Microsoft Business Solutions

Terminology Handbook

MICROSOFT BUSINESS SOLUTIONS–NAVISION

TERMINOLOGY HANDBOOK

DISCLAIMER

This material is for informational purposes only. Microsoft Business Solutions ApS disclaims all warranties and conditions with regard to use of the material for other purposes. Microsoft Business Solutions ApS shall not, at any time, be liable for any special, direct, indirect or consequential damages, whether in an action of contract, negligence or other action arising out of or in connection with the use or performance of the material. Nothing herein should be construed as constituting any kind of warranty.

COPYRIGHT NOTICE

TRADEMARK NOTICE

Published by Microsoft Business Solutions ApS, Denmark.

Published in Denmark 2003.

DocID: NA-370-IGU-001-v01.00-W1W1

PREFACE

This book is intended as the basis for a training course aimed at terminologists working with Microsoft® Business Solutions–Navision®. It can also be used as a tool by anyone working with naming objects in Navision.

The Terminology Handbook explains the concept of terminology and how we work with it. It also describes the underlying structure of Navision and outlines our naming conventions for objects.

The information in this book supplements the information about naming in the GUI guide and the C/AL™ Programmer's Guide. It is aimed at Microsoft® Business Solutions employees but also has relevance for anyone working with customizing or localizing Navision.

TABLE OF CONTENTS

Table of Contents

Chapter 1
General Principles

In this chapter, we will define terminology in general and look specifically at how we work with it.

We will look at two aspects of terminology:

- Basic Terminology
- Application Terminology

1.1 Definition of Terminology

According to the Random House Webster's Unabridged dictionary, terminology is:

the system of terms belonging or peculiar to a science, art, or specialized subject; nomenclature: the terminology of botany.

In the same dictionary, term is defined as:

a word or group of words designating something, esp. in a particular field, as atom in physics, quietism in theology, adze in carpentry, or district leader in politics.

We are interested in establishing terms for concepts to ensure clear and unambiguious communication. A term is the linguistic representation of its concept. For our purposes, it is important to remember that a term is linked to its context.

1.2 BASIC TERMINOLOGY

Basic terminology consists of the fundamental terms used to describe a application area. It is established in the envisioning and planning stages of a project. The list of basic terminology for a feature is included as a section in the functional specification. The teams provide lists of new terminology (with definitions) and have the terms entered into the terminology database.

The following are some things to consider when choosing basic terminology:

- Correctness: Is a term the right term? Is it the term that people in the relevant field use? Is it the standard US term for the concept?
- Terminology standard: If there is a terminology standard that we have decided to use (APICS for example), is the term the one used in the standard?
- Spelling: If there is more than one acceptable spelling for a term, we must choose one and use it consistently.
- One term for one concept: If there is more than one correct term for a concept, we must choose one of them and use it exclusively. (For example, replan versus reschedule in manufacturing).
- One concept for one term: If a term can mean more than one thing, we should try whenever possible to find an appropriate alternative term for one of the concepts. If there is no alternative term, both definitions must be included, and there must be an explanation of the different contexts the term is used in (this is very important for localization).
- Existing terminology: How does the term fit in with existing terminology? Is the same term used elsewhere in the application to mean something else? Is a different term used elsewhere in the application to mean the same thing? If so, we may have to pick a different term or change the existing terminology. (All new terms must be checked in the terminology database.) An example is safety stock vs. inventory buffer.
- Abbreviations: Will we need to abbreviate the term? If so, is the abbreviation a standard US abbreviation? Is it consistent with abbreviations of the same term elsewhere in the program? Is the same abbreviation used for another term elsewhere in the program? (All new abbreviations must be checked in the terminology database.) Abbreviations are often added when you start working on application terminology because of the restrictions on the numbers of characters that can be used.

The following are some things to avoid:

- Translating directly: Direct translation from Danish (or another language) often results in an incorrect term.
- Making up terms: Always assume that there is an established term for a concept and try to find out what it is.
- British English: The British term for a concept is often different than the US term for the same concept.

Terminology Lists

When you write a terminology list, you should compose your definitions according to the following guidelines:

- A definition should be a single, simple sentence containing the characteristic features of the concept.
- The definition should be clear and adequate - no more and no less information than is necessary.
- Use clear words in the definition that are either of a general nature, familiar to the user or defined elsewhere in the terminology database.
- One concept - one term: give the term a unique identification.
- Do not include general terms that are not specific to the application area. If a term can be used in the same way and mean the same thing generally, then it doesn’t belong on a terminology list.
- Terms should be written in lowercase letters unless they are always capitalized (for example, Web or Internet).
- Style: use Arial font size 12, with the definition in bold (see the following example).

EXAMPLE

annuity loan

A loan that is repaid with the same amount of repayment (consisting of interest and principal) at each due date.

Things to avoid when defining a term

Avoid the following:

- Repeating the term within the definition

EXAMPLE

macro: A macro is a particular set of instructions that are carried out ....

- Over-defining a term

EXAMPLE

mortgage debt: A name for debt for which property is pledged.

- Circular definitions (defining the term by using the term itself)

EXAMPLE

pledge: If a company has pledged assets as collateral for a loan this must be disclosed...

- Non-definitions

Giving additional information about the term but not defining the term.

EXAMPLE

notes: Notes make up an integrated part of a company's financial statements.

- Classification

The terms you are describing will already be classified according to subject matter. Therefore you do not need to include a classification in the definition.

EXAMPLE

notification: A notification in Relationship Management is an e-mail message that is....

Working with Basic Terminology Lists

We should update the basic terminology lists on a regular basis and put them in VSS in the terminology folder (PD Document Database/Navision Financials®/Function Groups/User Education Group/Language and Terminology/Terminology Lists). When a product is released, the list is taken from VSS and shipped with the product CD.

1.3 Application Terminology

The following table illustrates the relationship between the terms on the basic terminology list and the application terminology:

Basic term (with definition)	Related application terms (with identifier)
customer: A customer in Microsoft Business Solutions–Navision is the term for a person or company for which there is, has been or will be a sale either on credit or in cash. (abbreviation: cust.)	T_18: "Customer" (table) T_21_3: "Customer No." (field) F_21: "Customer Card" (form) R_101: "Customer - List"
prospect: In Navision, a prospect is usually the name for a prospective customer, but a prospect can also be a prospective vendor, an established customer or vendor, a competitor, dealer, agent, consultant or partner.	T_5001: "Prospect Status" (table) F_5001: "Prospect Card" (form) R_5000: "Prospect - Profile" (report) D_5000: "Import/Export Prospect" (dataport) T_5002_1: "Prospect No." (field)

Always Check

Always check terms with the terminology lists or in the terminology database. New terms should be cross-checked in the terminology database to make sure that they aren't used with a different meaning elsewhere.

What Does it Do and Where?

A rule of thumb in good terminology work is to always remember to ask, "what is it?" A name can sound perfectly reasonable and yet the programmer could have meant something entirely different than what you thought. It's important to see the named object in context.

Long Adjective Strings

Try to avoid long strings of adjectives, for example, Custom Message Record Variable Filters (EDI990542)! They are hard to translate, but sometimes they are unavoidable.

How Specific Should a Name Be?

Objects

Give names that are specific enough to identify the object in question. It's not enough to just say "information" or "entries." Here are some examples:

- Company information, Confidential Information
- Production Order Component, BOM Component
- EDI Event Log, Message Log (Web Shop)

Fields

Field names are sometimes less specific because they are modified by the table or window name. For example, on the Customer Card: No., Name, Address, Address 2.

Object Names Must Be Unique

Two objects of the same type can’t have the same name.

