

MODULE 1: MICROSOFT DYNAMICS NAV DEVELOPMENT ENVIRONMENT

Module Overview

The key to developing solutions in Microsoft Dynamics NAV 2013 is understanding the basic objects that are available in Microsoft Dynamics NAV 2013, learning about the Microsoft Dynamics NAV Development Environment fundamentals, and understanding database structure concepts.

Objectives

- Present the basic object types in Microsoft Dynamics NAV 2013.
- Describe fundamental aspects of Microsoft Dynamics NAV Development Environment. This includes the UI, application objects, and basic metadata concepts.
- Explain the physical and logical database structure.
- Explain the features for multi-developer environments.

Basic Objects in Microsoft Dynamics NAV

There are seven basic object types available in Microsoft Dynamics NAV 2013. As a developer, you can edit existing objects or create new objects of the specified types.

The seven types are as follows:

Object	Remarks
Table	Describes how data is stored in the database and how it is retrieved. Table is the most important object type, because most object types depend on them.
Page	Enables users to view, add, change, or delete records in a table.
Report	Prints or previews the data from a Microsoft Dynamics NAV 2013 database.
Codeunit	An organized unit of programming code, typically for managing a single process. Codeunits are called from other objects to complete a specific task.
Query	Defines a relational data model for direct and efficient querying of the underlying Microsoft SQL Server database.
XMLport	Imports or exports data to or from a Microsoft Dynamics NAV 2013 database, in XML or text format.
MenuSuite	Describes menus that are displayed in the Departments in the Microsoft Dynamics NAV 2013 client for Windows.

Is Microsoft Dynamics NAV 2013 Object Oriented?

Microsoft Dynamics NAV 2013 is not object-oriented but object-based. This is an important difference. In an object-oriented language or environment, developers can create new types of objects based on the objects that are already in the system. In Microsoft Dynamics NAV 2013, there are seven types of application objects. You can create new objects of any of these types. However, you cannot create new object types, or inherit any functionality between objects of the same type.

Limiting developers to these seven object types makes development faster and more efficient. The biggest benefit from this limitation is stability. It is fairly difficult to create a severe bug in Microsoft Dynamics NAV 2013.

Object Designer Fundamentals

Development in Microsoft Dynamics NAV 2013 is performed using the Microsoft Dynamics NAV Development Environment. This environment includes features for application development, database maintenance, and license administration. The central development feature is the Object Designer. Object Designer consists of the seven object designers for designing different application object types in Microsoft Dynamics NAV 2013. For example, tables are created by using the Table Designer; pages are created by using the Page Designer, and so on.

Object Designer provides access to all application objects in a Microsoft Dynamics NAV 2013 application.

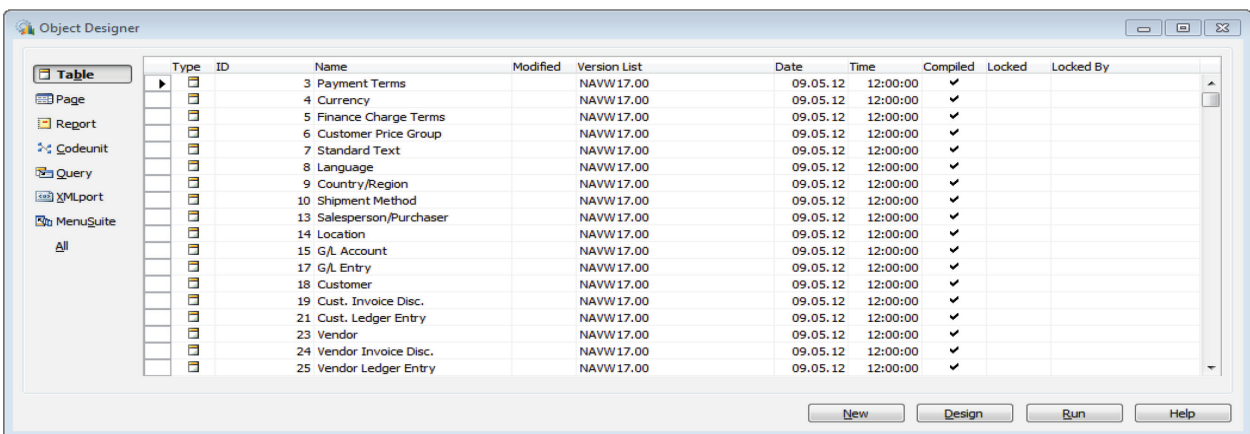


FIGURE 1.1: OBJECT DESIGNER

Application objects are created by using several general concepts. A fundamental knowledge of these concepts speeds up the C/SIDE application development process.

Opening a Database

Before you start development, you must first open a database. To open the database, click **File > Database > Open**. The **Open Database** window opens.

The **Open Database** window enables you to connect to a Microsoft SQL Server database that stores the Microsoft Dynamics NAV 2013 application objects and data.

