

Microsoft Business Solutions–Navision Developer’s Toolkit

MICROSOFT BUSINESS SOLUTIONS–NAVISION

**MICROSOFT[®] BUSINESS SOLUTIONS–NAVISION[®]
DEVELOPER'S TOOLKIT**

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.

The example companies, organizations, products, domain names, email addresses, logos, people and events depicted herein are fictitious. No association with any real company, organization, product, domain name, e-mail address, logo, person, or event is intended or should be inferred. The names of actual companies and products mentioned herein may be the trademarks of their respective owners.

COPYRIGHT NOTICE

Copyright © 2005 Microsoft Business Solutions ApS, Denmark. All rights reserved.

TRADEMARK NOTICE

Microsoft, Great Plains, Navision, FRx, AssistButton, C/AL, C/FRONT, C/ODBC, C/SIDE, FlowField, FlowFilter, Navision Application Server, Navision Database Server, Navision Debugger, Navision Financials, Microsoft Business Solutions–Navision, SIFT, SIFTWARE, SQL Server, SumIndex, SumIndexField, Windows, Windows 2000, Windows 2000 Server, Windows XP are either registered trademarks or trademarks of Microsoft Corporation or Great Plains Software, Inc., FRx Software Corporation, or Microsoft Business Solutions ApS or their affiliates in the United States and/or other countries. Great Plains Software, Inc., FRx Software Corporation, and Microsoft Business Solutions ApS are subsidiaries of Microsoft Corporation.

DocID: NA-400-DVG-002-v03.00-W1W1

PREFACE

This book is a manual for Microsoft® Business Solutions-Navision® Developer's Toolkit. It describes various aspects of using Navision Developer's Toolkit in your daily work.

The manual provides an overview of what Navision Developer's Toolkit contains, what it can do and how it works. All examples you find in this manual are based on Microsoft Business Solutions-Navision version W1 4.00.

Because the book just gives an overview, it does not cover every detail of Navision Developer's Toolkit. If you need to find out more about certain fields or windows that are not described in this book, you can use the online Help, which provides guidance for all the windows in the program.

Using the Navision Developer's Toolkit manual will be easier if you are familiar with the symbols and typographical conventions used in them. The list below indicates how various fonts, type styles and symbols are used to refer to elements of the program:

| Appearance | Element |
|----------------------|--|
| CTRL | Keys on the keyboard. They are written in small capitals. |
| Description | Field names. They appear in bold and start with a capital letter. |
| Source Finder | Names of tables, windows, boxes and tabs. They appear in bold italics and start with a capital letter. |
| <i>Customer</i> | Text that you must enter, for example: "...enter Yes in this field." It is written in italics. |
| <code>fin.flf</code> | File names and commands. They are written with the Courier font and lowercase letters. |
| <i>User's Guide</i> | Names of manuals are written in italics. |
| ↑ ↓ ▼ ► ... | The special symbols that can be seen in the windows on the screen. |

Note, Tip or Warning

.....
Sections that look like this one, delimited by dotted lines, contain advice about what you should or should not do, or information about something to which you should pay special attention.
.....

TABLE OF CONTENTS

| | |
|---|-----------|
| Chapter 1 Introduction to Navision Developer's Toolkit | 1 |
| The Concept of Navision Developer's Toolkit | 2 |
| The Features of Source Analyzer | 4 |
| The Features of Compare & Merge Tools | 6 |
| The Contents of This Book | 7 |
| Chapter 2 Navision Developer's Toolkit Database | 9 |
| Creating a Navision Developer's Toolkit Database | 10 |
| Importing Object Data | 11 |
| Setting up the Demonstration Database | 18 |
| Upgrading or Converting an Existing Database | 19 |
| Chapter 3 Source Analyzer | 21 |
| Overview of Source Analyzer | 22 |
| Object Administrator | 24 |
| Object Functions | 26 |
| Object Views | 36 |
| Object Tools | 39 |
| Chapter 4 Examples of How to Use Source Analyzer | 47 |
| Introduction | 48 |
| Viewing Object Data | 49 |
| Analyzing Objects | 51 |
| Searching for Object Data | 60 |
| Chapter 5 Compare & Merge Tool | 67 |
| Overview of Compare & Merge Tool | 68 |
| Merge Setup | 72 |
| Compare & Merge | 75 |
| Compare Two Versions | 79 |
| Export Objects | 82 |
| Chapter 6 Installing and Starting Navision Developer's Toolkit | 83 |
| Installing and Uninstalling Navision Developer's Toolkit | 84 |
| Starting The Program and Connecting to a Database | 86 |

Table of Contents

Chapter 1

Introduction to Navision Developer's Toolkit

Navision Developer's Toolkit is a collection of tools that allows you to analyze and update the structure of a Microsoft Business Solutions-Navision database.

This chapter contains an introduction to the concept and features of Navision Developer's Toolkit and an overview of the contents of this book.

This chapter contains the following sections:

- The Concept of Navision Developer's Toolkit
- The Features of Source Analyzer
- The Features of Compare & Merge
- The Contents of This Book

1.1 THE CONCEPT OF NAVISION DEVELOPER'S TOOLKIT

Navision Developer's Toolkit is designed to help you analyze, customize and update Navision applications. Regardless of whether you are working as a Navision project manager, developer, supporter, trainer or sales person, having the ability to look behind the application is a vital part of being able to create and work with the optimal Navision design.

