade.

- The functionality of the old customizations may be covered by the new version’s standard functionality. In this case, standard upgrade procedure applies, and data that is used by the old customizations is transferred to respective new standard tables. Customized objects are not transferred to the new version because they are replaced by the standard objects. You must change any objects that reference the old customizations to use the new standard functionality.
- There may be customizations within base application objects. In this case, you must create a set of modified objects that are based on the new version and incorporate the necessary modifications. Before you do this, you must investigate the purpose and intent of the modifications. Then rethink and re-engineer them to fit and use the new base application and its features and functions. Such analysis must be a part of the upgrade plan and the quote.
- The customer may want to upgrade from an old version with customizations to a newer version with customizations, but may decide not to transfer historical transactions. For example, ledger entries can be compressed, but master data (customer, vendor, item and so on) must be transferred. In this case, the standard upgrade procedure applies to objects only.

Consider the following when you are defining project scope:

- **Target build version** – Generally, an upgrade is targeted to the latest version. However, the customer may require a specific version with a particular hot fix or localization.
- **Upgrade goal** – For evaluation of results at the end of the project, define whether the upgrade is needed to improve usability, add new features, or other purposes.
- Customer's expectations and budget capacity
- **Upgrade versus new implementation** – In some cases, the new version’s standard functionality may cover customizations in the source solution or an ISV solution that works with it. Therefore, it may be more efficient to implement the new solution instead of upgrading the current one.

You can gain a good indication of a project’s scope by examining the number of customizations that were made to the original version. The more changes to the original version, the more time that will be required for the upgrade.

Upgrade Estimates

Estimates provide information to calculate the budget and identify the resources and time that is required to perform the work. To give the customer a realistic quote, you must estimate the time and resources that you must have to upgrade the solution with an account of all phases of the project.

The customer must have a clear picture of what parts, phases, and processes the upgrade project has and what they can expect to receive, depending on the project scope when the upgrade is complete. The customer must be aware that the budget and time span of the full solution upgrade is defined by the number of resources and the estimated time that is spent on the following aspects:

- Code upgrade, object transformation, data upgrade, and testing for accuracy of the upgraded application and data
- Actual implementation of the upgraded solution on the customer’s site that has live data upgrade and additional acceptance testing that is performed by the customer's users

It is useful to record all estimations for time and for costs in a comprehensive document called the *Upgrade Costing Worksheet*, and to approve it with the customer. You can find this document in the Microsoft Dynamics Sure Step methodology.

Code Upgrade

The primary way to define estimates for the code upgrade is to use a *file compare tool*. This kind of tool offers facilities to compare different versions of objects, display the comparison results in general numbers, and highlight particular objects that caused conflicts.

You can use a tool such as WinDiff for this task.

Information about the WinDiff utility is available at the following website.

How to Use the Windiff.exe Utility

http://go.microsoft.com/fwlink/?LinkId=277018

The WinDiff.exe utility graphically shows the differences between ASCII text files that you specify, or the difference between folders that contain ASCII text files. This utility is especially useful for comparing program source code. You can use the WinDiff.exe to compare whole subfolder trees. The display shows either a summary of the comparison status of a list of files (outline mode) or a detailed line-by-line comparison of the files (expanded mode).

There are other tools and ISV solutions that you can use to compare objects. To obtain a list of specialized ISV solutions that you can use, go the following website.

Microsoft Pinpoint

http://go.microsoft.com/fwlink/?LinkId=277019

These tools are usually more extensive and can contain features such as the following:

- File Compare
- Folder Compare
- Version Control
- Conflict detection and resolution
- Automatic intelligent merging of files and folders in a three-way comparison
- In-place editing of files
- Track changes
- Code Analysis

In addition to estimating the code upgrade by using WinDiff, consider the following test activities when you prepare the estimates:

- **Code review** – Usually, it takes an average of 40 percent less time to perform code review than it takes developers to merge the objects. In other words, the estimated hours for code review are equal to 60 percent of estimated development hours. Only standard objects that are changed by the customer or partner must be estimated during code review.
- **User interface testing** – When estimating user interface testing, you must consider the total number of new customer/partner objects, in addition to that of standard pages, reports, and XMLports that were changed by the customer or partner. Estimates for user interface testing are prepared, depending on the average time that is spent on each type of object.
- **Functional testing** – When you are estimating functional testing, you must define the list of standard Dynamics NAV areas and features that are changed by the customer or partner. In addition, you must also define the approximate number of cases to execute for each area and feature, depending on the objects to be merged.

Additionally, you must have a clearly defined scope for the functional testing of the customer's solution. You can do this with the help of any technical documentation, such as the following:

- User guides
- Manuals
- Test scripts and cases that are provided by customer/partner

The “Estimating Code Upgrade” demonstration shows how to use WinDiff to estimate code upgrade.

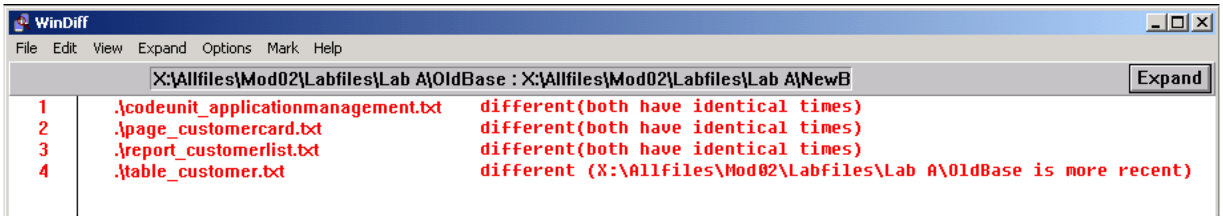
Demonstration: Estimate Code Upgrades

To estimate code upgrade by using WinDiff, perform the following procedure.

Demonstration Steps

1. Start WinDiff.
   a. Click **Start**.
   b. Type “WinDiff” in the **Search programs and files** text box, and then click **Enter.** The **WinDiff** window opens.