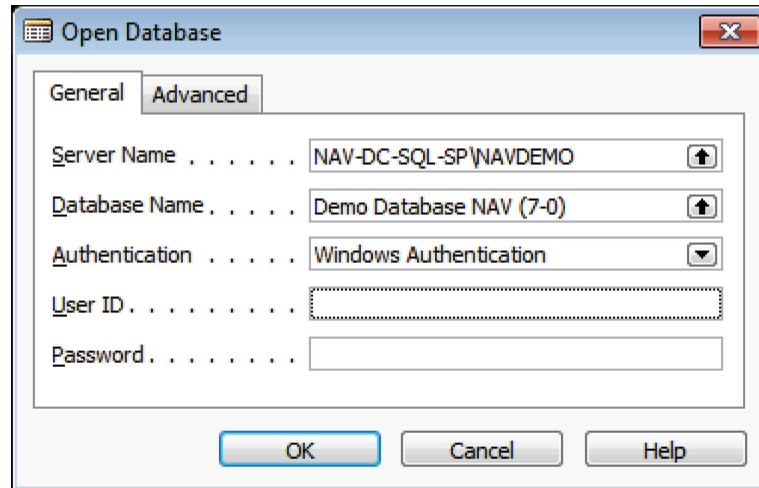


FIGURE 1.2: THE OPEN DATABASE WINDOW

You first must specify the Microsoft SQL Server that hosts the Microsoft Dynamics NAV 2013 database. You can either enter the server name in the **Server Name** field, or you can select a server in the **Available Microsoft SQL Servers** window.

The Available Microsoft SQL Servers window displays the list of Microsoft SQL Server instances that are detected to be running in the network.

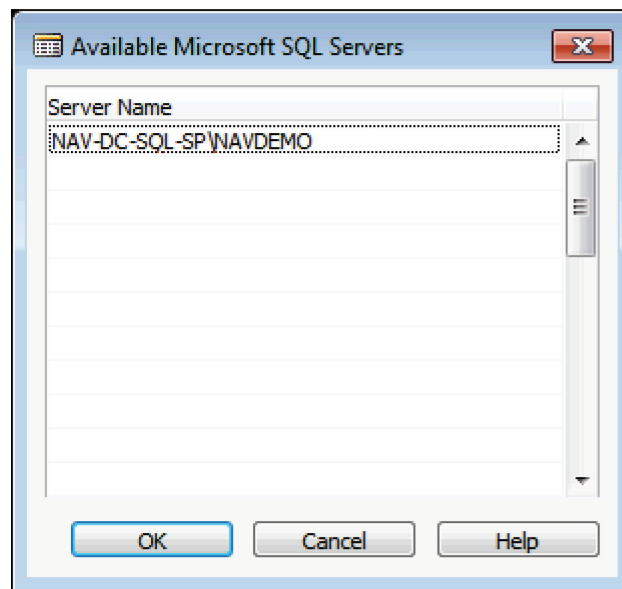


FIGURE 1.3: THE AVAILABLE MICROSOFT SQL SERVERS WINDOW

You must select the authentication mode in the **Authentication** field. You can select from **Windows Authentication** and **Database Authentication** modes.

Finally, you must select the Microsoft SQL Server database. You can enter the name in the **Database Name** field, or you can select the **Database Name** field to access the **Available Databases** window.

The **Available Databases** window displays all Microsoft Dynamics NAV databases.

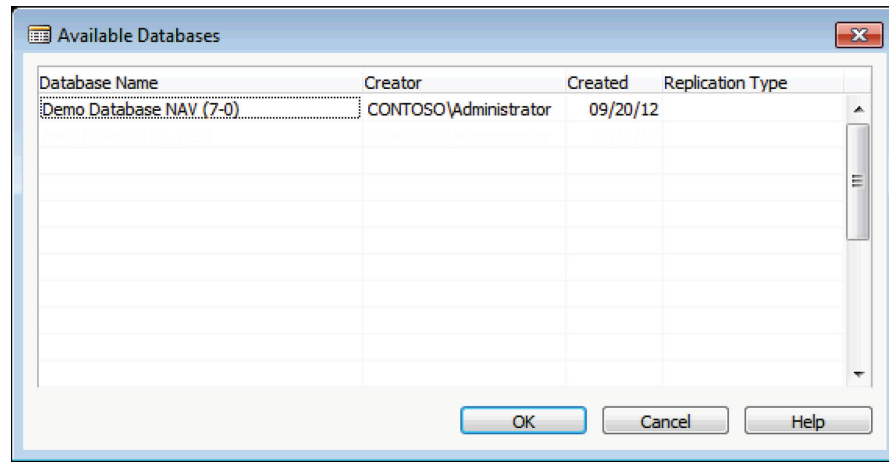


FIGURE 1.4: THE AVAILABLE DATABASES WINDOW

Note: The **Available Databases** window displays only the Microsoft Dynamics NAV databases that are stored on the selected Microsoft SQL Server, and only those databases to which you have access. Any other databases stored on the server are not shown. However, the list does not differentiate between Microsoft Dynamics NAV 2013 and earlier versions. Databases that belong to any version of Microsoft Dynamics NAV are displayed in this window.