Examining the structure of a Navision database during development means looking in detail at the relationships between objects, their code and their properties. Such a task can consume a great deal of development time when customizing, developing or updating a Navision database. With customized databases especially, architectural documentation frequently does not exist, and when it does, it is often out of sync with the implemented functionality.

Navision Developer's Toolkit provides you with a collection of analysis and development tools that can help reduce the time required to examine the structure of a Navision database and to update customized solutions.

Note

.....
Before you use Navision Developer's Toolkit, you should read and understand the *Navision Application Designer's Guide*.
.....

The Navision Developer's Toolkit Database

The Navision Developer's Toolkit database is the heart of the application. All information about object data is stored in this database. Navision Developer's Toolkit uses a Navision database.

In working with Navision Developer's Toolkit, you have different options for looking at Navision objects. You can work with object data that specifically comes from your application database. You can also work with object data from the demonstration database.

If you want to work with object data from your application database, you must export all objects from the application database in text format and import this object file into the Navision Developer's Toolkit database. You can also use a running Navision client connection to access the Navision application database directly.

If you have worked with a previous version of Navision Developer's Toolkit, you can upgrade an existing Navision Developer's Toolkit database.

You can access the Navision Developer's Toolkit database as a local or a server database. Navision Developer's Toolkit uses C/Front to access the database.

The Source Analyzer

The Source Analyzer is a graphical cross-reference tool. It provides you with a collection of tools to view and analyze object data from a Navision database. Source Analyzer gives you fast access to the object structure in your application database.

The Compare & Merge Tool

The Compare & Merge tool is designed to compare three different versions of Navision objects and to merge these versions into a fourth version. The Compare & Merge tool is based on the structure of a Navision object. This structure is used to apply compare and merge rules that will lead to an improved and structured merge result.

1.2 THE FEATURES OF SOURCE ANALYZER

Navision Developer's Toolkit - Source Analyzer is a graphical cross-reference tool that makes it easy to view and analyze object data from a Navision database. It gives you fast access to the object structure in your application database.

The following are the major features of Source Analyzer:

Analyze Object Structure

The Object Administrator displays the data of all imported objects. You can use the Object Administrator to view information about an object on all levels such as fields, keys, properties, triggers, controls and procedures. Based on this data, you can analyze the relations between objects by using one of the three basic functions:

- Relations to Tables
- Relations from Objects
- Where Used

You can see the results either in an **Object Tree** or an **Object Diagram** window. You can continue to use the functions on these results to continue your analysis.

View C/AL Code

The C/AL Code Viewer shows the code lines of an object. Forms and reports especially tend to contain a large amount of C/AL code in the various triggers. In the Code Viewer, you always see all code lines in one window. This gives you quick and easy access to code lines even in large objects.

The Method Flow displays C/AL code lines grouped into blocks by keywords. In the **Method Flow** window, you can explore these blocks by expanding them, even if these lines contain a trigger or procedure call to another object.

Source Finder

The Source Finder is a powerful tool for searching in all object data for any specific character string. You can include and exclude data areas where the program should search. For example, you could specify a search only in properties or field names. You can use either the objects from the Object Administrator or the result of another Source Finder search as a data reference.

Documentation

Use the print and preview functions from nearly all windows to print object data, such as C/AL code lines or the results of your analysis.

You can save all Object Views to store the results or to load and print them without processing the whole analysis again.

In addition, you can export all results of your analysis in bitmap or text format, or you can use the clipboard to copy the information into other applications for further documentation or training purposes.

1.3 THE FEATURES OF COMPARE & MERGE TOOLS

Navision Developer's Toolkit - Compare & Merge Tool is a collection of tools designed to help document, compare, upgrade and maintain modified customer solutions. This tool is also useful for implementing or upgrading add-on solutions in a modified customer version.

The following are the major features of Compare & Merge Tool:

Merge Setup

The Merge Setup allows you to set up all information necessary to compare three reference versions and to merge them into a fourth version. Regardless of whether the objects from the reference versions are already imported in the database or you want to import one or more versions before you start the merge process, the Merge Setup is your general entry point for a new merge process. A special Merge Wizard is available to guide you through the Merge Setup.

Compare & Merge Versions

The compare and merge process is based on three reference versions that are compared in order to create a fourth version. The **Compare & Merge** window shows you all four versions in one window. The objects for each object type are listed in a tree structure similar to the existing **Object Administrator** window. Differences are colored and the four columns are synchronized to always show the same detail in each window. This window shows you the automatic merge result and enables you also to edit the merged version.

A separate **Compare & Merge Code Lines** window shows the C/AL code lines for one object for all four versions. This window offers you code line editor functions to do manual changes in C/AL code lines.

Compare Two Versions

This is a special tool for comparing two versions in the Navision Developer's Toolkit database. The objects for each object type are listed in a tree structure to make differences visible not only on a detail level but also on a group or object level.

The program compares C/AL code lines in a separate window that shows only the code lines of one object in two versions.

Export Objects

In the Export Worksheet, you can export all objects from Navision Developer's Toolkit database into either a Navision-format text file or directly into a Navision application database.