Standard Heading Capitalization

When naming, use standard heading capitalization: capitalize all nouns, verbs, adverbs and pronouns. Capitalize the first and last words, regardless of their part of speech. Do not capitalize coordinate conjunctions, prepositions and "to" in an infinitive phrase. Watch out for hyphenated words; only capitalize the second word if it is a proper noun, adjective or the words have equal weight.

Chapter 2

The Structure and Components of Microsoft® Business Solutions–Navision®

In order for us to name things in the program correctly, we must first understand how the program is built up.

In this chapter, we will take a look at Navision and its components. We will look at the difference between the database and the application objects that make up Navision:

- Tables
- Forms
- Reports
- Codeunits
- Dataports

2.1 THE DATABASE

When we talk about the database, we simply mean a collection of data. When we talk about the application, we mean data that is tied together to form a coherent whole.

The Navision database consists of data that is gathered into records, which are stored in tables. The tables make up the database. The database is then divided into companies. Cronus is the demonstration company.

2.2 Application Objects

In order to be able to work with application terminology, it is important to have a basic understanding of the components that make up Navision and how Navision is structured. Navision consists of five types of application objects:

- Tables
- Forms
- Reports
- Codeunits
- Dataports

For our purpose, it is enough to say that an object is a self-contained component of the program that can contain both data and procedures (code) for manipulating that data.

Tables are the fundamental objects. They store the actual data, which is organized within the tables in individual records.

Forms are used to access the data in the tables. They are used to insert, modify and view the data in the tables.

Reports are used to present data that contains summary information, for example, a list of customers. Batch jobs are also a type of report.

Codeunits contain code that is used in many application objects. (Note: C/AL is the programming language Navision is programmed in, so C/AL code is the type of code contained in the codeunits.)

Dataports are used to import and export data from other applications, for example, a list of potential contacts for the Contact Management application area from an external source.

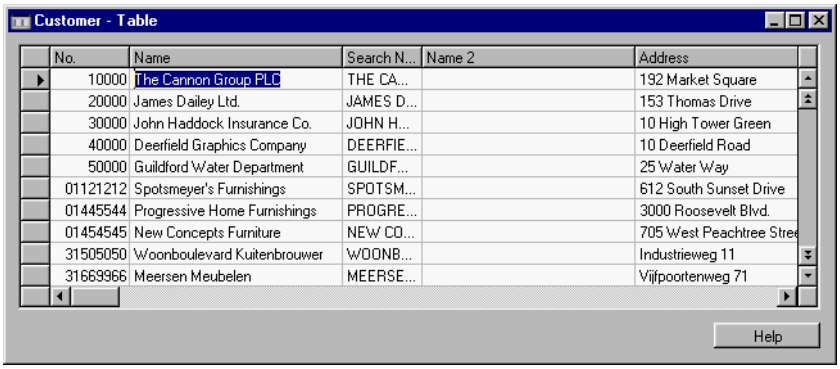
Tables - the "Building Blocks"

What Is a Table?

All data in Navision is stored in fields within tables. Every field that you see in a window, a report or anywhere else is a field in a table that you are viewing through the window, report, or so on. End users do not see the tables.

What does a Table look Like?

A table consists of rows (lines) and columns. Each row is one record (for example, all of the information about a particular customer). Each column is one field (for example, the **No.** field containing the number of the customers whose records are on the various rows). The ***Customer*** table is an example of a table:



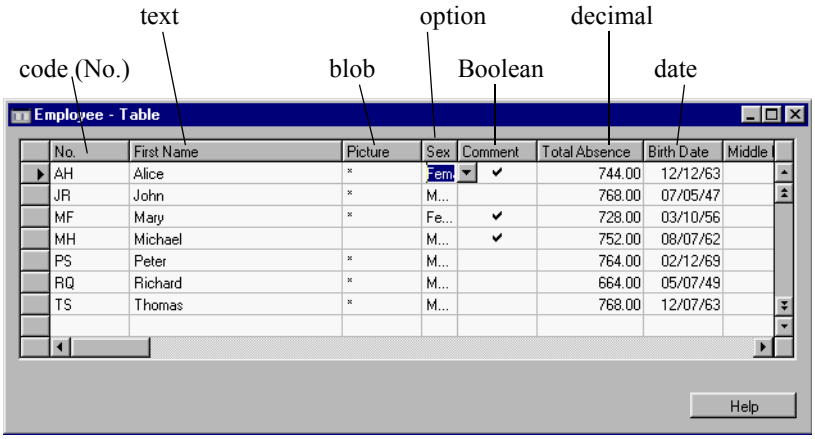
Components of Tables

Tables consist of things that are both visible to end users (through other application objects) and things that are not visible. The following are visible table components that we are involved in naming:

- Table Name
- Fields
- Text Messages (examples from a blanket sales order)

EXAMPLE

There are different types of fields and the field type is sometimes relevant to how the field is named. The following picture shows some of the different types of fields:



Here is a list of different fields:

- Integer
- Option (option string)
- Boolean (check box)
- Date
- Code
- Blob (binary large object, bitmaps pictures)

- Time
- Decimal
- FlowField®
- FlowFilter®

We will only discuss field types in the context of their relevancy for naming.

Variables

Variables are not visible to end-users, but we are involved in naming them. A variable is a name for something in the code that can have different values. For example, in the code that calculates the VAT amount for items, there is a variable called VAT%. The formula is Price * VAT % = VAT amount. Because Navision is used in different countries, the VAT percentage is variable and therefore a variable is used to represent the VAT percentage in the code. If, on the other hand, you were only interested in calculating VAT in Denmark you might choose to use a constant instead of a variable, for example, 0.25. However, this would make your code less flexible: it would have to be changed if the Danish VAT % was changed.

Variable names are a string of words, often abbreviated, pushed together with the first letter of each word capitalized regardless of what part of speech it is. Here are some examples of variable names taken from a log:

```
**************************
Variables:
**************************
--------- Before --------
T27-V24-P2818-L30: NoSeriesMgt
T27-V25-P2818-L30: MoveEntries
T27-V26-P2818-L30: MfAddOnIntegrMgt
T27-V27-P2818-L30: CurrencyIsFound
--------- After ---------
T27-V24-P2818-L30: ItemCategory
T27-V25-P2818-L30: UserSetup
T27-V26-P2818-L30: ItemSubstitution
T27-V27-P2818-L30: NoSeriesMgt
T27-V28-P2818-L30: MoveEntries
```

The naming of variables is important because the variable tells the program where to go to get the relevant data. When variables are named or abbreviated inconsistently, it can cause mistakes in the translated code.

Forms (Windows)

Forms are the windows through which the end user views the tables. Here, the user can enter, modify and view data stored in the fields in the tables. An example of a form is the customer card.

The fields in the form are the same as the fields in the table. For example, the **No.** field in the customer card corresponds to the **No.** field in the ***Customer*** table. Forms also have buttons and tabs that are not present in the tables.

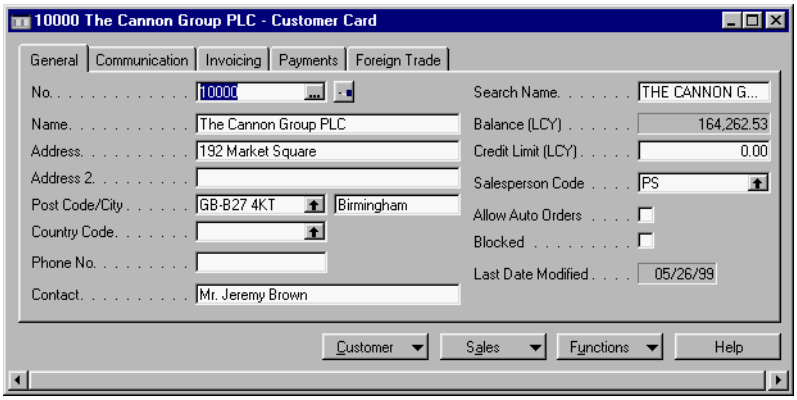
Exercise: Open up a customer card and press F3 to create a new customer. Enter some data about your new customer. Now go "behind the scenes" and open the ***Customer*** table to see how your new customer has affected the data in the ***Customer*** table.