2. Compare folders.
   a. In WinDiff, select **File**, **Compare Directories.** The **Select Directories** window opens.
      i. In **Dir 1,** type the path of the OldBase folder: “X:Allfiles\Mod02\Labfiles\Lab A\OldBase”.
      ii. In **Dir 2,** type the path of the NewBase folder: “X:Allfiles\Mod02\Labfiles\Lab A\NewBase”, and then select **OK.**
      iii. WinDiff now shows the differences between the files as shown in the “WinDiff Compare Two Directories” figure.



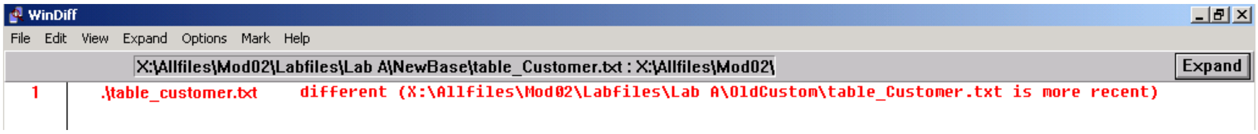
```
X:\Allfiles\Mod02\Labfiles\Lab A\OldBase : X:\Allfiles\Mod02\Labfiles\Lab A\NewB
1   .\codeunit_applicationmanagement.txt   different(both have identical times)
2   .\page_customercard.txt                different(both have identical times)
3   .\report_customerlist.txt              different(both have identical times)
4   .\table_customer.txt                   different (X:\Allfiles\Mod02\Labfiles\Lab A\OldBase is more recent)
```

FIGURE 2.2:WINDIFF COMPARE TWO DIRECTORIES

iv. By comparing the OldBase and NewBase files, you can view the number of objects that were modified between the Old Base (Microsoft Dynamics NAV 2009) and New Base (Microsoft Dynamics NAV 2013) version of Microsoft Dynamics NAV.

Repeat steps i. to iv. for the folders OldBase and OldCustom. Then you can view the number of objects that were customized in the OldCustom version.

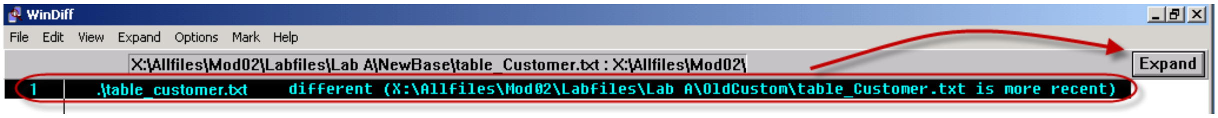
3. Import the files to compare.
   a. In WinDiff, select **File**, **Compare Files**. The **Select First File** window opens.
   b. Browse to the "X:Allfiles\Mod02\Labfiles\Lab A\OldCustom" folder where you find the first file that is named Microsoft table_Customer.txt., and then select **Open**. The file is imported. The **Select Second File** window opens.
   c. Browse to the "X:Allfiles\Mod02\Labfiles\Lab A\NewBase" folder and find the file that is named table_Customer.txt.,and then select **Open**. WinDiff now shows that there are differences between the two files as shown in the "WinDiff Compare Two Files" figure.



```
X:\Allfiles\Mod02\Labfiles\Lab A\NewBase\table_Customer.txt : X:\Allfiles\Mod02\
1   .\table_customer.txt     different (X:\Allfiles\Mod02\Labfiles\Lab A\OldCustom\table_Customer.txt is more recent)
```

FIGURE 2.3:WINDIFF COMPARE TWO FILES

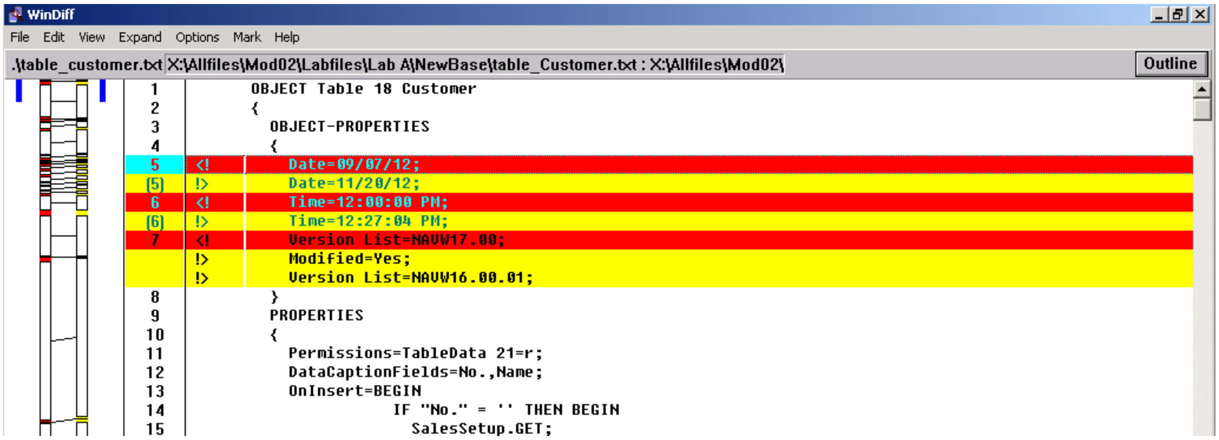
4. Compare the files in detail.
   a. In the WinDiff application, select the line, and then click **Expand** as shown in the "WinDiff Click Expand" figure.



```
X:\Allfiles\Mod02\Labfiles\Lab A\NewBase\table_Customer.txt : X:\Allfiles\Mod02\
1   .\table_customer.txt     different (X:\Allfiles\Mod02\Labfiles\Lab A\OldCustom\table_Customer.txt is more recent)
```

FIGURE 2.4:WINDIFF CLICK EXPAND

   b. WinDiff now displays the two files that have the differences shown as colors in the "WinDiff Expanded View" figure.



.\table_customer.txt X:\Allfiles\Mod02\Labfiles\Lab A\NewBase\table_Customer.txt : X:\Allfiles\Mod02\

```
  1         OBJECT Table 18 Customer
  2         {
  3           OBJECT-PROPERTIES
  4           {
  5    <!       Date=09/07/12;
 (5)   !>       Date=11/20/12;
  6    <!       Time=12:00:00 PM;
 (6)   !>       Time=12:27:04 PM;
  7    <!       Version List=NAVW17.00;
       !>       Modified=Yes;
       !>       Version List=NAVW16.00.01;
  8           }
  9           PROPERTIES
 10           {
 11             Permissions=TableData 21=r;
 12             DataCaptionFields=No.,Name;
 13             OnInsert=BEGIN
 14                        IF "No." = '' THEN BEGIN
 15                          SalesSetup.GET;
```

FIGURE 2.5:WINDIFF EXPANDED VIEW

c. Scroll down or use the menu in WinDiff to find the differences as shown in the "WinDiff View Menu" figure.