1.4 THE CONTENTS OF THIS BOOK

This book contains a short description and some examples of how to use Navision Developer's Toolkit. It does not cover every detail of Navision Developer's Toolkit. If you need to find out more about certain fields or windows that are not described in this book, you can use the online Help, which provides guidance for all the windows and functions in the program.

This book is divided into the following chapters:

- Chapter 1, "Introduction to Navision Developer's Toolkit".
- Chapter 2, "Navision Developer's Toolkit Database", describes how to create an Navision Developer's Toolkit database and how to import object data.
- Chapter 3, "Source Analyzer", describes the basic windows and functions of the Navision Developer's Toolkit - Source Analyzer.
- Chapter 4, "Examples of How to Use Source Analyzer", describes some typical cases from your daily work and how to work with the Source Analyzer.
- Chapter 5, "Compare & Merge Tool", describes the basic windows and functions of the Navision Developer's Toolkit - Compare & Merge Tool
- Chapter 6, "Installing and Starting Navision Developer's Toolkit", describes how to install Navision Developer's Toolkit and how to start the program.

Chapter 2

Navision Developer's Toolkit Database

The Navision Developer's Toolkit database is the heart of the application. All information about object data is stored in this database. This chapter shows you how to create a Navision Developer's Toolkit database and how you can import the object data of your application database.

If you have installed Navision Developer's Toolkit with example files, you can read more about how to create the Navision Developer's Toolkit demonstration database in this chapter.

This chapter contains the following sections:

- Creating a Navision Developer's Toolkit Database
- Importing Object Data
- Setting up the Demonstration Database
- Upgrading or Converting an Existing Database

2.1 CREATING A NAVISION DEVELOPER'S TOOLKIT DATABASE

You must have a database to be able to work with Navision Developer's Toolkit. The installation program does not provide a Navision Developer's Toolkit database automatically. Because Navision Developer's Toolkit uses a Navision database, you need at least a single-user installation of Navision version 4.00 or higher to create and manage the database.

Creating a New Database

You must use a Navision client to create and manage the Navision Developer's Toolkit database.

- 1 Start the Navision client. If you are creating a database for SQL Server, the client must be a Navision SQL client, and you must connect to the SQL Server.
- 2 Create a new database by clicking the File menu, pointing to Database, and clicking New.

If you are not working with SQL Server, then, in the **New Database** window, create the database in the directory where you installed Navision Developer's Toolkit. If you want to import the object data of a standard Navision application database later on, you should use at least 500 MB as the file size for this new database.

If you are working with SQL Server, then, in the **Select Server** window, select the SQL Server. In the **New Database** window, insert a new database name.

- 3 Open the Object Designer and import the file `DevTool.fob` from the directory where you have installed Navision Developer's Toolkit.
- 4 Create a new company. You can use different companies to store object data from different application databases in one Navision Developer's Toolkit database.

Copying a Database

If you have already created a Navision Developer's Toolkit database, you can copy the database. If this database has one or more companies, you must delete these companies in order to have a new and empty database. You do not need to import the file `DevTool.fob` again.

2.2 IMPORTING OBJECT DATA

Usually, you have an application database with different object types. To analyze the structure of this database and the relations between the application objects, you must import the object data from the Navision application database into the Navision Developer's Toolkit database.

You can either import these objects from an object file in text format or you can use a running Navision client that is connected to the application database to transfer objects directly into the Navision Developer's Toolkit database.

If you choose to import object data from a text file, you must first export these objects from the Navision application database into an object file in text format.

Exporting Application Objects

To create an object file with all object data, do the following:

- 1 Start the Navision client.
- 2 Connect to your application database and open the **Object Designer** window.
- 3 Click All on the left side of the **Object Designer** window and mark all objects.
- 4 Export these objects to an object file using text format.

You do not always need to export all objects of your application database. For example, if you have changed only some objects and you want to analyze them, you can select and export only these objects.

Note

.....
 If you want to use objects in a compare and merge process, you must export all object versions with the same Navision client version (for example Navision 4.00), because of changes in the Navision text file format between the versions.

Importing Object Data

Before you can analyze the application database objects, you must import the object data from a Navision application database into the Navision Developer's Toolkit database.

You must always have an import version before you can import object data in the Navision Developer's Toolkit database. You can import the object data into an existing import version, or you can create a new import version.

Creating a New Import Version

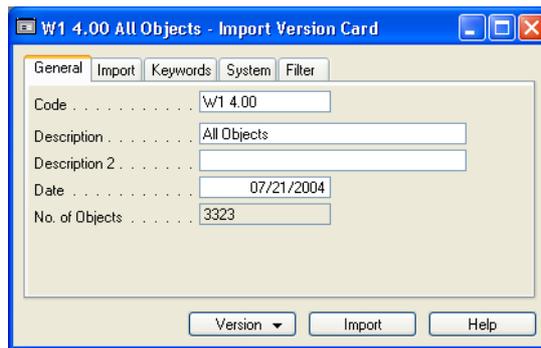
In the import version, you must define some settings before you can start the object import.

Note

Before you can create an import version in a new database, you must create a company in the Navision Developer's Toolkit database. You should use at least one company per customer database or different application languages. You can also use different databases, each with one company.

To create a new import version, do the following:

- 1 Start Navision Developer's Toolkit.
- 2 On the File menu, point to Database and click Open to open a Navision Developer's Toolkit database or to connect to a Navision Developer's Toolkit server database.
- 3 On the File menu, click Import. The **Import Version Card** window appears.