To complete opening the database, in the **Open Database** window, click **OK**.

Note: Microsoft Dynamics NAV Development Environment remembers which database was opened last, and opens the same database automatically when you start the Microsoft Dynamics NAV Development Environment.



Managing Objects

You manage Microsoft Dynamics NAV 2013 application objects from **Object Designer**. To access **Object Designer**, click **Tools > Object Designer**, or press SHIFT+F12. **Object Designer** has three areas: object type buttons, object list and object design buttons.

The object list is the central part of the **Object Designer**. It displays the application objects together with their most important properties. You can set the following properties:

Property	Remarks
Type	Object type of the object. It is represented by an icon. You cannot change the type of an existing object.
ID	<p>Unique identifier of the object within the object type. You assign the ID during object creation.</p> <hr/> <p> Best Practice: Do not change the ID of an existing object. Variables and some object properties refer to other objects by using their IDs. If you change an ID, existing code can fail or more serious bugs can occur.</p> <hr/>
Name	<p>Unique name of the object within the object type. You assign the name typically when you create the object. However, you can change it later. A name can contain up to 30 characters, and may contain any character that is permitted by the code page that is specified in the collation option of the database.</p> <hr/> <p> Best Practice: Changing the object name is possible, but not recommended. If you change an object name, you do not affect any of the compiled objects. However, you may prevent uncompiled objects from compiling. You should only consider changing object names for the objects that you created, and before you refer to them from other objects.</p> <hr/>

Module 1: Microsoft Dynamics NAV Development Environment

Property	Remarks
Modified	<p>Indication that an object was changed by a developer. The Modified flag is switched on automatically by Microsoft Dynamics NAV Development Environment when you save an object. You can switch the flag on and off as needed.</p> <hr/> <p> Best Practice: <i>When working on an object, it is best to ignore the state of the Modified flag. As soon as you are ready to deploy the object to customer's live environment, you should clear this flag.</i></p>
Version List	<p>List that contains the versions that have changed an object. Each version is represented by a tag, and version tags are separated by commas. A version list can contain up to 80 characters.</p> <hr/> <p> Best Practice: <i>Always tag the objects that you change by appending another version tag to the version list. Include both the alphanumeric and numeric part of the version list.</i></p>
Date	<p>Field that displays the date when an object was last saved. It is set by Microsoft Dynamics NAV Development Environment. However, you can change it manually.</p>
Time	<p>Field that displays the time when an object was last saved. It is set by Microsoft Dynamics NAV Development Environment. However, you can change this field manually.</p>

Property	Remarks
Compiled	Indication that an object was saved in a compiled form. You cannot change this property manually. However, you can decide whether to compile an object any time that you save it, or you can compile it by clicking Tools > Compile . You cannot run or otherwise use an uncompiled object in a Microsoft Dynamics NAV 2013 application.
Locked	Indication that an object is locked by a developer. A locked object can be saved only by the developer who locked it.
Locked By	Indication that someone has locked an object. Only the developer whose user ID is shown in this field can save the object.

Object type buttons are located to the left of the object list, and they filter the object list by a specific type. A button represents each object type in Microsoft Dynamics NAV 2013, and there is an additional **All** button. If you click a button that represents an object type, the object list filters to display only the objects of the selected type. If you click **All**, the object list displays all objects regardless of type.



Note: You can apply additional filters to the list, by clicking one of the filter buttons in the toolbar, or by selecting either **Field Filter** or **Table Filter** on the **View** menu.

Object design buttons are located at the bottom of the object list. The **New** button creates a new object of the currently selected object type. The **Design** button opens the selected object in the appropriate designer. The **Run** button runs the object in the Microsoft Dynamics NAV 2013 client for Windows.



Note: For the **Run** button to actually run the object, you must first open a company. To open a company, click **File > Company > Open**.

Creating an Object

The following steps show how to create an object.

1. Select the object type by clicking the appropriate object type button to the left of the object list
2. Click **New**.

For example, to create a new table, click **Table**, and then click **New**.

Each object type has a different designer in Microsoft Dynamics NAV Development environment. Depending on the selected object type, a different designer is displayed when you are creating a new or editing an existing object.

 **Note:** Different object type designers are explained in detail in specific chapters in this course.

Editing an Object

To edit an object, select that object in **Object Designer**, and then click **Design**. This opens the selected object in the appropriate designer for the object type.

For example, to edit table 18, Customer, click **Table**, and then click **Design**. This opens the **Table Designer** window.

Saving an Object

When you are designing an object, either by creating a new one or editing an existing one, you can save the object to the database.

To save an object, click **File > Save**, or press CTRL+S on your keyboard.

The **Save As** window will show if you are saving an object for the first time.