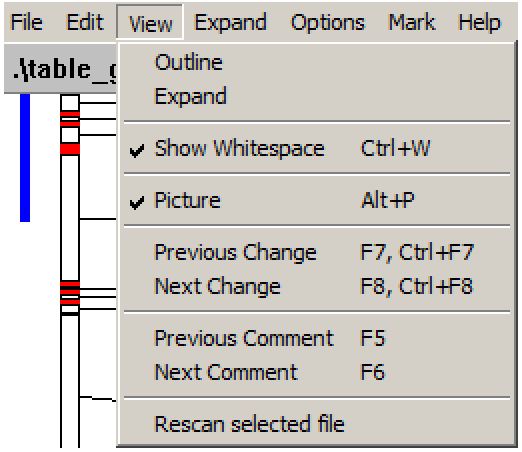


FIGURE 2.6:WINDIFF VIEW MENU

d. To determine the changes for table Customer between the OldBase and OldCustom version, you must repeat the process for the files table_Customer.txt in the OldBase and OldCustom folder.

Best Practice: *It is important to perform a three-way comparison between OldBase and OldCustom and OldBase and NewBase so that you can detect the customizations that were added for the customer in the OldCustom version, and the changes that were made by Microsoft between the OldBase and New base versions of Microsoft Dynamics NAV.*

Report Transformation

In Microsoft Dynamics NAV 2009, reports could have a Classic Layout in the sections and also contain a Report Definition Language for the Client (RDLC 2005) layout that was created in Visual Studio. In Microsoft Dynamics NAV 2013, reports only have an RDLC 2008 layout.

More information is available in the following website.

Report Upgrade

http://go.microsoft.com/fwlink/?LinkId=277020

To upgrade the customer's functionality so that it is in compliance with the new report architecture, you must transform the new customer report objects and retransform standard customized report objects. To measure the work that is required for report transformation, the project team and the customer must agree on which reports to transform: all new customer and standard changed reports, or specific reports that are requested by the customer.

Then you must research how to transform each object report type.

Microsoft Dynamics NAV 2013 has several ways to visualize data. For example, for simple reports, you can export data into Excel and then build a layout there, for example, on a second tab. Then you continue to update the data on the first tab by using the Microsoft Dynamics NAV Excel Add-in to update the report.

You can use Accounts Schedules and Sales Analyses reports and specific charts that are built in addition to those features. Or you can use generic charts that you can build on almost any table.

You can also build a query object by preparing the dataset that you must have for a report. Then you can expose it as an OData web service that can be consumed by other reporting solutions that are currently available.

You should understand the reports that the customer actually uses to understand whether all reports must be upgraded.

To estimate the time that you must have for report transformation, you must divide the transformed reports into the following categories:

- **Document reports** - These are the most time-consuming reports. An example of document reports is Report 202 Sales Document – Test.
- **List-type reports that use complex data item structure, grouping, and calculation of totals** – These reports require less time than document reports, because typically they are processed correctly according to basic transformation rules.

An example of such reports is Report 130 EC Sales List.

List-type reports that have simple grouping - In most cases, these reports process correctly according to basic transformation rules. Therefore, only minor changes may be necessary.

An example of such reports is Report 1 Chart of Accounts. This report is an ideal candidate for an export to Microsoft Excel.

The time that you must prepare, transform, and test the report differs, depends on the report type.

***Note:** It is not always a question of only transforming report layout. It is also important to consider performance. Where it is possible, you should review and restructure list reports, some document reports, and all large reports that process a potentially large volume of data for better performance.*

Multiple standard application reports were changed in Microsoft Dynamics NAV 2013 to take this into consideration (for example report 1001 Inventory Valuation).

Data Upgrade

Upgrade of customer data is one of the most time-consuming tasks in the upgrade project. Its complexity and time consumption depend on the composition and size of the database, the amount of data to upgrade, and types of database servers. Moreover, the data upgrade usually requires significant computing power.

The data upgrade process can be divided into the following phases:

- Trial data upgrade (verifying and benchmarking)
- Testing of data upgrade results
- Pre-production data upgrade
- Acceptance testing
- Live data upgrade (final data upgrade to production)

There is no fixed formula for estimating data upgrade. However, you can use the following guidelines:

1. Set up a test environment by using the production database backup and check the size of the database. Check the number of companies that the database contains; by skipping some companies, you can reduce the estimates.

2. Run database integrity tests. Use the standard Database Test window in the Development Environment to test the database for everything except table/field relationships. Then repeat the test and test only for table/field relationships. If any errors occur, you must also account for the time that is required to resolve data inconsistency or table relations, because it may not possible to back up a corrupted database.
3. After the trial data upgrade, create a document that contains the step-by-step script of the upgrade procedure to prevent any errors during the live data upgrade. To create such a script is a time-consuming process.
4. If the estimated time frame that was established after you performed the trial data upgrade and created the script is too large, analyze whether the database contains any obsolete data that can be compressed or truncated without any harm.
   a. Start by analyzing the size of tables. Identify the 20 largest tables.
   b. Identify the business functionality that is related to the data in these tables. Based on this, identify data that can be archived or compressed. Typically, the best candidates for archiving or compression are ledger entries.
   c. Database cleanup is a challenging and time-consuming process and cannot be done without close cooperation with the customer.
5. Include hours for testing data upgrade results (including data acceptance and functional testing), depending on project scope and customer/partner preferences.
   a. To help you estimate the data upgrade process, Microsoft Dynamics Sure Step contains the Data Upgrade Checklist document.

Results of the trial data upgrade define the complexity and time that you must have for the pre-production and live data upgrade.

Define Upgrade Test Strategy

Successful and comprehensive testing of the upgraded application and upgraded data is very important to building customer acceptance of the Microsoft Dynamics NAV solution. The upgraded solution must include all the needed ISV add-ons and customizations, in addition to all available data in the source database.