Types of Forms

Form is the technical word for windows. There are various types of forms used in Navision.

Card forms

Card forms look like filing cards and they often have several tabs:



Examples of card forms include the customer card and the item card. They are used to view a single record in the table.

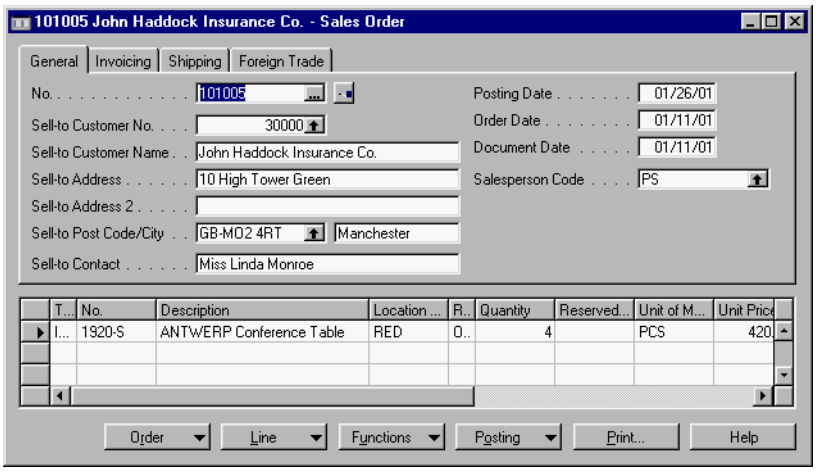
List Forms

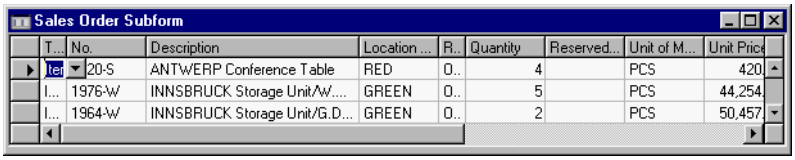
List forms look like tables and are used to view many records at once.

Examples of list forms include the ***Customer List*** window, the ***Item List*** window and the ***Items by Location*** window. List forms are also referred to as tabular forms, which is because they are laid out like a table, in columns and rows.

Main/Subforms

There are also forms that consist of a form and a subform. These are called Main/Subforms:





Examples of Main/Subforms include the ***Sales Order*** window and the ***Sales Order Subform***, and the ***Purchase Order*** window and the ***Purchase Order Subform***.

These type of forms always look up to two tables: a header table and a line table. Header tables provide all the general information, while line tables contain specific information, for example, about an item.

Wizards

Another type of form is a wizard. Wizards guide the user through a complex process by largely automating the task. There are two types of wizards: simple wizards and advanced wizards. Simple wizards typically consist of three or fewer pages. Advanced wizards have more pages and often require multiple decision points (multiple paths). The advanced wizard includes a Welcome and a Completion page. Use a conversational rather than an instructional style with wizards; try to avoid sentences like "Choose a layout." but rather make them questions. Boolean fields should not be questions but positive statements.

The Main Menu

The main menu and the menus for the application areas are also forms, as are the request windows for reports.

Often data from one table is displayed in more than one form. For example, the ***Customer*** table is referenced by the ***Customer Statistics*** window (form 151), the

Customer Sales window (form 155), the customer card (form 21), the ***Customer List*** window (form 27) and many more.

Components of Forms (what you see)

The following is a list of possible form components that we are involved in naming:

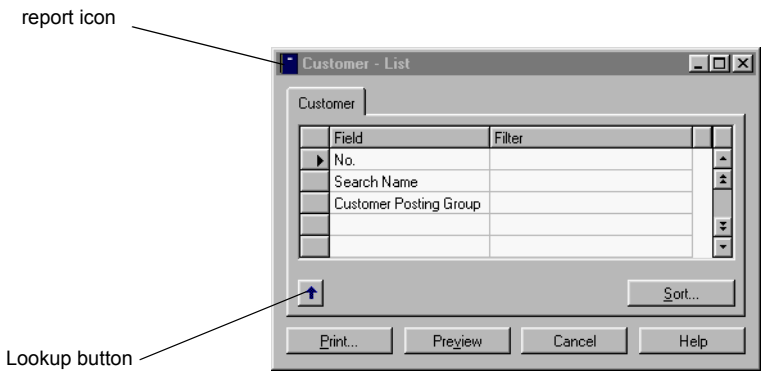
- Form Name
- Fields (Remember, these are really part of a table and are only *displayed* on the forms.)
- Tabs
- Menu items (the buttons at the bottom of the window)

Reports

Reports are used in Navision to display summarized information (for example, a list of customers), to print documents (such as invoices) and to process data (batch jobs). For our purposes we will talk about two different kinds of reports:

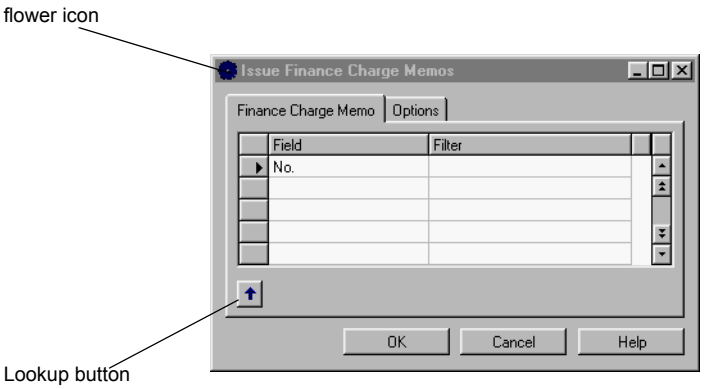
- Reports (On the main menu, reports are listed under Reports.)
- Batch Jobs (On the main menu, batch jobs are listed under Periodic Activities.)

An example of a report is the ***Customer - List*** report:



Like all reports, it has a Print button and a report icon (a picture of a file). You can see that you are dealing with a report and not a batch job by checking for the Print button or the report icon. Reports have tabs and each tab has a lookup button to a form. The tab is named according to the form it looks up to. In the ***Customer - List*** report, the tab is called ***Customer*** and the lookup is to the ***Customer List*** window. There can be several tabs looking up to several different windows.

Batch jobs process data. An example of a Batch job is the Issue Finance Charge Memos batch job:



Batch jobs do not have a Print button, but they always have an OK button. Batch jobs also have a flower icon on the title bar and tabs that look up to other windows (which should be named in the same way as for reports).

The following is a list of report components that we are involved in naming:

- Buttons
- Controls and Labels
- Fields

Codeunits

We are involved in naming the codeunits themselves and in checking variable names and error messages generated by codeunits.

Dataports

We are involved in naming these and their components.

Chapter 3
Tools

In this chapter, we will look at the tools available to help us with terminology work in the application.