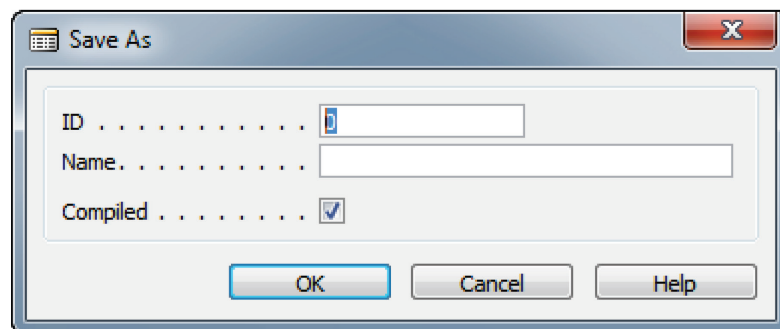


FIGURE 1.5: THE SAVE AS WINDOW

The **Save As** window requires you to specify the **ID** and **Name** properties for the object. Both the **ID** and **Name** must be unique within the object type. If you accidentally specify the ID of an existing object, Microsoft Dynamics NAV Development Environment asks you if you want to replace the existing object. If you specify the name of an existing object, then the save operation fails with an error.



Note: The IDs that you can specify when you save objects is regulated by the object range assignments and permissions in the Microsoft Dynamics NAV 2013 license.

In the **Save As** window, you can decide whether to compile the object before you save it, or to save it in the uncompiled state.

If you are saving the object which was saved earlier, after you click **File > Save**, instead of the **Save As** window, the **Save** window is displayed.

The **Save** window is displayed when you are saving an object which has already been saved in the database.

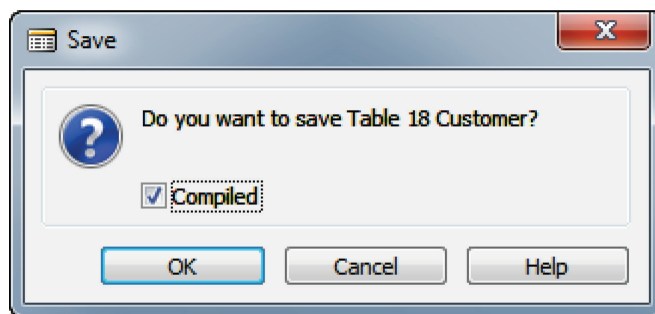


FIGURE 1.6: THE SAVE WINDOW

Similar to the **Save As** window, the **Save** window also enables you to decide whether to also compile the object before you save it, or to leave it in the uncompiled state.



Note: If you want to create a copy of an existing object, you can save the object under a different ID and Name. To do this, on the **File** menu, click **Save As**. This opens the **Save As** window, where you must change both the ID and the Name properties, and then click **OK**.

Compiling an Object

An object can be saved in its compiled, or uncompiled state. Only compiled objects can be run from the **Object Designer**, or otherwise accessed by a user or application code.

You can compile an object both from the **Object Designer** window, or when you edit the object in its specific designer. To compile an object, click **Tools > Compile**.

Module 1: Microsoft Dynamics NAV Development Environment

The **Compile** function is checking the definition of the object and the C/AL code in the object, and makes sure that the object can be run by Microsoft Dynamics NAV 2013. If a problem prevents the object from being run, the compile function displays the error which explains the problem. If **Compile** succeeds, no message is shown. If you are accessing the **Compile** function from **Object Designer**, and the function succeeds, the **Compiled** flag in the object list changes the state to selected.

You can compile multiple objects at the same time, by selecting the objects in the **Object Designer**, and then accessing the **Compile** function. When you compile multiple objects, the compiler does not stop when it encounters an error, but continues compiling until all objects are compiled. If there were any errors during compiling, the **Error List** window is shown, that specifies which objects contain errors.

The **Error List** window shows all compile-time errors after compiling multiple objects at the same time.

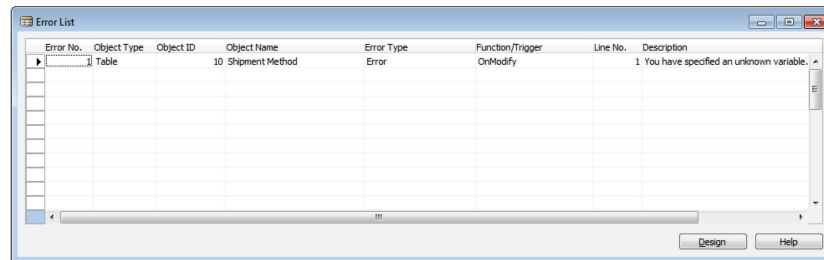



FIGURE 1.7: THE ERROR LIST WINDOW

 **Note:** If you click **Design** in the **Error List** window, if it is a code-level error, the object opens in the appropriate designer, with the **C/AL Editor** positioning the code at the line where the compile-time error occurs.

When you compile multiple objects, all objects that contain a compile-time error are automatically marked by the compiler. You can view only the marked objects, by clicking **View > Marked Only**. To view all objects again, on the **View** menu, click either **Show All**, or **Marked Only** again.

Marked objects are represented by a small dot in the row selection box, to the left of the **Type** column.