Define Upgrade Test Strategy

To define *test upgrade strategy* means defining the scope of testing, all testing objectives, and expected test activities to make sure of the following:

- The business solution environment (application and data) is stable.
- The introduced functionality corresponds to the customer's business processes.
- The resulting solution complies with the customer’s internal testing and quality standards.

Defining test upgrade strategy and test planning are initiated in the Analysis Phase by establishing a high-level plan for testing of the upgraded application and data. The Upgrade Test Plan document includes the list of test scripts, cases, and scenarios to show exactly what will be tested. The Upgrade Test Plan must consider both technical and functional aspects of the upgrade.

Note: *It might be relevant to identify the set of automated test scripts, for example, the most basic scenarios in the application. Then develop test scripts for those scenarios in Microsoft Dynamics NAV 2009 and then run them in Microsoft Dynamics NAV 2013 after the upgrade.*

Microsoft developed several test scripts that are suitable for reuse. These test scripts are included for both Microsoft Dynamics NAV 2009 and Microsoft Dynamics NAV 2013.

More information about test scripts for Dynamics NAV is available at the following website.

Application Test Toolset for Microsoft Dynamics NAV 2013

http://go.microsoft.com/fwlink/?LinkId=277021

According to the Microsoft Dynamics Sure Step methodology, the Upgrade Test Plan document may include the following areas.

- Overall test objectives and goals
- Test objectives and goals for each iteration
- Test approach and assumptions
- Testing responsibilities
- Quality and test scope
- Expected test results
- Test tasks and deliverables
- Definition of interactions with other organizations

- Testing procedures and walkthrough
- Test status tracking and reporting
- Test environment and resource requirements
- Test schedules
- Definition of test specification (script) template

The Upgrade Test Plan document also covers the test activities during the following testing stages:

- Testing code upgrade
- Object transformation testing
- Testing data upgrade results
- Acceptance testing

The Upgrade Test Plan document outlines the following types of testing that should be performed during each previously mentioned stage:

- **Code Review** - This test makes sure that all code is transferred correctly by developers and that no functionality is lost during the code upgrade. The code review procedure is handled by testers who use a special comparison tool.
- **User Interface Testing** – This test verifies that the mainstream functionality (standard and custom) runs without any unexpected error messages, for example, error messages about missing active keys, incorrect sorting order, and so on.

This type of testing determines whether the program runs and its interfaces are available and responsive. In other words, it touches all areas of the application that were changed by the customer, and answers the following basic questions:

  - Is it possible to start the test item at all?
  - Do the buttons and menu items on the pages trigger the correct actions or open the requested windows?

- **Testing of Transformed Objects** – This test makes sure that each object is transformed correctly, that the RoleTailored client principles do not conflict with the customer's solution, and that no functionality is lost during the transformation.
- **Functional Testing** – This test includes the stand-alone testing of selected features that had modifications (custom code) in the source version, features that are introduced in the target version, or a combination of both.

The scope of functional tests can vary from a particular feature in the system to a separate module in the product, depending on the project and scope of the upgrade. Testers perform functional testing during the Development phase. The customer and the consultants can also perform this functional testing during project acceptance.

- **Data Acceptance Testing** – This test verifies the accuracy of the upgraded data as tested by application consultants and the customer. It is important not only to verify the migrated data, but also to validate that the data can be inquired, reported, and transacted upon. Therefore, the testing team must define appropriate testing methods to test the accuracy of upgraded data.
- **Performance Testing** – This test verifies that performance of the upgraded system (business processes and integration) under high load, such as large transaction volume, meets user needs .
- **Stress Testing** - This test determines the capacity of the upgraded application to withstand the expected system load for number of transactions, number of concurrent users, batch processing, and so on.
- **Integration Testing** – This test includes testing of integration points between the upgraded Microsoft Dynamics NAV solution and other systems.
- **End-to-End Scenario Testing (User Acceptance Testing)** – This test runs end-to-end scenarios that involve multiple modules in the product to make sure that the upgraded solution works to the customer's satisfaction.

You may require customer involvement to execute end-to-end scenario testing. You must develop test cases and scripts to support the testing effort. The test cases and scripts must include business scenarios that are typically performed by the end-to-end users. The customer or the test team can develop these test cases and scripts with help of the customer and consultants.

Plan Upgrade Workflow Based on Analysis

Before you start to plan upgrade workflow, you must finish the following activities of the Diagnostic and Analysis phases:

- Define the project strategy and the project scope.
- Obtain and study the customer's business requirements.
- Complete the source solution overview and analysis.

When you plan upgrade project phases, it means that you must prepare the high-level project documentation that describes the following:

- Scope
- Deliverables
- Skills requirements
- Dependencies
- Activities
- Time lines
- Budget restrictions

Usually, these tasks are the subject of the Project Charter document that governs the whole engagement, and the Project Plan that is a key instrument for relaying status and progress throughout the life cycle of an upgrade project.

The Planning phase is mainly driven by the Project Manager.

Analysis and Design

Start to plan the upgrade workflow by planning the remaining activities of the Analysis phase. This creates the high-level project plan and accumulates the details of the upgrade process. At the same time, you can establish environments for analyzing customizations in the source system.

The following activities define the details of the upgrade process:

- Create a *Work Breakdown Structure*. This is a summary of the high-level scope that is based on the requirements and that is documented in the previous Diagnostic and Analysis phases. It is good practice to note the out-of-scope tasks.
- Define the roles that are required for this project and the skills that you must have to perform the project tasks. You must include the training that is required to perform the required tasks.
- Evaluate the timeframe that is needed to execute the implementation based on the estimation. This includes any dependencies that are owned by the customer, that are outside direct project control.
- Identify possible internal and external risks and outline a mitigation activity for each risk. You must consider all possibilities.
- Plan communication and define the responsible resources from the customer side.
- Define the change management procedure to achieve changes in the most efficient way while minimizing the business effect, costs, and risks.

Development

The next step is to plan the upgrade process. This includes the following activities.