The tools we will learn about are:

- The Object Designer
- Show Column and Zoom
- Object Text Files

3.1 The Object Designer

The Object Designer is the tool that developers use when they design an object. Using the Object Designer is also the most accurate way to view an object and this is what we use it for. You can use the object's number and name to find the object. This means that you can be quite certain that you are viewing the correct object.

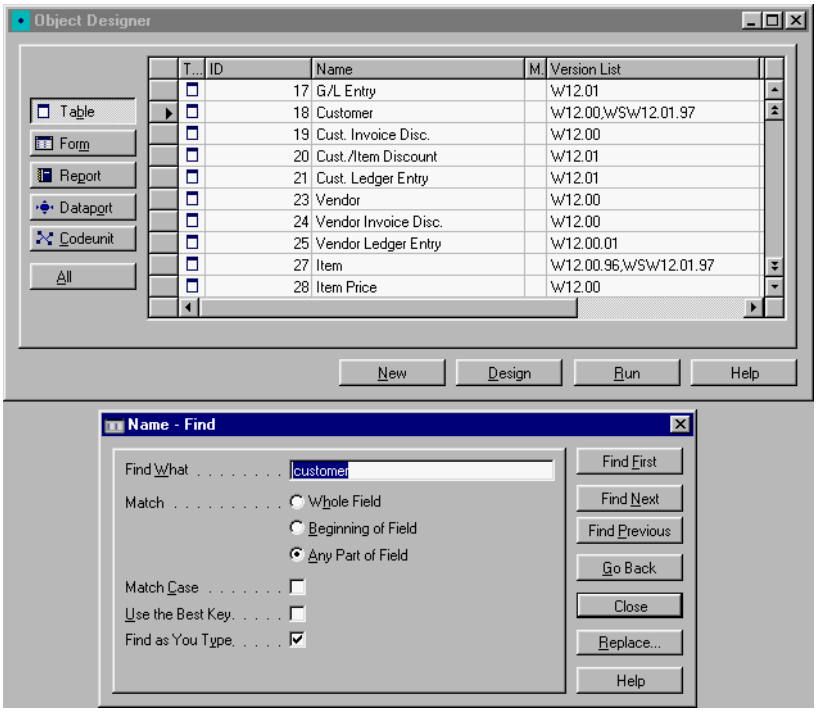
The Object Designer lists all of the objects in Navision and provides information about them. You can see the object type, the object ID (number), the object name, whether the object has been modified and the version list number for the object. The object ID and the object name are most useful for our purposes.

The object ID is a unique number assigned to the object that identifies it. We will look at these more closely when we discuss compare logs and the object text lists in Chapter 7.

You can use the Object Designer to view objects in two ways. You can:

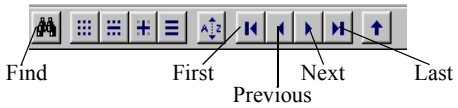
- Click the Run button to see the actual object.
- Click the Design button to see the design view. This tells you important information about the design of the object, for example, what the field types are.

Using the Object Designer



Searching

On the main toolbar, there are the following buttons that can be used for searching:



Click the Find button to use the ***Find*** window. You can also simply place your cursor in the field you want to search on in the Object Designer, and start to type the ID or the name of the object that you are looking for. The Find window appears.

Note: don't type if the name is highlighted (blue) as this will rename the object.

Buttons for Sorting

The following buttons in the Object Designer window can be used for sorting:

- Table
- Form
- Report
- Codeunit
- Dataport
- All

Buttons for Viewing Objects

- Run
- Design

Using the Object Designer with Tables

Clicking the Run button shows you the table laid out in the usual columns and rows.

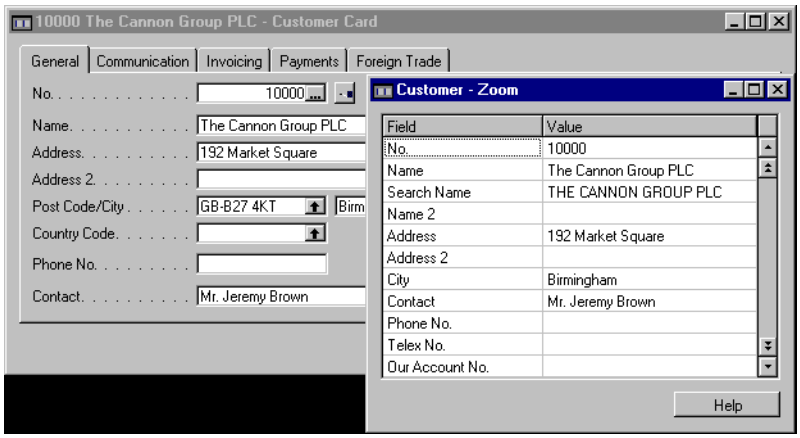
Clicking the Design button shows the design of the table with the ID numbers of the individual fields. You can also see what the field types are. This is useful when you are checking the name of a field and you need to know if it is an option field or a boolean field.

3.2 SHOW COLUMN AND ZOOM

Not all fields are part of the standard form setup. To look at fields that have been hidden, on the menu bar click View, Show Column. The ***Show Column*** window appears which shows the field names of all the columns in the window. If you want a column to be visible in a tabular form, click to insert a ✓ in the check box to the left of the field name.

You can use Zoom to display the source table and all the fields associated with forms. Open the relevant window and then on the menu bar, click Tools, Zoom (or CRTL+ F8), the ***Zoom*** window appears. The name of the source table is displayed on the title bar.

In the following example, we can see that the customer card is based on the ***Customer*** table:



3.3 OBJECT TEXT FILES

Object text files contain extracted text from the Navision Objects with their IDs. The following is an example of extracted text from the Customer table:

```
T18-F1-P2818-L30:No.
T18-F2-P2818-L30:Name
T18-F3-P2818-L30:Search Name
T18-F4-P2818-L30:Name 2
T18-F5-P2818-L30:Address
T18-F6-P2818-L30:Address 2
T18-F7-P2818-L30:City
T18-F8-P2818-L30:Contact
T18-F9-P2818-L30:Phone No.
T18-F10-P2818-L30:Telex No.
T18-F14-P2818-L30:Our Account No.
T18-F15-P2818-L30:Territory Code
T18-F16-P2818-L30:Department Code
T18-F17-P2818-L30:Project Code
T18-F18-P2818-L30:Chain Name
T18-F19-P2818-L30:Budgeted Amount
T18-F20-P2818-L30:Credit Limit (LCY)
T18-F21-P2818-L30:Customer Posting Group
T18-F22-P2818-L30:Currency Code
T18-F23-P2818-L30:Price Group Code
T18-F24-P2818-L30:Language Code
```

You can use the IDs, together with the Object Designer to find things that already exist in the program. For example, you can find out how a particular word is usually used in the program, locate something you vaguely remember seeing, and so on.

3.4 TOOLS EXERCISES

1 Use the Object Designer to find the names or numbers of these objects:

- Table 90
- The Item table
- Form 346
- Resource Picture

2 Use the object designer to find the names or numbers of these reports and find out whether they are reports or batch jobs:

- Sales Person Commission
- 188

3 Use the Object text file and the Object Designer to find examples of these fields (or fields that contain this text):

- Customer No.
- Comment

What is the name of the table the field belongs to? What is the field number? What kind of field is it?

4 Use the Object text file and the Object Designer to see how the word "priority" is used in the program.

5 Find out the names of the tables the ***Sales Journal*** window and the ***Applied Customer Entries*** window are based on.

Chapter 4
Naming Objects

In this chapter, we will learn about the naming conventions for the different objects in Navision.