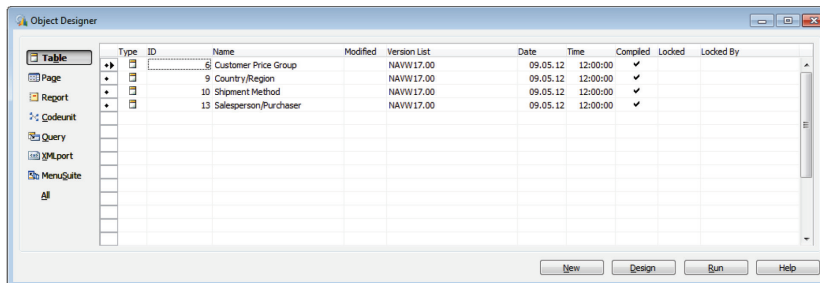


FIGURE 1.8: THE MARKED ONLY VIEW OF THE OBJECT DESIGNER



Note: The mark on an object is automatically cleared when you compile multiple objects again, and the object compiles successfully. To toggle the mark on an object manually, selecting the object in the **Object Designer**, and then on the **Edit** menu, click **Toggle Mark**, or press **CTRL+F1** on your keyboard.

When you are saving an object, the Microsoft Dynamics NAV Development Environment you can decide to also compile the object as a part of the saving process. To compile objects during saving, you can select the **Compiled** check box. This is available in both the **Save**, and the **Save As** windows. If you select the **Compiled** check box, Microsoft Dynamics NAV Development Environment will first try to compile the object, and only if there were no errors, it will save the object to the database.

Properties

Properties define the characteristics of an object or an element of an object, such as behavior or appearance. For example, the **InsertAllowed** property defines if a user can create new records in a table, and the **Editable** property defines if a user can edit the contents of a control. Depending on the object type, there are different properties that you can set on an object, and all objects, except for MenuSuites, enable you to change the behavior or appearance of the object by specifying properties.

You can only access the properties of an object while editing the object in the designer. To access the properties of an object, select the first empty row in the designer, and then click **Edit > Properties** or press **SHIFT+F4** on your keyboard. Or, you can click the **Properties** button in the toolbar. This opens the **Properties** window.



Note: For codeunits, you do not have to select any row specifically, and you can always access the codeunit properties by clicking **Edit > Properties**.

Module 1: Microsoft Dynamics NAV Development Environment

The empty row is indicated by an asterisk to the left of the row selection indicator. In the following figure, the empty row is selected in the **Table Designer** window.

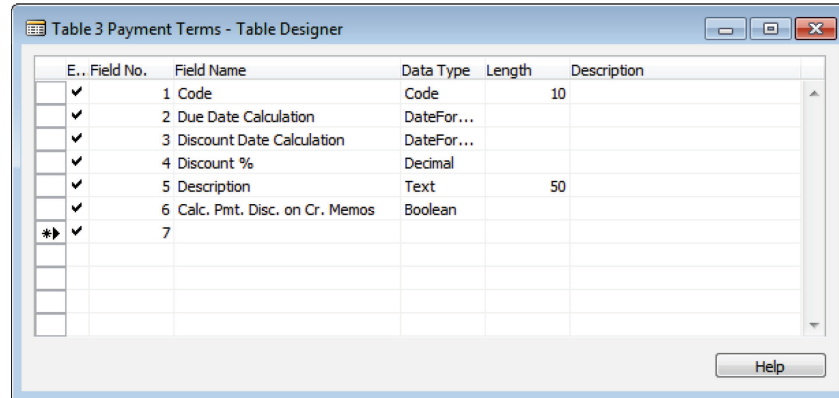


FIGURE 1.9: EMPTY ROW IN THE TABLE DESIGNER



Note: If you access **Properties**, and a non-empty row is selected, then you are accessing the properties of the object element currently selected in the designer, instead of the properties of the object itself.

Regardless of if you are accessing the properties of an object, or an object element, such as a control or an action, the **Properties** window has the same structure, and is presented as a table with two columns. The **Property** column contains the name of the property. This is noneditable, and the **Value** column contains the value of the property, which you can set. Some properties, such as **ID**, **Name**, **CaptionML** or **Description**, are common for all object types, whereas most other properties are specific to the object type of the object that you are editing.

The Properties window shows the properties of an object of an object element.