- Setting up the environment
- Merging of standard customized objects (code upgrade)
- Performing additional development, if it is required
- Testing code upgrade and fixing bugs
- Extending the upgrade toolkit that is provided by Microsoft to include procedures for upgrading data in customized tables or tables that are added to the customer's database
- Transforming objects for the RoleTailored client
- Testing of transformed objects and fixing bugs
- Upgrading trial data
- Testing data upgrade results and fixing bugs
- Developing automated test scripts
- Creating the appropriate documentation

You typically perform these tasks sequentially. However, you can change the order or run them independently if it is required. For example, you can start the trial data upgrade immediately after tables are merged.

Note: *When you plan upgrade testing, define the testing scope, all testing objectives and activities to do, and approve them with the customer or partner. Additionally, outline the test scripts, cases, and scenarios to show exactly how tests will be performed and what data will be used in scope of the upgrade project.*

Deployment

The next step is to plan the core activities that are required to implement the upgrade into the production environment. This includes the following activities:

- Installation
- Configuration
- Live data upgrade
- Adjustments to settings to obtain the final working solution for Microsoft Dynamics NAV 2013
- Acceptance testing
- User training

Customers must take part in the planning process to make sure that they have the correct resources that are available, and that deployment fits with his or her schedule.

Customers should consider if they want to keep their Dynamics NAV application running on premise, or switch to running it in a hosted environment.

This is very relevant when estimating the cost of the upgrade and cost of the deployment.

In Microsoft Dynamics NAV 2013 all server components are 64-bit only. Therefore, instead of purchasing new hardware, customers should consider a hosting scenario where they do not have to perform many maintenance tasks.

When planning an upgrade project, it is a good time to take this in consideration.

Operation

The Operation phase consists of planning project closure activities like sign off, post mortem, post-project support of the upgraded solution, and so on.

Preparation of Environments for the Upgrade

Another important aspect of the upgrade project is to make sure that the upgrade environment is established correctly to meet the requirements of the upgrade team.

Prepare Environments for the Upgrade

The upgrade environment is separated into development and testing environments.

The prerequisites for the development environment are as follows:

- Set up a dedicated workstation (or several workstations) at the customer site for use by the team.

Note: *Different Microsoft Dynamics NAV versions may be required during the upgrade. A good environment to use is a Microsoft Virtual Machine, where separate client(s) and service tiers are installed, all the required DLLs are available, and the required user permissions are simulated.*

Microsoft Dynamics NAV 2013 and Microsoft Dynamics NAV 2009 can run on the same box side by side.

- Install the following upgrade tools and make sure that they work correctly:
  - **Microsoft Dynamics NAV 2013** –For correct installation, refer to appropriate documents at the following website.

Working with Microsoft Dynamics NAV 2013 Setup

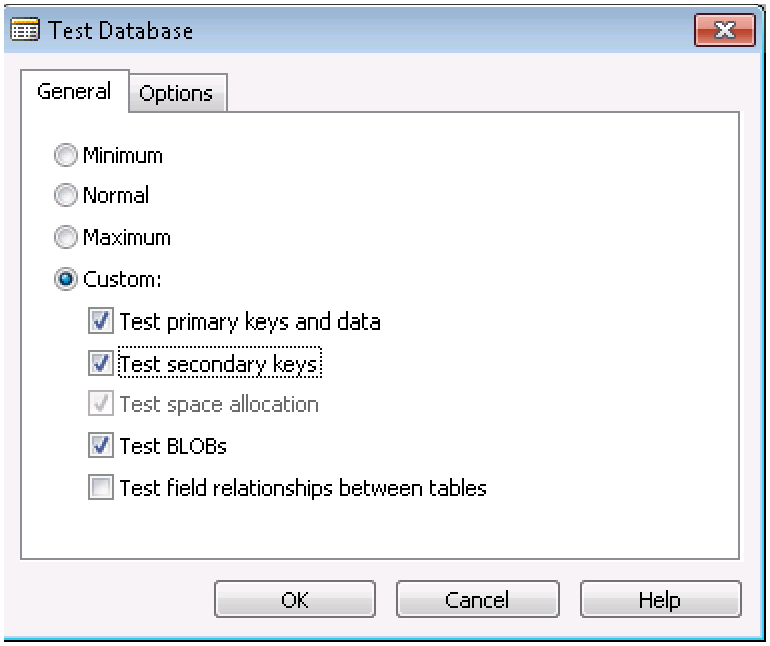
http://go.microsoft.com/fwlink/?LinkId=277022

  - **Microsoft SQL Server** – The SQL Server Express edition is installed with Microsoft Dynamics NAV. If the SQL Server Express functionality is insufficient, install another edition to meet your requirements.
  - **Upgrade Toolkit** – This toolkit is included in the documentation for Microsoft Dynamics NAV 2013, and documentation on MSDN.

- Upgrade the solution developer's license file and the customer's license file to the newest version of Microsoft Dynamics NAV.
- Restored to a new database of the same version of Microsoft Dynamics NAV, if the customer provided you with a backup file of his or her database. If the database has roles and permissions set up, you must create a super user by using a blank password if one is not provided.

Note: *When each database runs with the Microsoft Dynamics NAV 2013 client, it will be approximately double the size that it was with earlier version(s). This occurs because of Unicode compliance of the data. Make sure that you leave sufficient space on the disk for increased database size when you upgrade.*

- You first must test the database that you use during the upgrade for everything except table/field relationships. Then test for table/field relationships. You must resolve all compilation errors.



○ Minimum

○ Normal

○ Maximum

● Custom:

☑ Test primary keys and data

☑ Test secondary keys

☑ Test space allocation

☑ Test BLOBs

□ Test field relationships between tables

FIGURE 2.7:TEST DATABASE WINDOW: TESTING THE DATABASE FOR EVERYTHING EXCEPT FIELD RELATIONSHIPS BETWEEN TABLES

The testing environment must install the following instances of Microsoft Dynamics NAV and their respective databases:

- Old Base
- Current Custom
- New Base
- New Custom

Lab A: Estimate a Code Upgrade

Scenario

Victor, the solution architect who is responsible for upgrading Microsoft Dynamics NAV 2013 for Cronus International Ltd., plans the upgrade process and must estimate the code upgrade. He wants to analyze the backup of the customer database that contains both the objects and the data. To create the estimates, Victor will use WinDiff.