We will specifically concentrate on the five types of objects:

- Tables
- Forms
- Reports and Batch Jobs
- Codeunits
- Dataports

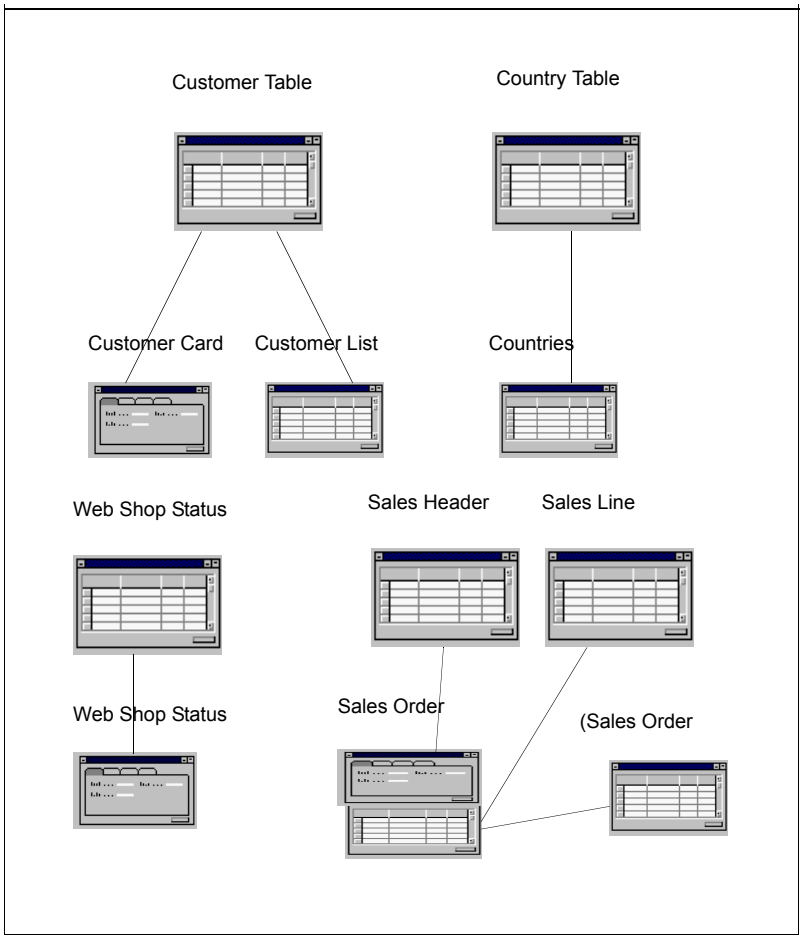
4.1 INTRODUCTION

There are certain rules for naming objects. When we work with terminology, the most important thing is to be consistent. If you notice new rules, write them down.

4.2 TABLES

- Table names are always singular. The name of the table corresponds to one record in the table: an item, a customer, a salesperson, and so on.
- Table names should be as descriptive as possible.
- Tables on which the headers for main/subforms are based should have the word header in the name, for example, the ***Purchase Header*** table(38).
- Tables on which the lines for main/subforms are based should have the word "line" in the name, for example, the ***Purchase Line*** table(39).
- Table names should be easily identifiable as being related to the "main" forms that are based on them. For example, the ***Item Statistics*** window, the item card and the ***Item List*** window are all based on the ***Item*** table.

The following diagram illustrates the relations between tables and forms.



4.3 FORMS

- Form names should be related to the names of the tables they are based on. For example, the ***Item Statistics*** window, the item card and the ***Item List*** window are all based on the ***Item*** table.
- If a table can be accessed by both a card and a list (tabular) form, the word "card" must be part of the name of the card form and the word "list" must be part of the name of the list form. For example, customer card and ***Customer List***.
- Forms that are subforms must have the word "subform" as part of their name. For example, ***Purchase Order Subform*** (54).
- There should be a clear relationship between the name of a form and the "path" you follow to get there. For example, from the item card via Item button, Statistics, Entry Statistics to the ***Item Entry Statistics*** window.
- Setup forms for an application area include the word "Setup."

Types of Forms

Some forms always (or almost always) have names in the plural.

Tabular (list) forms

Tabular forms usually have plural names. The logic behind this is that tabular forms display many records. For example, the ***Salespeople/Purchasers*** window lists all the salespeople and purchasers that have been set up in the program. These windows are often lists of records with "codes."

However, tabular forms with a name that ends with the word "list," "sheet," "worksheet," "journal" or something similar, are singular. Some examples are, ***Requisition Worksheet***, ***Customer List***, ***Item Journal*** and ***Vendor List***.

Note: some tabular forms are noneditable. For example, you can't modify the ***Customer List*** window or the ***Applied Customer Entries*** window as these types of tabular forms (with names ending in List or Entries) are noneditable.

Card Forms

Card forms have singular names because they only display one record. For example, the customer card shows data for one customer at a time.

Wizards

Wizard names can either be a noun and the word Wizard, for example ***Setup Wizard***, or be in the imperative, for example, ***Create Interaction***.

Wizards should generally be named in the imperative because this clearly signals to the user that something will happen as a result.

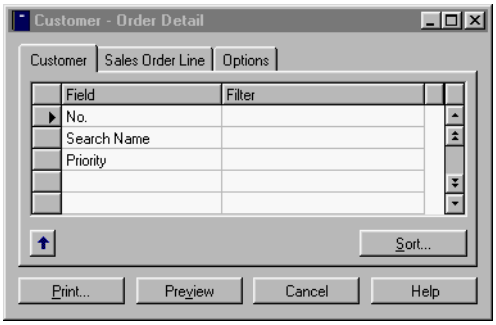
4.4 Reports and Batch Jobs

Reports

- Reports are often given the name of the main area set off from the specific area with a hyphen (not an en-dash.)

  For example: ***Customer - Labels***, ***Customer - Detail Trial Balance***, ***Vendor - List***, ***Vendor - Order Detail***.

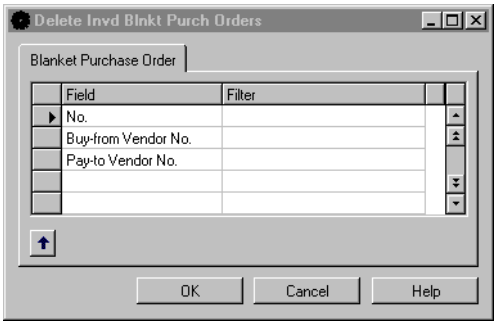
- The report name should also be the same as the heading on the printed report itself. You can see what this is by clicking Preview. The name can be either singular or plural depending on what the report is about. For example, if the report is a list of several things (documents, transactions...) then the name should reflect this and be in the plural.
- Each tab on a report should have a name similar to the form that it looks up to. For example, on the ***Customer - Order Detail*** report, the ***Customer*** tab looks up to the ***Customer List*** window and the ***Sales Order Line*** tab looks up to the ***Sales Lines*** window.



Batch Jobs

The names of batch jobs are in the imperative. For example, ***Combine Shipments***, ***Post Inventory Cost to G/L***.

The names of batch jobs say something about what the batch job does. The rules governing the naming of tabs are the same as for reports. In the following example of a batch job, the ***Blanket Purchase Order*** tab looks up to the ***Blanket Order - Purchase List*** window.