Fill in the fields as necessary, according to the guidelines below. Some fields have no guideline because they are self-explanatory or because they are already filled in:

| Field | Comments |
|---------------------|--|
| General Tab: | |
| Code | Enter a unique number or name for the import |

| Field | Comments |
|------------------------------------|--|
| Description | |
| Description 2 | |
| Date | |
| No. of Objects | The program automatically fills in this field with the number of objects that you have already imported in this version. |
| Import Tab: | |
| Import File Name | If you want to import object data from an object file, enter the name of the object file in this field. |
| Import from Navision Client | Check this option to import the objects directly from a Navision application database. After you start the import, select a running Navision client instance in the Select Navision Client window. After you have selected the Navision client instance, a window appears where you can set object filters. The program automatically transfers all filtered objects from the Navision application database to the Navision Developer's Toolkit database and stores them in the current version. |
| New Version | Check this option, if the import should treat these objects as a completely new version. New version means that this version can be selected in the Object Administrator window and can be used in the compare and merge process. |
| Base Version Code | In this field, you can select an existing version of objects. If the New Version option is checked, the program copies all objects from the selected base version and replaces existing objects with the objects from the import file. If the New Version option is not checked, you can update this base version. You must then select <i>Replace</i> in the Update Option field. |
| Update Option | If you have already imported object data in this company, you can use this option to select the update method. If you select <i>Add</i> and the current object already exists in this company, the existing object data is saved as an older data version and the imported new object is added as the current data version. If you select <i>Replace</i> , an existing object is deleted before the new object data is imported. Your choice in this field is used as default in the Import Worksheet window. |
| Date | This field shows the date of the last import. It is always the system date from the computer where you start the import. |

| Field | Comments |
|--------------------------------|--|
| Duration | The program automatically fills in this field with the duration of the last import in this version. |
| Keywords Tab: | |
| STX Filename | You can select a Navision STX file that belongs to the version of the application database from which you have exported the object file. After you have entered the STX Filename , the program automatically fills in the other fields on this tab. |
| Version List Expression | The country-specific expression for the keyword Version List in the object file. |
| Date Expression | The country-specific expression for the keyword Date in the object file. |
| Time Expression | The country-specific expression for the keyword Time in the object file. |
| Modified Expression | The country-specific expression for the keyword Modified in the object file. |
| Yes Expression | The country-specific expression for the keyword Yes in the object file. |
| No Expression | The country-specific expression for the keyword No in the object file. |
| System Tab: | |
| Date in Import File | This field is filled in automatically when you fill the Import File Name field. You can see the date format of the first object in the import object file. |
| Date Format | Enter the date format for the date field in the import file. You can use your country-specific shortcuts for day, month, and year, but you must make sure that the delimiters are the same as in the field Date in Import File field. |
| Time in Import File | This field is filled in automatically when you fill the Import File Name field. You can see the time format of the first object in the import object file. |
| Time Format | Enter the time format for the date field in the import file. You can use your country-specific shortcuts for hour, minute, and second, but you must make sure that the delimiters are the same as in the field Time in Import File . |
| Filter Tab: | |
| Object Type | You can enter filters in these fields to reduce the number of objects that will be transferred during the import through a Navision client. |
| Object ID | |
| Date | |
| Time | This field are only editable if you have selected the Import from Navision Client field on the Import tab. |
| Version List Modified | |

Updating an Existing Import Version

If you have already created an import version, you can update an existing import version using the existing import version card. If you use the same object file name and path to import the objects, you can use the same import version card. When you do this, make sure that the settings in the **Import Version Card** window correspond to the object file you want to import and the version that you want to update. Otherwise, if you wish to use a new object file or one with a different path, you must create a new import version card.

Note

.....
 If you only want to analyze the current object version of your application database, you should use the *Replace* option in the **Update Option** field. This option ensures that the program always deletes the old data version of an object before importing the new object. This reduces the duration of the import and the space needed in the Navision Developer's Toolkit database.

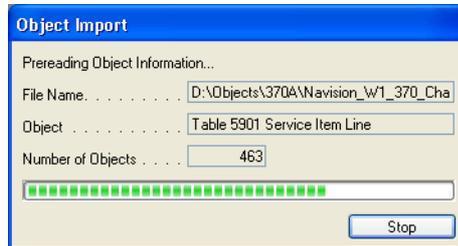
Starting the Object Import

After you have filled in and checked the fields in the **Import Version Card** window, click Import to import the object data from the object file. Depending upon the contents of the file, the program reacts as follows:

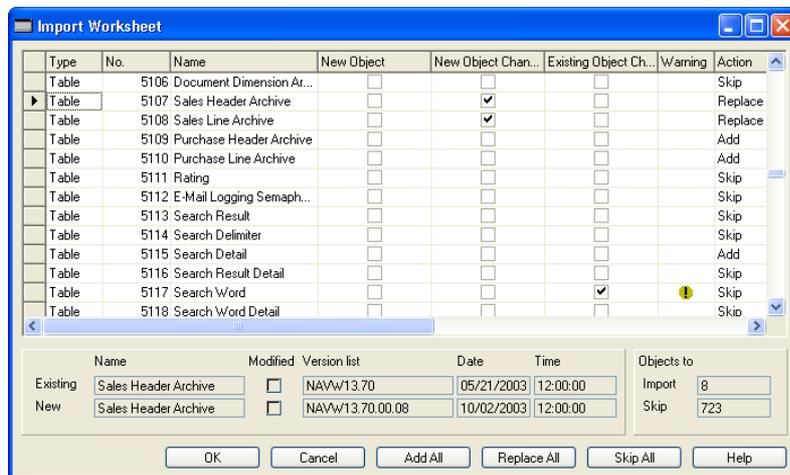
- If the company is empty, the program asks if you want to import the objects now. Click Yes to import the objects. Click Cancel to cancel the import process. Click No to open the **Import Worksheet** window (described below).
- If the company already contains objects, the program asks if you want to continue the import by opening the import worksheet, or whether you want to cancel the import. You can open the **Import Worksheet** window by clicking OK, or you can cancel the import by clicking Cancel.
- If there are conflicts, the program tells you this, and you can choose either to cancel the import or to go to the **Import Worksheet** window by clicking OK from the dialog.

There are two kinds of conflicts that can occur between the versions: The first may occur if the object name, date, time, or version list of the new and the existing objects are not the same, indicating that the new version contains changes. Another type of conflict arises if you have canceled a processing import and restarted this import again. When the program finds a conflict, you see a warning icon in the **Warning** field and the **Action** field is set to *Skip*.

When you first click the Import button in the import version card, the following status window appears:



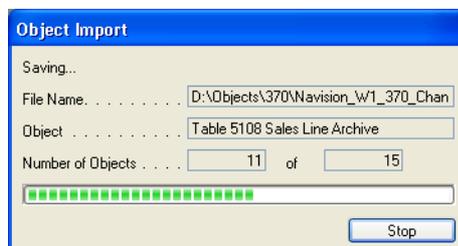
When the program has scanned the object file and you have opened the import worksheet, the **Import Worksheet** window appears:



Each line in the import worksheet displays information about one object and lets you decide how to handle the object by setting an option in the **Action** field.

Click OK to start the import.

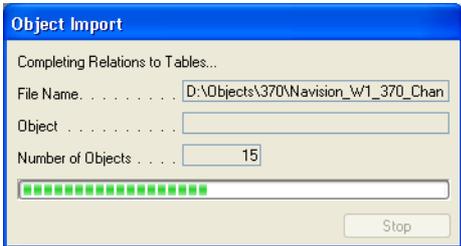
During the object import, the program displays the following status window:



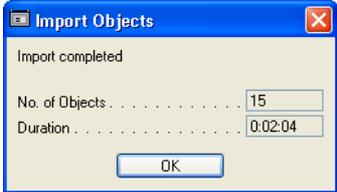
Note

.....
If an import error occurs, the program stops and displays an error message indicating in which object the error has occurred. Using the error message, you can check the areas that have not been imported to find the corresponding line in the object file to correct the error. All objects that are already imported when the error occurs remain in the database.
.....

After the object import has finished, the information about the relations between objects is updated. During this process, the program displays the following status window:



After the update has finished, the objects in the **Object Administrator** window are updated and the following window appears:



Attention

.....
The import process uses much of the processor resources. Other applications might work more slowly during the import.
.....

2.3 SETTING UP THE DEMONSTRATION DATABASE

When you install Navision Developer's Toolkit with example files, the installation program copies a Navision backup file to the directory where you install Navision Developer's Toolkit. This backup file contains a backup of the objects from the Navision demonstration database.

If you want to set up the Navision Developer's Toolkit demonstration database, use the following procedure:

- 1 Start the Navision client. If you are using SQL Server, then be sure the client is a Navision SQL client and connect to the SQL Server.
- 2 Create a new database by clicking the File menu, pointing to Database, and clicking New.

If you are not working with SQL Server, then, in the **New Database** window, create the database in the directory where you installed Navision Developer's Toolkit. If you want to import the object data of a standard Navision application database later on, you should use at least 500 MB as the file size for this new database.

If you are working with SQL Server, then, in the **Select Server** window, select the SQL Server. In the **New Database** window, insert a new database name.

- 3 Restore the Navision backup file `DevTool1.fbk` into this database from the directory where you installed Navision Developer's Toolkit.
- 4 After the restore is finished, close Navision.
- 5 Start Navision Developer's Toolkit. Point to Database on the File menu and click Open to open the previously created Navision Developer's Toolkit database.

The demonstration database includes one company with an import of all objects from Navision application database version W1 4.00.

2.4 UPGRADING OR CONVERTING AN EXISTING DATABASE

If you have used a previous version of Navision Developer's Toolkit, you can upgrade the database that you used with that version so that it can be used with the newest version of Navision Developer's Toolkit.

Since, starting with version 2.0, you can now use SQL Server-based databases, it is also possible to convert your existing Navision Developer's Toolkit database to a database that runs on SQL Server.

Upgrading an Existing Database

In order to use an older database with the newest version of Navision Developer's Toolkit, you must import new Navision Developer's Toolkit objects into the database.