Victor bases his estimates on the following calculations:

- The average time that it takes an average developer to resolve a conflict that is identified by WinDiff is one time unit.
- The average time to adapt and compile a new customer object for an average developer is one time unit.
- Standard Microsoft Dynamics NAV features that are changed by the customer, and the list of test cases and scripts that are prepared by the customer to execute.
- Average time that you must have for user interface testing of each object type.

Exercise 1: Create Estimates for Code Upgrade.

Exercise Scenario

Mort, the system developer, must estimate the time that is required to upgrade the Microsoft Dynamics NAV 2009 SP1 objects to Microsoft Dynamics NAV 2013. He will focus on the objects that were customized in Microsoft Dynamics NAV 2009 SP1.

Task 1: Locate the Old Custom Object(s)

High Level Steps

1. Locate the file(s) on the AllFiles drive.

Detailed Steps

1. Locate the file(s) on the AllFiles drive.
    a. Verify that the following files are available on the HyperV in folder X in the following location: X\ Allfiles\Mod02\Labfiles\Lab A\OldCustom:
        i. table_Customer.txt
        ii. codeunit_ApplicationManagement.txt
        iii. page_CustomerCard.txt
        iv. report_CustomerList.txt

Task 2: Export the Old Base Version Objects into a Text File

High Level Steps

1. In the Object Designer, select the object to export.
2. Export the Current Custom version objects into a text file.

Detailed Steps

1. In the Object Designer, select the object to export.
   a. To start the Development Environment of Microsoft Dynamics NAV 2009 SP1 click **Start > All Programs > Microsoft Dynamics NAV 2009 Classic with Microsoft SQL Server.**
   b. To open the customer's database, in the development environment, select **File > Database > Open**. The **Open Database** window opens.
   c. In the **Server Name** field, type **"**NYC-SVR1."
   d. In the **Database Name** field, type "Demo Database NAV (6-0)".
   e. In the **Authentication** field**,** select "Windows authentication".
   f. Click **OK**.
   g. Select **Tools, Object Designer** to open Object Designer.
   h. In Object Designer, click **Table**.
   i. Select table 18, **Customer**.

2. Export the Current Custom version objects into a text file.
   a. In Object Designer, click **File**, and then **Export**. The **Export Objects** window opens.
   b. In the **File name** text box, type "table_Customer".
   c. Set **Save as type** to Text Format (*.txt).
   d. Click **Save**.
   e. Repeat steps a. through d. for the following objects:
      i. Codeunit Application Management
      ii. Page Customer Card
      iii. Report Customer - List

In case the training license does not let you export objects, you can find the objects at the location: X:\Allfiles\Mod02\Labfiles\Lab A\OldBase.

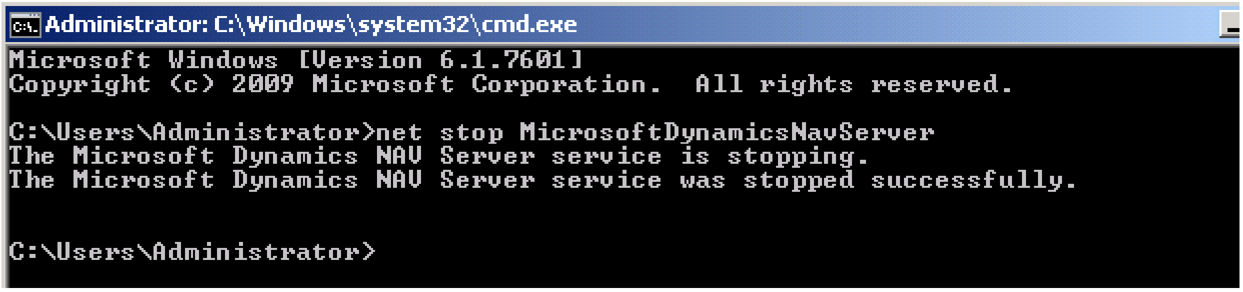
Task 3: Install Microsoft Dynamics NAV 2013

High Level Steps

1. Stop the Dynamics NAV Server.
2. Install Microsoft Dynamics NAV 2013.
3. Import the License for Microsoft Dynamics NAV 2013.

Detailed Steps

1. Stop the Dynamics NAV Server.
    a. Click **Start**, and then **Run** to open the **Run** window.
    b. In the **Run** window, in **Open,** type "cmd", and then click **OK**.
    c. In the window that opens, type "net stop MicrosoftDynamicsNAVServer", and then select **Enter** as shown in the figure "Stop Dynamics NAV Server."



```
Administrator: C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\Administrator>net stop MicrosoftDynamicsNavServer
The Microsoft Dynamics NAV Server service is stopping.
The Microsoft Dynamics NAV Server service was stopped successfully.


C:\Users\Administrator>
```

FIGURE 2.8:STOP DYNAMICS NAV SERVER

2. Install Microsoft Dynamics NAV 2013.
    a. On the X-drive of the HyperV, locate the installation media for Dynamics NAV 2013. It should be available at the location: X:\NAV 2013 Setup files\DVD_BUILD33781\**.**
    b. Run the Setup.exe file.
    c. Click **Next > I Accept > Install Demo** to start the installation.
        i. In case there is an error:
            - Restart the HyperV.
            - Stop the Dynamics NAV Server.
            - Run the setup again.

3. Import the License for Microsoft Dynamics NAV 2013.
    a. Select **Start > All Programs > Microsoft Dynamics NAV 2013 Development Environment**.
    b. Click **OK**.
    c. In the **Open Database** window, in **Server Name,** type "NYC-SRV1\NAVDEMO**".**
    d. In **Database Name,** type "Demo Database NAV (7-0)".
    e. In **Authentication,** select **Windows Authentication**.
    f. Click **OK** to open **Object Designer**.
    g. Select **Tools**, and then **License Information** to open the **License Information** window.
    h. Click **Upload** to open the **Upload License File** window.

i. In the **File name** text box, type "X:\NAV 2013 Training License\NAV 2013 training demo_dev license.flf", and then click **Open.**

j. Click **OK** to confirm.

k. Close the Dynamics NAV 2013 Development.