Delete Invd Blnkt Purch Orders
Blanket Purchase Order
Field
Filter
No.
Buy-from Vendor No.
Pay-to Vendor No.
OK
Cancel
Help

4.5 CODEUNITS

The name of a codeunit should be the name of the object that the codeunit processes followed by a hyphen, followed by what the codeunit does (in the imperative). The first part of the name should be abbreviated as a variable with no spaces or periods. The second part of the codeunit name (what is being done) should be an imperative or imperative phrase with spaces and without abbreviation if possible. For example:

VendEntry-Apply Posted Entries

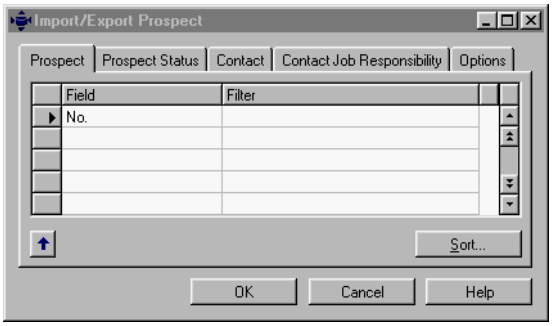
FinChrgMemo-Issue

Notice that there are no spaces between the hyphen and the words on either side of it. When the name of the codeunit includes the word Management, it doesn't have a hyphen or the imperative in the name, for example, AccSchedManagement, which looks more like a variable than a codeunit.

The naming of codeunits is currently very inconsistent in Navision. We should start to use this rule again without correcting old inconsistencies.

4.6 DATAPORTS

Dataports import and export data to and from Navision. For example, you can use them to have an entire price list from a vendor placed in your Item table.



These are few and far between at present, but there will probably be an increasing number of dataports in the future. Names should be descriptive and include Import/Export. As with reports, the tabs should be named according to the window they look up to.

4.7 Exercise: Designing an Application

This exercise will give you a chance to apply some of the things we have learned about naming items in Navision.

Design an application. Your application can do whatever you want it to, and it doesn't have to be logical or technically plausible. Concentrate on naming.

Your application should include:

- At least two tables
- At least two forms (windows)
- One batch job
- One report

Chapter 5
Naming Fields

In this chapter, we will look at how we name fields. There are different naming conventions for each type of field, so we will look at the different types of fields:

- Option Fields (option string)
- Boolean Fields (check box)
- Date Fields
- Code Fields
- Other Fields

5.1 INTRODUCTION

Field names should be logical, intuitive and consistent. We want the user to be able to guess what a field does, just by looking at the name. We should avoid abbreviations as much as possible, and if we have to use them we should try to choose ones that are known to the user or that already exist in the program.

5.2 Option Fields (option string)

These fields have a drop-down AssistButton® ▾ and a choice. The field name often includes a word like "method," "type," or "system."

For example: T99000754-F36-P2818-L30: Simulation Type

T99000754-F36-P10340-L250: Moves,Moves When Necessary,Critical

Exercise: Find Option Fields on the customer card and the item card.

5.3 Boolean Fields (check box)

These are check boxes with or without a ✓. A check mark in the box means "yes" and an empty check box means "no."

If the program inserts a check mark, the field name should preferably be a past participle, for example, "Blocked" or "Tested."On the other hand, if the user inserts the check mark, the field name should be in the imperative, for example, "Block" or "Test.". The name can also be an implied question as with **Reconciliation Account** (T15). Is this a reconciliation account? Yes or no?

Exercise: Find Boolean fields on the customer card and the G/L account card.

5.4 DATE FIELDS

- Include the word "Date" in the field name as in **Posting Date** (T32 Item Ledger Entry).
- An exception is a field that is used to indicate a date interval such as **Allow Posting From**.
- **Starting Date** means the date something is valid from.
- **Ending Date** means the date something is valid until.

5.5 CODE FIELDS

Many records have a code. A code is a unique identifier that is used to avoid having to type the same information in many times. For example, in the ***Country*** table, information is set up about many countries. Therefore, when you set up a new customer, you don't have to put in all the information for their country again. You just attach the country code to the customer's record. Most code fields are called Code, but code fields that represent the main tables (and therefore represented on card forms) and called No. This second type of code field is often also part of a number series.

There are two types of code fields:

- The code field for a record is called just "**Code**" or "**No.**" as in the ***Country*** table and the ***Customer*** table. This type of code field does not have an AssistButton. Fields that "look up" to the code field for a record are named with a combination of the name of the table they look up to and the word "**No.**" or "**Code.**" A lookup AssistButton ⬆ to the right of this type of code field displays information that comes from another table and this can be inserted in the field. For example, the **Country Code** field and the **Bill-to Customer No.** field.

There are some exceptions to this rule:

- "**Code**" can be left out of a field name if there is no room for it.
- No. fields that look up to the Number Series table are called just "**No.**"
- If a field looks up to G/L Accounts, don't use "**Code**," use "**Account**."
- Leave out the word "**Code**" when the field looks up to a posting group. For example, T32 **Item Ledger Entry**.

5.6 Other Fields

Blob (binary large object, bitmaps, pictures)

These type of fields often contain pictures and therefore often have the word "Picture" as part of their names. An example is the Picture field in the Item table.

Time

Use time in the field name, for example, Starting Time, Ending Time (T5406). There are not too many of these.

From/To, Starting/Ending, First/Last and Before/After

From/To	Use From or To when referring to a line in a ledger entry table. For example, **From Entry No.**
Starting/Ending	Starting Date is used as a "valid-from" date; Ending Date means "valid until."
First/Last	First means the earliest. Last means the latest. For example, **Last Sales Inv. No.**
Before/After	Use Before or After with an amount before or after a calculation. For example, **Amount Added Before**.

FlowFilter

These fields must include "Filter" in their names. See p. 54 C/AL Programming Guide.

Text

No general rules for naming. When a text field is linked to a code field (in a lookup window) it can be called either Name or Description.

Decimal

Contain a decimal number. No general rules for naming.

FlowFields

No general rules for naming.

Integer

Contain an integer. No general rules for naming.

Percentage

Fields that contain a percentage must include a % sign. There must be a space between the word and the %.

Quantity

Include the word "Quantity" (or "QTY" if there isn't enough space) in the names of fields that contain a quantity. If a quantity field contains an adjective, the **Qty.** should come at the end of the name, for example, **Unreserved Qty.** If the name contains an adjective and a prepositional phrase, **Qty.** should come first, for example, **Qty. Reserved for Customer**.

Amount

Include the word "Amount" (or "Amt.") in the names of fields that contain an amount. In FlowFields, use "Amounts."

Omit the word Amount in fields that contain one of the following words: Adjust, Balance, Base, COGs, Discounts, Fee, Net Change, Payment, Purchase, Sales, and Usage.

Cost

Include the word "Cost" in the name of a field that contains a cost.

Price

Include the word "Price" in the name of a field that contains a price.

Chapter 6

Naming Buttons, Menu Items, Tabs and the Main Menu

In this chapter, we will look at the rules governing how we name buttons, menu items and tabs in forms and the main menu.

6.1 BUTTONS

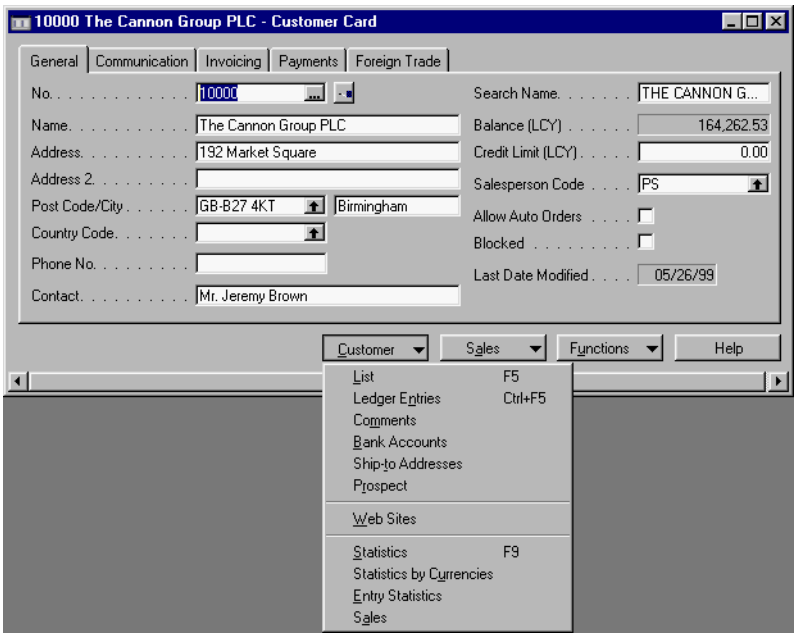
At the bottom of most windows, there are a number of menu buttons. These buttons usually have a drop-down list of menu items underneath them. There are two types of menu buttons, information buttons on the left and action buttons on the right.