- 1 Start the Navision client, and open the Navision Developer's Toolkit database.
- 2 Open the Object Designer.
- 3 Click the File menu, click Import. In the dialog box that opens, browse to the `DevTool.fob` file in the directory where you installed the newest version of Navision Developer's Toolkit. Select the file and click Open.
- 4 Click Report in the Object Designer. Select report 99003600 **Upgrade to Version 2.00**, click RUN (ALT+R) and follow the instructions. The upgrade process may take a while, depending on your database size.
You must run this report for each company.

The database is now ready to be used with the latest version of Navision Developer's Toolkit.

Converting an Existing Database to an SQL Server Database

Starting with version 2.0 of Navision Developer's Toolkit, you can use Navision Developer's Toolkit with SQL Server. If you use SQL Server, you may want to convert an existing database that is not SQL Server-based to one that you can use with an SQL Server.

Before you begin the process to convert the database, we recommend that you delete all users. You can create the users again in the SQL Server database when you are finished.

- 1 Start the Navision client and open the existing Navision Developer's Toolkit database.
- 2 Open the Object Designer.
- 3 Click the File menu and click Import. In the dialog box that opens, browse to the `DevTool.fob` file in the directory where you installed the newest version of Navision Developer's Toolkit. Select the file and click Open. Your database now has the newest Navision Developer's Toolkit objects.

- 4** Click Report in the Object Designer. Select report 99003600 ***Upgrade to Version 2.00***, click RUN (ALT+R) and follow the instructions. The upgrade process may take a while, depending on the database size.
You must run this report for each company.
- 5** In the Tools menu, click Backup and create a backup file of all objects in your database.
- 6** Close the Navision client.
- 7** Start a Navision SQL client and connect to the SQL Server.
- 8** Create a new database by clicking the File menu and clicking Database, New.
- 9** In the new database, restore the backup of the objects from the old database by clicking the Tools menu and clicking Restore.

Once you have restored the objects to the new SQL Server-based database, the database is ready to use in Navision Developer's Toolkit.

Chapter 3

Source Analyzer

The Navision Developer's Toolkit - Source Analyzer is a graphical cross-reference tool for Navision databases. Based on the imported object data, you can use various windows and functions to view and analyze the structure of an application database.

This chapter contains basic information about the components of the Source Analyzer.

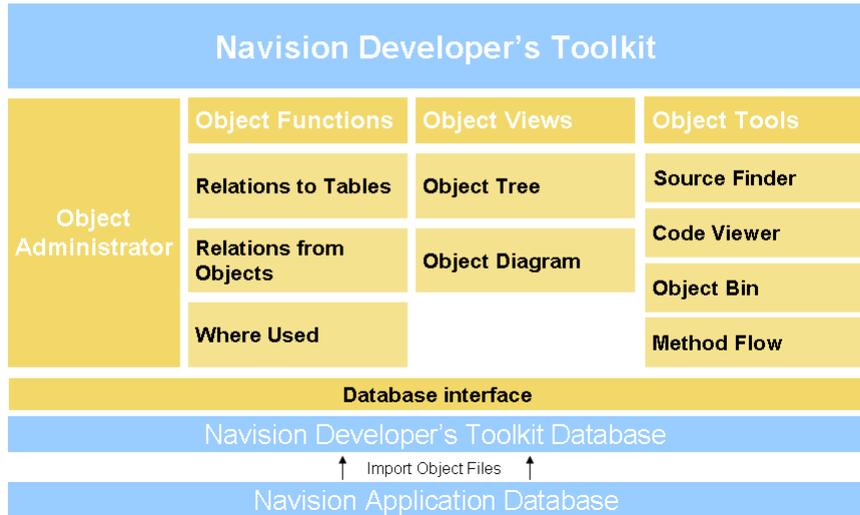
This chapter contains the following sections:

- Overview of Source Analyzer
- Object Administrator
- Object Functions
- Object Views
- Object Tools

3.1 OVERVIEW OF SOURCE ANALYZER

Navision Developer's Toolkit - Source Analyzer is a collection of tools for viewing and analyzing object data from a Navision database. It gives you fast access to the object structure in your application database. The picture below shows the structure and the elements of the Navision Developer's Toolkit - Source Analyzer:

Source Analyzer - Overview



Navision Developer's Toolkit Database

The Source Analyzer examines the object data that you have imported from the application database into the Navision Developer's Toolkit database. During the object import, all information about an object is divided into different areas in the Navision Developer's Toolkit database. These areas are:

- Fields & Controls
- Variables & Parameters
- Functions & Triggers
- Property Values
- Keys
- C/AL Code
- Table Relations

The Source Analyzer offers you different views, tools and functions for analyzing the object data in these areas. Depending on the information you are looking for, you will use the Object Administrator, Object Views, Object Tools or Object Functions to examine the object data.

Object Administrator

The **Object Administrator** window displays the objects you have imported into the Navision Developer's Toolkit database.

Object Functions

You can use the functions below to analyze the structure of the database and the relations between objects:

- Relations to Tables
- Relations from Objects
- Where Used

This chapter describes the main purpose of each window and function. It also shows an example of how to access each window and function.