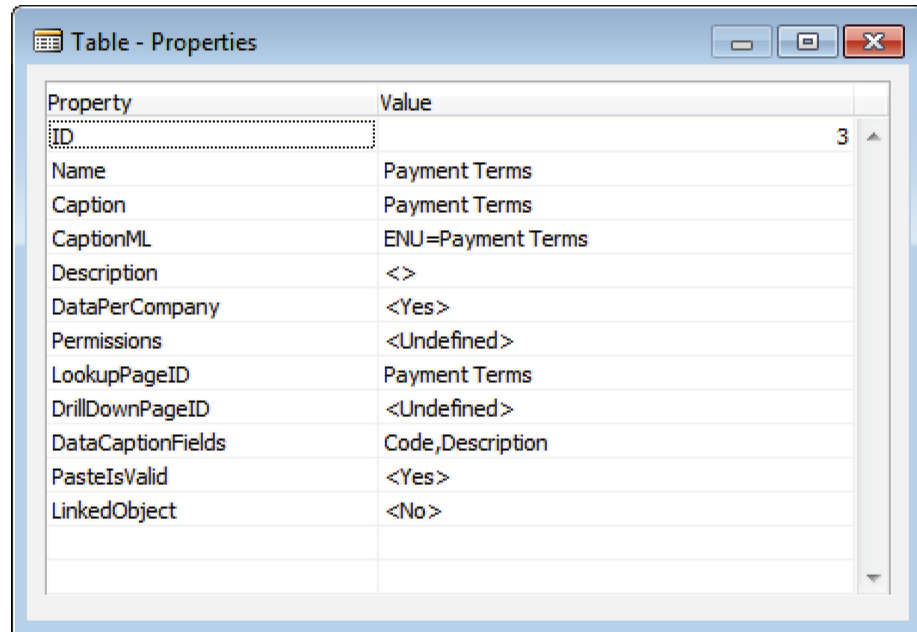



FIGURE 1.10: THE PROPERTIES WINDOW

Each property has a default value. This is indicated by < and > around the property value. For example, the **DataPerCompany** property for a table has the default value of **Yes**, which is displayed as <Yes>. You can always revert to the default property value, by deleting the property value in the **Value** column.

To change a value of a property, you can do the following:

- Enter the value in the property manually.
- Select the value from the drop-down list. For example, the **DataPerCompany** table property enables you to select values of **Yes** or **No**.
- Select the value from the look-up list. For example, the **LookupPageID** table property enables you to select from the look-up list.
- Access the specific property editor, by clicking the **AssistEdit** button on the right side of the property value column. For example, the **Permissions** property on any object lets you access the **Permissions** window where you can define the property value more intuitively, than by entering the value manually.

 **Note:** You can access drop-down, look-up or a specific property editor by pressing the F6 key or the ALT+DOWN combination on your keyboard.

Triggers

Triggers are predefined functions that execute when certain actions occur in Microsoft Dynamics NAV 2013. For example, triggers occur when users insert rows into a table, click an action, or run a page.

All objects, except for MenuSuites, contain triggers. There are *object triggers* that occur when the event applies to the object itself (such as **OnInsert** trigger in a table), and object element triggers that occur when the event applies to an object element, such as fields, controls or actions. When triggers relate to an object element, they are referred to with the specific element name, for example *field triggers*, *control triggers* or *action triggers*.

Each object and object element defines different triggers. However, you can always access the triggers in a uniform way, regardless of which object type that you are designing. To access the triggers of an object, select the first empty row, and click **View > C/AL Code**, or press **F9** on your keyboard. Or, you can click the **C/AL Code** button in the toolbar.

Other Object Definition Elements

Other than properties and triggers, application object definitions contain several other types of elements. Some of those elements belong to all object types, but most elements have similar purpose and follow the same basic design and behavior principles. For example, most object types enable you to define the C/AL globals, such as variables, text constants and functions. These can be accessed from the C/AL code within the object. Also, some elements, such as fields or field groups, belong only to tables, and cannot be defined in any other object type.

Those specific elements are explained in detail with explanations of specific object types later in this course.

Team Development Features

Microsoft Dynamics NAV Development Environment supports environments where multiple developers are working together on a single Microsoft Dynamics NAV 2013 application. The highest risk of having multiple developers working on the same application is a possibility that one developer could overwrite the changes that were made by another developer. This results in lost work and possible bugs.

To make the development in team environments easier and less risky, Microsoft Dynamics NAV Development Environment lets developers lock and unlock the objects. This prevents accidental overwrites of the same objects by other developers.

Locking

If two developers are editing the same object at the same time, Microsoft Dynamics NAV 2013 will only keep the version of the object which was saved the last. To prevent other developers from accidentally changing an object that you are editing, you can lock the object.

To lock an object, select the object in the Object Designer, then click **File > Lock**, or press CTRL+ALT+L on your keyboard. After you lock an object, **Locked** field for the object is automatically selected, and the **Locked By** property shows the name of the user who locked the object.

When an object is locked, only the developer who has locked it can save the object. Other users can still do the following:

- Open the object in the read-only mode.
- Compile the object.
- Run the object.

When an object is lock, no other user can lock it until the user who has originally locked it unlocks it.

Microsoft Dynamics NAV Developer Environment asks for confirmation if a developer accesses an object locked by another user.

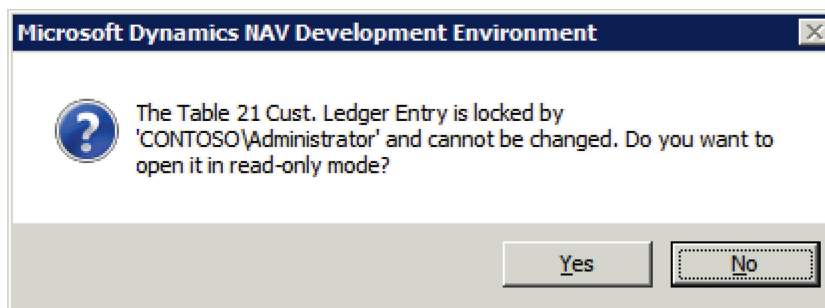



FIGURE 1.11: MICROSOFT DYNAMICS NAV DEVELOPMENT ENVIRONMENT WINDOW WITH YES/NO BUTTONS.