Information Buttons

An information button groups menu items that give you information related to the entry. The menu items underneath allow you to view, create or modify information related to the entry.

Most card and list forms have more than one information button that you can click. The first button is given the name of the table the form is based on. For example, on the customer card, there is a information button called Customer, because the customer card is based on the ***Customer*** table. Subsequent buttons on the form have logical names that group the menu items under one heading. Examples of secondary information buttons are Sales (because the menu items underneath all relate to sales information related to the entry) and Line (because the menu items provide you with information about the lines).

Information buttons always have the symbol: ▾ to the right of the name.



Menu Items

Menu items under the information button are named with the second part of the name of the window you are about to open. For example, under the Customer button, there are the following menu items: List, Ledger Entries, and so on. The Statistics menu

item opens the ***Customer Statistics*** window, and the List menu item opens the ***Customer List*** window.

Menu items should preferably be organized in the following way:

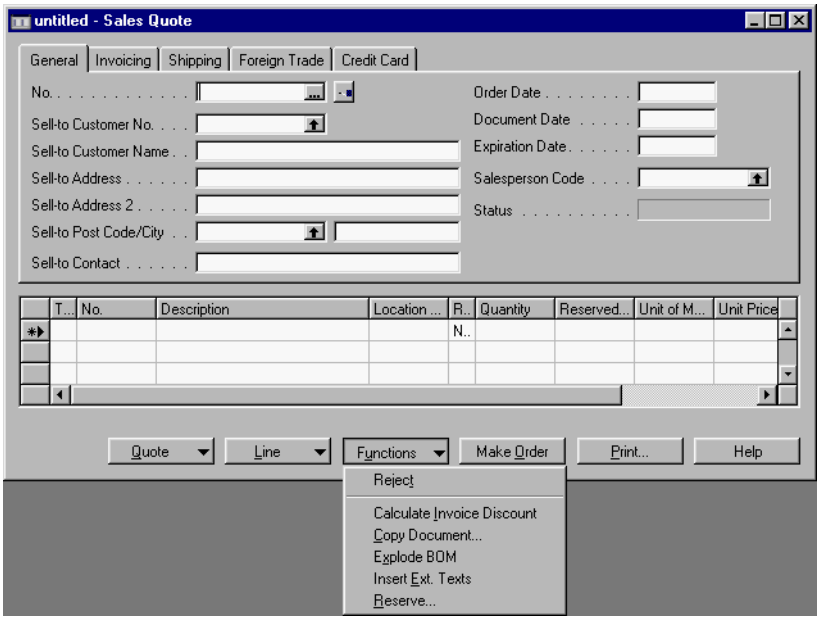
List F5
Ledger Entries
Important information specific to object For example: under the Item button this would include SKU, Variants, Units of Measure, Item Availability by, and so on.
Editable additional information Other information that is less specific to the object.
Document options: Text Translations
Comments Picture
Statistical history
Other kinds of history

Action Choice Button

An action choice button activates something. Its name should be a verb in the imperative. In the Sales Quote window, Print... and Make Order are examples of action choice buttons.

The verb (for example, print) is followed by an ellipsis(…) if you have to provide additional information before an action can be carried out. An ellipsis informs the user that no action will take place before they have entered the relevant information in the window that appears.

If there are many action choices, they can be gathered under a button with a name like Functions (as seen in the Sales Quote window) or Posting. The menu items under this kind of a button are also verbs in the imperative, with or without ellipses.



Exercise: Decide why the various buttons in the Sales Quote window are named as they are.

Wizard Buttons (Navigation Buttons)

Wizards should include the following buttons:

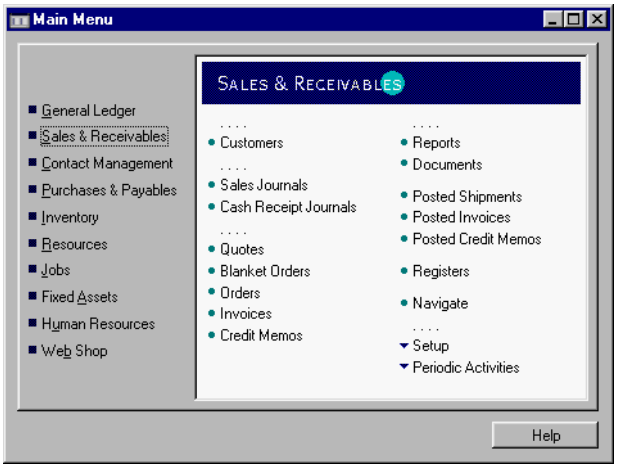
- <Back
- Next>
- Finish (this must only be an active button at a point when the program has enough information to complete the task)
- Cancel

6.2 TABS

For a description of the naming of tabs on Reports, Batch Jobs and Dataports, see Chapter 4 "Naming Objects"

6.3 The Main Menu

Let's look at the main menu and see what kind of generalizations we can make about naming.



The following table illustrates how the main menu is organized.

. . . . **Main Tables** Provide access to the most used of the main tables in the functional area. A selected command always opens a card form. . . . . **Journal Entries** Provide access to the most used journal type . . . . **Document Entries or Main Features** Provide access to the most used document type or to the most used functionality.	. . . . **Reports** One menu item that gives access to the list of reports that can be printed in the application area. **Documents** One menu item that opens the list of documents that can be printed in the application area. **Document History** Provide one menu item for each document type. **Journal History** Provide one menu item for each journal type. **Navigate** Provide one menu item that opens the Navigate form. . . . . **Setup** Menu button with access to all the setup tables. The menu list is ordered by frequency of use. **Periodic Activities** Menu button with access to the seldom or periodically used functionality.

Here we can see that objects are grouped together in a certain way. Notice that all the names are in the plural.

6.4 Exercise: Designing an Application 2

In the previous exercise, you designed an application concentrating mainly on naming the objects. In this exercise, you will be naming fields, buttons and menu items.

Your tables must include:

- Code fields (both lookup and for a record)
- Boolean fields
- Amount fields
- Option fields
- Calculation fields

Your forms must include:

- Tabs
- Buttons: routing choice, action choice and action choice with an ellipsis
- Menu Items

You must show the menu items that you click to open your forms.

You must name your report and batch job and you can include fields, tabs and options if you want to/have time to.

Chapter 7
Logs

A log is a text extract. When we work with application terminology, the document we usually work in is a log. This enables us to identify and compare newly developed objects with established ones.

In this chapter, we will look at how logs are organized and how we work with logs.

7.1 What Is a Log?

All the text in the program is extracted on a given day and then compared to a text file from an earlier version of the program. The result is a compare log (or change log) that shows all the changes made in the application since the earlier text extract. You can also make a log without comparing two files. This type of log will contain all of the text in the application (extracted text). Always use the database from the day of the last text extract to ensure that the text corresponds to the program.