Object Views

Object Views display the results of the Source Analyzer functions. There are two different windows available:

- Object Tree
- Object Diagram

Object Tools

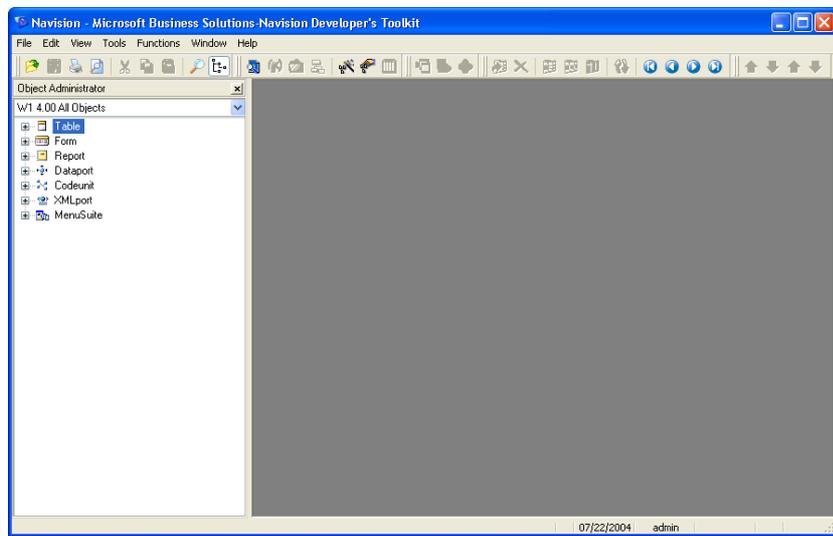
The Object Tools are designed to explore the object data, performing tasks like searching for a text string, viewing C/AL code, or collecting objects. The following Object Tools are available:

- Source Finder
- Code Viewer
- Object Bin
- Method Flow

3.2 OBJECT ADMINISTRATOR

The **Object Administrator** window shows all objects that have been imported to the Navision Developer's Toolkit database. The Object Administrator is the backbone of Source Analyzer. All activities are based on the data of the objects in the Object Administrator. These objects are grouped according to Navision object types (Table, Form, Report, Dataport, Codeunit, XMLport, and MenuSuite).

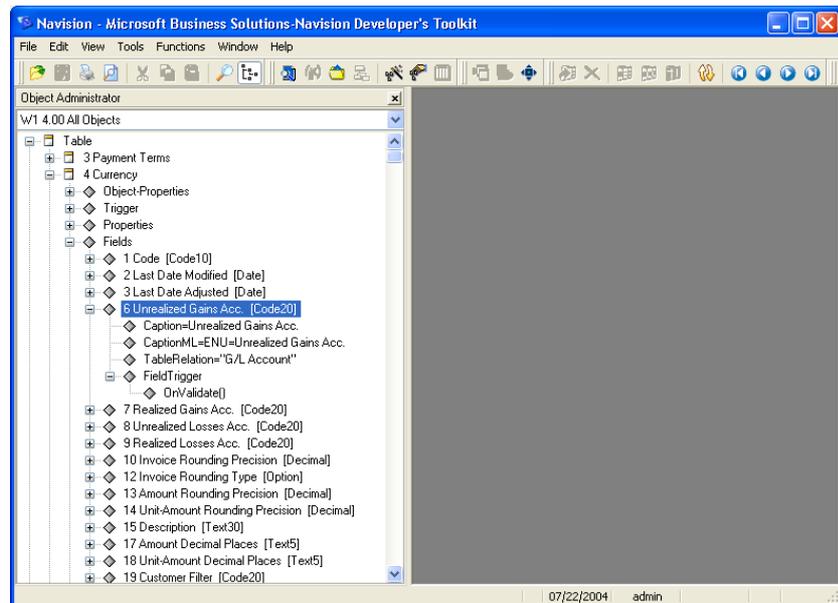
After you have connected to a database, the **Object Administrator** window appears automatically. In the header of the **Object Administrator** window, you can select a version of objects that are already imported. The **Object Administrator** window always appears on the left side of the Navision Developer's Toolkit desktop:



Using the Object Administrator

If you want to see detailed data for an object, you can expand the different levels of the object. For example, if you want to see the imported data for the **Unrealized Gains Acc.** field in the **Currency** table, follow this procedure:

- 1 Click the "+" sign in front of the Table object group to expand this group. Now you can see all tables.
- 2 Click the "+" sign in front of the **Currency** table to expand the first level of this object.
- 3 Click the "+" sign in front of the Fields group. You can see a list of all fields in this table.
- 4 Click the **Unrealized Gains Acc.** field to mark this field.
- 5 In the View menu, click Expand All to expand all levels below this field. The result appears in the **Object Administrator** window:



You can also use the shortcut menu to access some basic functionality for the currently marked line in the **Object Administrator** window.

3.3 OBJECT FUNCTIONS

You can use the Object Functions to see the relations between objects or to find out where an object or a part of an object is used.

Note

.....
If you start one of the Object Functions, the result is displayed in an Object View. The Object Views are described in detail later in this manual.
.....