When an object is opened in the read-only mode, the **Save** function on the **File** menu is disabled, and the caption bar of the object designer states that the object is [Read-Only].

 **Note:** If you are editing an object in the read-only mode, you can still save a copy of the object, by selecting **Save As** on the **File** menu.

Unlocking

When you complete the development work on a locked object, you can unlock it so that other developers can change it. To unlock the object, on the **File** menu, click **Unlock**, or press CTRL+ALT+U on your keyboard.

Unlocking the object will reset the **Locked** flag, and remove the value from the **Locked By** property. After unlocking, other users can open the object or lock the object for themselves.



Note: You can simultaneously lock or unlock more than one object, by selecting the objects that you want locked or unlocked, and then clicking the **Lock** or **Unlock** on the **File** menu respectively.

Force Unlocking

Sometimes a user who has locked an object is unavailable, and the object needs to be changed by another user. In these situations, you can force-unlock the object. Only the users who have SUPER permission set can force-unlock objects locked by other users.

To force-unlock an object, or a range of objects, select the objects in the Object Designer, then click **File > Force Unlock**.



Note: Because there is no automatic version control in Microsoft Dynamics NAV Development Environment, the Force Unlock will not return the object to the state it had before the original user has locked it, but to the last saved state before you force-unlocked it.

Auto-Locking

Users may frequently forget to lock the objects before opening them in the designer. You can configure Microsoft Dynamics NAV Development Environment to automatically lock the objects when a user designs them.

To switch auto-locking on, click **Tools > Options**, to open the **Options** window. In the **Options** window, set the **Auto-Lock on Design** property to **Yes**, and then click **OK**.



Note: This change affects only the user who has made it. Every user must set this option himself or herself.

When auto-locking is switched on, when you click **Design** on an object, Microsoft Dynamics NAV Development Environment immediately locks the object for you. If you are creating a new object, the object is locked after you save it for the first time.



Note: *There is no automatic unlocking in Microsoft Dynamics NAV 2013. As soon as you auto-lock an object, the object remains locked until you unlock it.*

Physical and the Logical Databases

Typical database users are not concerned with the size and the location of data on disk. Developers want to be sure that when they refer to a name, the correct value is returned. Microsoft Dynamics NAV 2013 provides a conceptual representation of data that does not include too many details about how the data is stored. An abstract data model uses logical concepts (such as objects, their properties, and their relations) that are easier to understand.

Users should only concern themselves with the structure of the data and the relationships between different pieces of information and not with how these structures and relations are implemented. The physical database deals with how the structures in the logical database and the search paths between them are implemented in Microsoft SQL Server.

When the term *database* is used, it generally refers to the logical database unless otherwise noted.

What is visible to the user is a coherent set of information in the C/SIDE database system that is stored in several physical disk files.

Access to the data is made possible by a well-defined logical organization consisting of:

- Fields
- Records
- Tables
- Companies

Lesson Objectives

Discuss the database structure.

Fields

A field is the smallest logical structure that is used in a Microsoft Dynamics NAV 2013 database. A field contains a single piece of information, such as a name or an amount. A field can contain only one specific type of information.

Fields are assembled into a structure called a record. A field can hold only a limited amount of information. By grouping these fields into records, you create a more flexible, organized information structure.

Records

A record is a logical structure that is assembled from an arbitrary number of fields. Fields are used to store a single entry in a database. The fields in a record store information about entity properties that are represented by the record. Records are organized in tables.

Tables

Think of a table as a matrix. Each row describes a record and each column describes a field in the record. Tables are organized in companies.

Companies

A company is the largest logical structure that is used in a Microsoft Dynamics NAV 2013 database. A company may be considered as a subdatabase; its primary use is to separate and group large sections of data in a database. A company can contain private tables and tables that are shared with other companies.

The following figure shows the logical structure of the data in a Microsoft Dynamics NAV 2013 database.

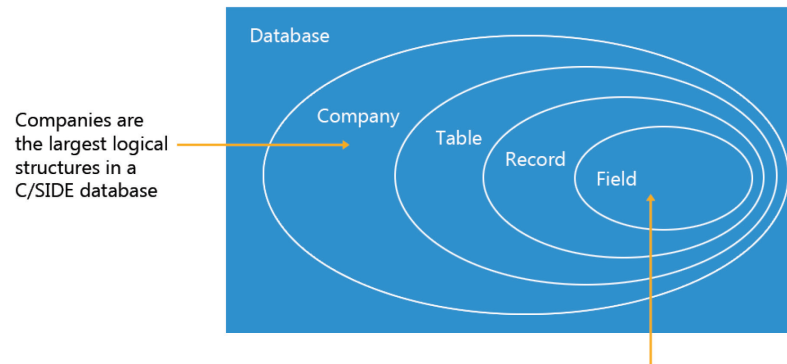


FIGURE 1.12: LOGICAL DATABASE STRUCTURE

Lab 1.1: Designing and Running an Object

Scenario

Isaac is a developer for Cronus International Ltd. and is learning how to design objects in Microsoft Dynamics NAV Development Environment. He creates an empty page and saves it to test the functionality of the **Object Designer**.