Task 4: Export the Corresponding Microsoft Dynamics NAV 2013 Version Objects into a Text File

High Level Steps

1. Open the Object Designer Filter to view the objects to export.
2. Export the corresponding Dynamics NAV 2013 version objects into a text file.

Detailed Steps

1. Open the Object Designer Filter to view the objects to export.
   a. To start the Development Environment of Microsoft Dynamics NAV 20013 click **Start > All Programs > Microsoft Dynamics 2013 Development Environment**.
   b. In Object Designer, click **Table.**
   c. Select table 18, **Customer**.

2. Export the corresponding Dynamics NAV 2013 version objects into a text file.
   a. In **Object Designer,** click **File**, and then click **Export.** The **Export Objects windo**w opens.
   b. In the **File name** text box**,** type "table_Customer".
   c. Set **Save as type** to **Text Format** (*.txt).
   d. Click **Save**.
   e. Repeat steps a. through d. for the following objects:
      i. Codeunit Application Management
      ii. Page Customer Card
      iii. Report Customer - List

In case the training license does not let you export objects, you can find the objects in the location: X:\ Allfiles\Mod02\Labfiles\Lab A\NewBase.

Task 5: Use the TextFormatUpgrade2013 tool to validate the Old Custom and Old Base Objects

High Level Steps

1. Run the text upgrade tool for the exported file(s).

Detailed Steps

1. Run the text upgrade tool for the exported file(s).
   a. On the X:\ drive, locate the Object Change Tools file in the following location: **X:\NAV 2013 Setup files\DVD_Build33781\UpgradeToolkit\Object Change Tools**.
   b. Copy the two files (TextFormatUpgrade2013.exe and TextFormatUpgrade2013.exe.config) in this folder to the Old Custom folder in the following location: **X:\ Allfiles\Mod02\Labfiles\Lab A\OldCustom**.
   c. On the HyperV, click Start, and then click Run. The **Run** window opens.
   d. In the **Open** field, type "cmd" and select Enter to open a Command Prompt.
   e. At the command prompt, type "cd X:\Allfiles\Mod02\Labfiles\Lab A\OldCustom", and then click **Enter**.
      i. As an alternative select **Start**, **Computer**. The Computer window opens.
      ii. Browse to the folder: "X:\Allfiles\Mod02\Labfiles\Lab A\OldCustom".
      iii. Select this folder then select click the **SHIFT** button and right-click to open a context window.
      iv. In the context window select "Open command window here" as shown in figure "Open command window here".

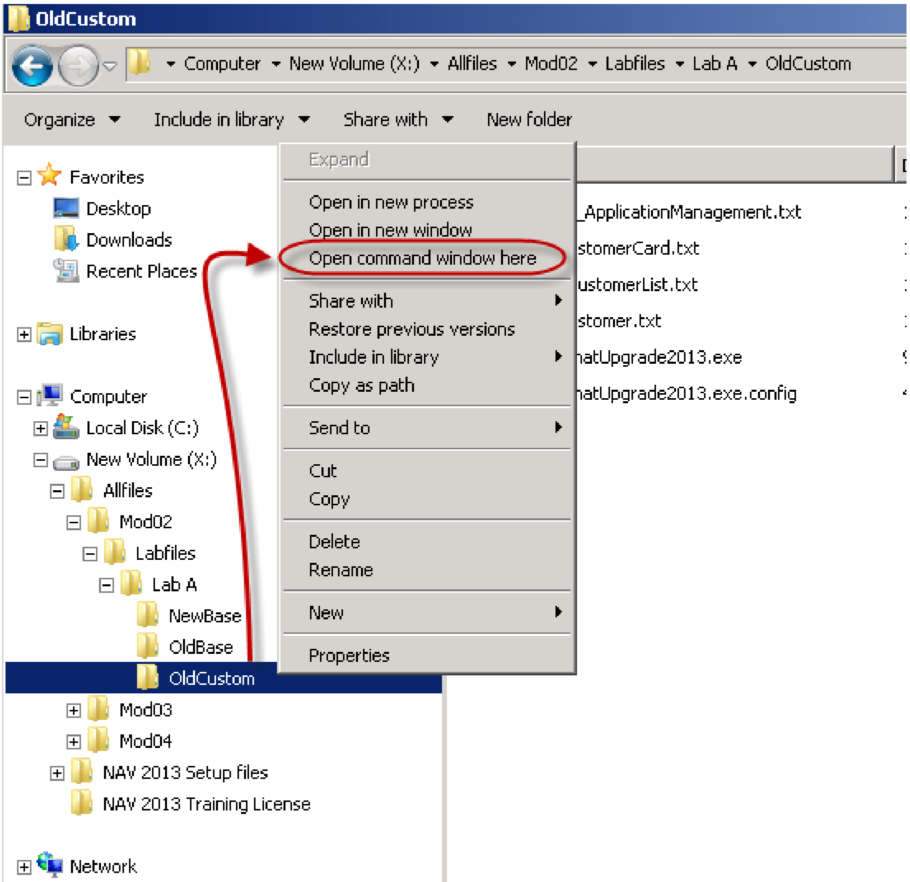


FIGURE 2.9:OPEN COMMAND WINDOW HERE

f. In the command window, type the command:

```
“TextFormatUpgrade2013 "X:\Allfiles\Mod02\Labfiles\Lab
A\OldCustom\table_Customer.txt"”
```

g. Click **Enter.** The Text Upgrade Toolkit executes and generates an output file in the folder named Converted.

h. Repeat steps f. through g. for the following objects that were exported in the task “Export the Current Base Version Objects into a Text File”:

  i. codeunit_ApplicationManagement.txt

  ii. page_CustomerCard.txt

  iii. report_CustomerList.txt

i. Repeat steps a. through g. for the objects in the folder: X:\ Allfiles\Mod02\Labfiles\Lab A\OldBase:
   i. table_Customer.txt
   ii. codeunit_ApplicationManagement.txt
   iii. page_CustomerCard.txt
   iv. report_CustomerList.txt

Task 6: Import and Compare the Current Base and Current Custom and New Base Objects from the Text Files into WinDiff.

High Level Steps

1. Start WinDiff.
2. Import the Objects to compare.

Detailed Steps

1. Start WinDiff.
   a. Click **Start.** Type "WinDiff" in the **Search programs and files** text box, and then click **Enter**. The **WinDiff** window opens.