We need to edit these logs on an ongoing basis to ensure that:

- the developers are using the correct terminology.
- there are no spelling or grammatical errors in any of the objects or error messages.
- that abbreviations are consistent throughout the application.
- we are on top of all the changes that developers are making during the Developing phase.

We receive logs automatically at fixed points in the development process, starting in the development phase.

7.2 Order of Objects

Logs are organized in the following order:

1 Tables

2 Forms

3 Reports

4 Dataports

5 Codeunits

The following is an example of extracted text:

```
T2000000005-F9-P10340-L250: ,Yes,Indirect
T2000000005-F10-P2818-L30:Execute Permission
T2000000005-F10-P10340-L250: ,Yes,Indirect
T2000000006-P2818-L30:Company
T2000000006-F1-P2818-L30:Name
T2000000203-P2818-L30:Database Key Groups
T2000000203-F1-P2818-L30:Key Group
T2000000203-F2-P2818-L30:Last Change
T2000000203-F2-P10340-L250: ,Enabled,Disabled
F1-P2818-L30:Company Information
F1-C1-P8630-A1033-L250:General,Communication,Payments,Shipping
F1-C46-P8629-A1033-L250:Post Code/City
F1-C9-P8631-A1033-L250:Hyperlink
F1-C12-P8631-A1033-L250:Hyperlink
F1-C52-P8629-A1033-L250:Ship-to Post Code/City
F1-C28-P8629-A1033-L250:&Picture
F1-C28-P8672-M29-P8629-A1033-L250:Import
F1-C28-P8672-M29-P21225-X1-L132:*.bmp
```

Order within Objects

Tables

The following are the item types found in tables and the order they appear in the logs:

- Name (and number)
- Users for the objects
- Referenced objects
- Changes to name
- Changes to fields
- Changes to text messages
- Changes to variables
- Changes to procedures

The name, users for the objects and referenced objects are supplementary information and not actual text from the program.

Forms

The following are the item types found in forms and the order they appear in the logs:

- Name (and number)
- Users for the objects
- Referenced objects
- Changes to name
- Changes to controls
- Changes to menu item captions
- Changes to text messages
- Changes to variables,
- Changes to procedures.

The name, users for the objects and referenced objects are supplementary information and not actual text from the program.

Reports

The following are the item types found in reports and the order they appear in the logs:

- Name (and number)
- Users for the objects
- Referenced objects
- Changes to name
- Data items
- Fields
- Variables
- Text messages
- Controls
- Functions

The name, users for the objects and referenced objects are supplementary information and not actual text from the program.

Codeunits

The following are the item types found in codeunits and the order they appear in the logs:

- Name (and number)
- Users for the objects
- Changes to name
- Variables
- Procedures
- Functions
- Text messages

Note: Two backward slashes (\\) in a codeunit indicates that it is internal, and therefore it doesn't need checking.

Identifiers

Every item in a log has an identifier. You can often use the identifier to find the item. Identifiers start with a letter that identifies the object type and are followed by an object number:

T = Table

F= Form

R= Report

D= Dataport

C= Codeunit

The object numbers 1-999 belong to the base application design. This also applies to the field numbers. The numbering system is random in tables and forms. However, reports are numbered in intervals, which means we can see which application area a report belongs to by looking at the number. For example, R1-99 belong to the G/L, R100-199 to Accounts Receivable, R200-299 to Sales, and so on. For further information on this, see the C/AL Programming Guide.

The first part of the identifier is followed by more specific information, such as F1, for field 1.

F= Field

V= Variable

C = Control

U = Lots of other things

D,P, S = Other possible letters

The last part of the identifier is a number preceded by the letter "L." The number indicates how many characters long the item can be. For example, in T99000751-P2818-L30: Shop Calendar, L30 means that the table name can have 30 characters. This is particularly interesting to us, because we want to avoid abbreviations where possible. If there's room, write it out.

Changed By

Objects are sometimes followed by "(changed by PHB)" or something similar. This means that the developer with those initials made the change. The information is often useful when you need to ask a question.

Ampersands

An ampersand within a button name in the log name tells the program to insert an access key. In some cases the access key is inserted without using an ampersand, so there is not always an ampersand. The following is a text extract showing ampersands in the following button names: Reserve, Posting, Post, Post and Print, Post Batch, and Line.

```
F52-C52-P8672-M191-P8629-A1033-L250:&Reserve
F52-C52-P8672-M93-P8629-A1033-L250:Apply Entries
F52-C57-P8629-A1033-L250:P&osting
F52-C57-P8672-M58-P8629-A1033-L250:Test Report
F52-C57-P8672-M59-P8629-A1033-L250:P&ost
F52-C57-P8672-M60-P8629-A1033-L250:Post and &Print
F52-C57-P8672-M61-P8629-A1033-L250:Post &Batch
F52-C99-P8629-A1033-L250:&Line
```

Variables

Variables are written as one word and each new word starts with a capital letter. Abbreviations must be used consistently. Everything else is the responsibility of the programmer.

The same goes for other similar things, such as procedures.

Error Messages

Variables in error messages, such as 1% or 2%, are replaced with something, often a field name. You can find the replacement in the code or ask a programmer. The following are examples of error messages.

```
**************************

Text messages:

**************************

--------- New -----------

T99001969-F21-P4962-X1-L132: This item %1 %2 is not set up for location %

T99001969-F55-P4962-X1-L132: This item %1 %2 is not set up for location %

T99001969-U3-X1-L132: You are attempting to pack more of item %1 %2 than
scheduled to ship.
```

Before, After, New, Deleted

Items in the log are labeled as follows:

- Before
- After
- New
- Deleted

Things that have only been moved

There are sometimes items in the log that have not been changed, but only moved. You can recognize them because:

- The only change is in the identifier.
- They appear under the headings Before and After.

If you have a Before/After section where most of the changes are not real changes, you will be able to recognize real changes by the "(changed by)" following the item.

7.3 WORKING WITH LOGS

When you work in a log, do the following:

- Delete things that are OK.
- Write comments for things that need to be changed under the heading Change. For example:

```
Change
R99002587-P2818-L30: Dead Stock List
Change to: Dead Stock - List
```

- Write comments for things you are unsure about under the heading Question.
- Write comments for suggestions under the heading Suggestion.
- Group similar comments together if you can.
- Go over your comments with the developer and give the developer a copy.

We may have to start grouping by developer when we are working in logs that include many different features.

7.4 TERMINOLOGY REVIEW CHECKLIST

Tables	✔
Are the table names singular?	
Do date fields contain the word Date in the name?	
Do code fields contain the word Code in the name?	
Do quantity fields contain the word Quantity or Qty. in the name?	
Do cost fields contain the word Cost in the name?	
Do price fields contain the word Price in the name?	
Do percentage fields contain a % sign?	
Do Boolean field names imply a question, or are they in the imperative? Does the name reflect whether the program fills in the information or that the user does this?	

Forms	✔
Are the forms named correctly? Do card forms contain the word card, list forms the word List, and so on?	
Does the first button have the same name as the window it appears in?	
Are the menu items named correctly?	
Are menu items placed in the correct order?	
The main menu: are all the names of the items in the plural? Are they grouped correctly?	

Reports and Batch Jobs	✔
Are reports hyphenated correctly?	
Are the tabs and the printed documents named correctly?	
Are the names of batch jobs in the imperative?	

Codeunits	✔
Are these hyphenated and named correctly?	

Variables	✔
Are these abbreviated correctly and consistently?	

INDEX