Exercise 1: Accessing the Object Designer

Exercise Scenario

Isaac accesses the Object Designer.

Task 1: Start Microsoft Dynamics NAV Development Environment

High Level Steps

1. Start Microsoft Dynamics NAV Development Environment from Windows.

Detailed Steps

1. Start Microsoft Dynamics NAV Development Environment from Windows.
 - a. Click **Start > Microsoft Dynamics NAV 2013 Development Environment**.

Task 2: Open a database

High Level Steps

1. Open the Demo Database NAV (7-0) database that is located on NYC-DCSQL-NAV server.

Detailed Steps

1. Open the Demo Database NAV (7-0) database that is located on NYC-DCSQL-NAV server.
 - a. Click **File > Database > Open**.
 - b. In the **Open Database** window, in the **Server Name** field, enter "NAV-DC-SQL-SP\NAVDEMO".
 - c. Make sure that **Authentication** is set to **Windows Authentication**.
 - d. In the **Database Name** field select "Demo Database NAV (7-0)".
 - e. Click **OK**.

Task 3: Start the Object Designer

High Level Steps

1. Start the **Object Designer**.

Detailed Steps

1. Start the **Object Designer**.
 - a. Click **Tools > Object Designer**.

Results

Object Designer opens in Microsoft Dynamics NAV Development Environment.

Exercise 2: Creating an object

Exercise Scenario

Isaac creates an empty page.

Task 1: Create a new page

High Level Steps

1. Create a new blank page.
2. Save the page as 90011, Test Page.

Detailed Steps

1. Create a new blank page.
 - a. In Object Designer, click **Page**.
 - b. Click **New**. The **New Page** window opens.
 - c. In the **New Page** window, select **Create blank page**, and then click **OK**. The **Page Designer** window opens.
2. Save the page as 90011, Test Page.
 - a. Click **File > Save**. The **Save As** window opens.
 - b. Type "90011" in the **ID** field, and "TestPage" in the **Name** field, and then click **OK**.

Task 2: Verify that the page is created

High Level Steps

1. Close the **Page Designer**.
2. Filter the list of pages to display only changed pages.

Detailed Steps

1. Close the **Page Designer**.
 - a. Click **File > Close** to close the **Page Designer** window.

2. Filter the list of pages to display only changed pages.
 - a. In Object Designer, click **View > Table Filter**.
 - b. In the **Field** column of the **Object – Table Filter** window, select **Modified**.
 - c. In the **Filter** column, enter “Yes”, and then click **OK**.
 - d. Verify that page **90011, Test Page** is displayed.

Results

Page 90011, Test Page.

Module Review

Module Review and Takeaways

A thorough understanding of Microsoft Dynamics NAV Development Environment enables developers to streamline their development process and to develop customizations and integrations for Microsoft Dynamics NAV 2013. The Object Designer is the central feature of Microsoft Dynamics NAV 2013 that enables developers to create new objects and change existing objects. Object Designer also enables multiple developers to work at the same time without accidentally overwriting another developer's work. This course explains how to use the Microsoft Dynamics NAV Development Environment, how to create and use the basic objects, and how to implement best practices to make sure smooth software upgrades in the future.

Test Your Knowledge

Test your knowledge with the following questions.

1. What are the elements of the logical database structure in Microsoft Dynamics NAV 2013?

2. Which object type is new in Microsoft Dynamics NAV 2013?

- () Page
- () XMLport
- () MenuSuite
- () Query
- () OData Web Service

3. Can another developer overwrite an object locked by you?

- () No
- () Yes
- () Only administrators can overwrite objects locked by other users

Module 1: Microsoft Dynamics NAV Development Environment

- Only users with SUPER permission set can overwrite objects locked by other users
4. Which object type do you use to enable users to view and enter information in Microsoft Dynamics NAV 2013?
- Table
- Page
- Report
- XMLport
- Query
5. You can run objects that are not compiled.
- True
- False

Test Your Knowledge Solutions

Module Review and Takeaways

1. What are the elements of the logical database structure in Microsoft Dynamics NAV 2013?

MODEL ANSWER:

Field, record, table, and company.

2. Which object type is new in Microsoft Dynamics NAV 2013?

Page

XMLport

MenuSuite

Query

OData Web Service

3. Can another developer overwrite an object locked by you?

No

Yes

Only administrators can overwrite objects locked by other users

Only users with SUPER permission set can overwrite objects locked by other users

4. Which object type do you use to enable users to view and enter information in Microsoft Dynamics NAV 2013?

Table

Page

Report

XMLport

Query

Module 1: Microsoft Dynamics NAV Development Environment

5. You can run objects that are not compiled.

True

False