2. Import the Objects to compare.
   a. In WinDiff, click **File**, and then click **Compare Directories**. The **Select Directories** window opens.
   b. In Dir 1, type: "X:\ Allfiles\Mod02\Labfiles\Lab A\OldCustom\Converted".
   c. In Dir 2, type: "X:\ Allfiles\Mod02\Labfiles\Lab A\OldBase\Converted".
   d. Click **OK**. WinDiff now shows whether there are differences between files in the two folders. Note the files that are different (indicated in red).
   e. Repeat steps a. through d. for the following directories:
      i. X:\ Allfiles\Mod02\Labfiles\Lab A\OldBase\Converted
      ii. X:\ Allfiles\Mod02\Labfiles\Lab A\NewBase

Lab B: Estimate Report Transformation

Scenario

Because of the Microsoft Dynamics NAV 2009 SP1 upgrade to Microsoft Dynamics NAV 2013, Cronus International Ltd. wants to transform all the reports. Victor, the Solution Architect, must analyze the customer database backup and create estimates for object transformation for new reports.

Victor knows that the time that an average developer must transform and test a report depends on the report type. The time that you must have for report transformation and then later testing of the transformed reports for an average developer is as follows, based on the report type.

Report Type	Required Dev Time Need (units)	Required Test Time (units)
Document reports	6	2
List-type reports that have complex date items structure, grouping, and totals calculation	4	1.5
List-type reports that have simple grouping	2	0.5

A *unit* is a measurement that you use to estimate time. For example, it can be one hour or two hours, depending on the experience of the resource that performs the task.

Victor is aware of the standard Microsoft Dynamics NAV features list that is changed by the customer, and the test cases and scripts that the customer prepared to execute

Exercise 1: Create Estimates for Report Transformation.

Task 1: Define the Type of Each Report to Be Transformed

High Level Steps

1. Count the number of reports of different types.

Detailed Steps

1. Count the number of reports of different types.
    a. To start the Development Environment of Microsoft Dynamics NAV 2009 SP1 click **Start > All Programs > Microsoft Dynamics NAV 2009 Classic with Microsoft SQL Server**.

    i. If there is an error message "The Demo Database NAV (7-0) database on the NYC-SVR1\NAVDEMO server cannot be opened by this version of Microsoft Dynamics NAV Classic. The database has already been converted by a newer program version. You must upgrade Microsoft Dynamics NAV Classis to the latest version to open the database", click **OK** and continue.

b. To open the customer's database in the development environment, select **File > Database > Open**. The **Open Database** window opens.

c. In the **Server Name** text box, type "NYC-SVR1".

d. In the **Database Name** text box, type "Demo Database NAV (6-0)".

e. In **Authentication,** select **Windows Authentication**.

f. Click **OK**.

g. Select **Tools**, and then select **Object Designer** to open **Object Designer**.

h. Apply a filter to display only the modified reports in the customer's database.

    i. Select **View**, **Table Filter** to open the Object – Table Filter window

    ii. In the Object – Table Filter window in the first column replace **Name** with **Modified**.

    iii. In the second column type **Yes** and then click **OK**.

i. Note the result in this table.

Report Type	Quantity
List-type reports that have complex data items structure, grouping, and totals calculation.	
List-type reports that have simple grouping.	
Batch jobs with code on the GroupFooter/Header sections.	

j. Provide estimates for report transformation that are based on the average transformation time that each report type requires.

k. Calculate the time that you must have to test the transformed reports.

l. Define the approximate list of test cases to execute during functional testing of standard and customized functionality. Calculate the total hours for functional testing.

Summary

The Analysis and Planning phase of the upgrade project is one of the most important phases in the upgrade process, as correct planning, identification of business processes, and scoping improve the chances of success for the project. The following activities are included in this phase:

- Defining requirements
- Estimating the upgrade
- Defining the testing strategy
- Planning upgrade workflow
- Preparing upgrade environments

Module Review

Test Your Knowledge

Test your knowledge with the following questions.

1. Conducting Fit-Gap Analysis is necessary only if the customer wants to create additional functionality during the upgrade.

   ( ) True

   ( ) False

2. Functional Testing is intended to verify that all code is transferred correctly, and that no functionality is lost during the code upgrade.

   ( ) True

   ( ) False

3. Name the documents that you must develop when planning the workflow of an upgrade project.

____

____

____

____

4. Which of the following actions do you perform when you are preparing environments for the upgrade?

   ( ) Make a clean installation of Microsoft Dynamics NAV 2013

   ( ) Clone the source solution to a dedicated workstation

   ( ) Use the customer's license file to the target version

   ( ) Create a backup of the source database

5. According to the Microsoft Dynamics Sure Step methodology, an upgrade project starts with the diagnosis and analysis of the system that will be upgraded, and planning the activities that the project team will perform. What are the main project analysis and planning activities?

____

____

____

____

Test Your Knowledge Solutions

Module Review and Takeaways

1. Conducting Fit-Gap Analysis is necessary only if the customer wants to create additional functionality during the upgrade.

   (√) True

   ( ) False

2. Functional Testing is intended to verify that all code is transferred correctly, and that no functionality is lost during the code upgrade.

   ( ) True

   (√) False

3. Name the documents that you must develop when planning the workflow of an upgrade project.

   MODEL ANSWER:

   The documents that describe the workflow of the upgrade project are Project Charter and Project Plan.

4. Which of the following actions do you perform when you are preparing environments for the upgrade?

   (√) Make a clean installation of Microsoft Dynamics NAV 2013

   ( ) Clone the source solution to a dedicated workstation

   ( ) Use the customer's license file to the target version

   ( ) Create a backup of the source database

5. According to the Microsoft Dynamics Sure Step methodology, an upgrade project starts with the diagnosis and analysis of the system that will be upgraded, and planning the activities that the project team will perform. What are the main project analysis and planning activities?

   MODEL ANSWER:

   The project analysis and planning activities are as follows:

   - Defining requirements for the target solution
   - Estimating all upgrade actions
   - Defining the upgrade testing strategy
   - Elaborating a comprehensive upgrade workflow plan
   - Establishing the environments for supporting the upgrade process