Relations to Tables

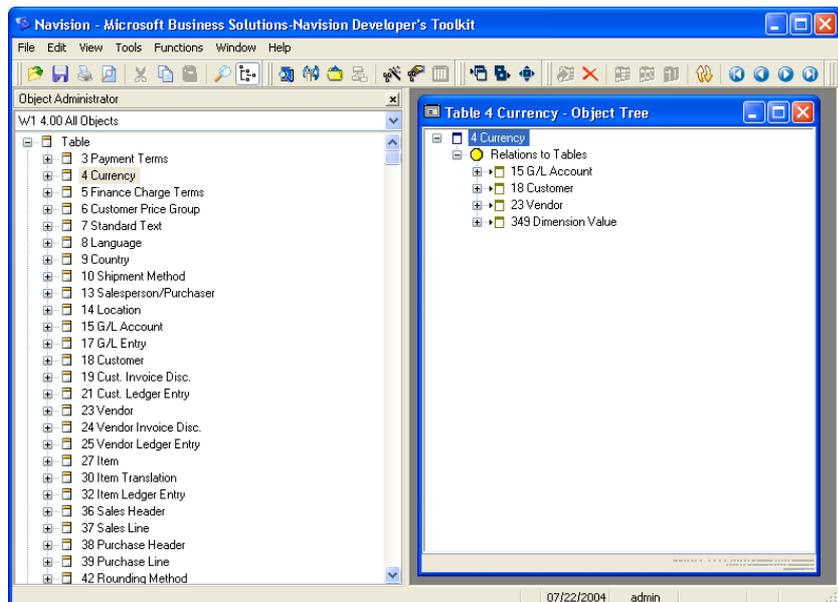
The Relations to Tables function shows you all table relations in the current object pointing to other tables.

In Navision, table relations can be defined in the properties of table fields and data controls, such as textboxes. This means that you can use these functions for tables, forms, reports and dataports.

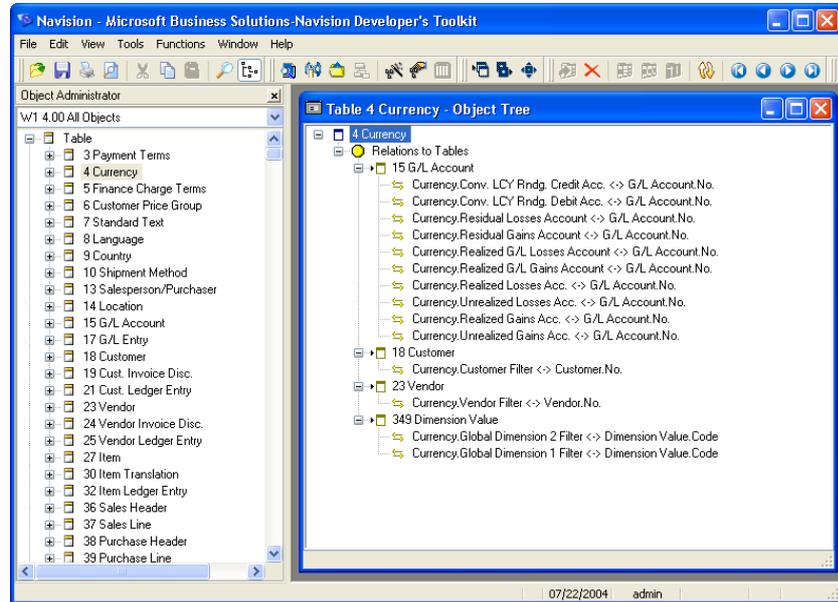
Using Relations to Tables

To see the relations that are defined in the **Currency** table, follow this procedure:

- 1 Click the **Currency** table in the **Object Administrator** window to mark the table.
- 2 On the Functions menu, click Relations to Tables, or click the Relations to Tables command on the shortcut menu. The **Object Tree** window appears showing you the objects defined as table relations in the **Currency** table:



- 3 Click the **Currency** table in the **Object Tree** window to mark this table and click **Expand All** on the View menu to expand all levels.



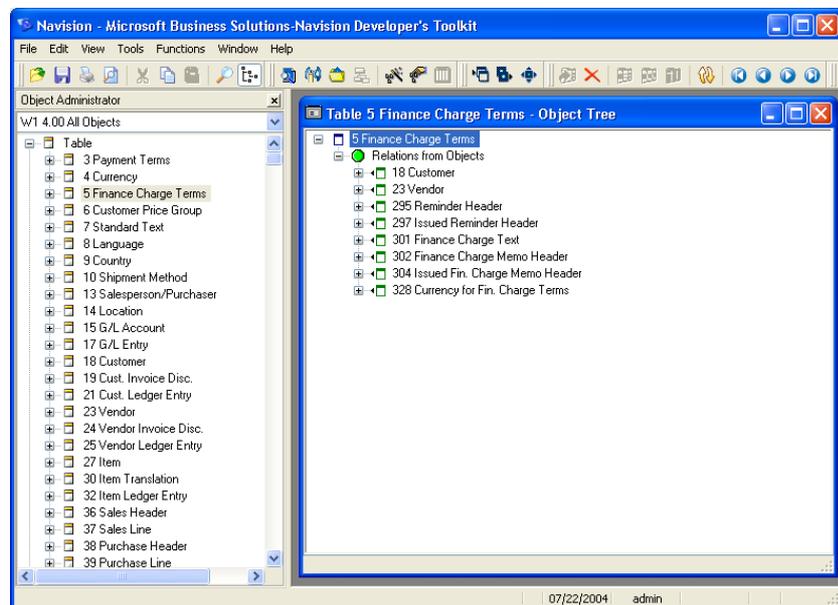
Relations from Objects

The **Relations from Objects** function shows you all relations from other objects that point to the current object. In Navision, you can only define a relation to a table. Consequently, you can only use this function for tables.

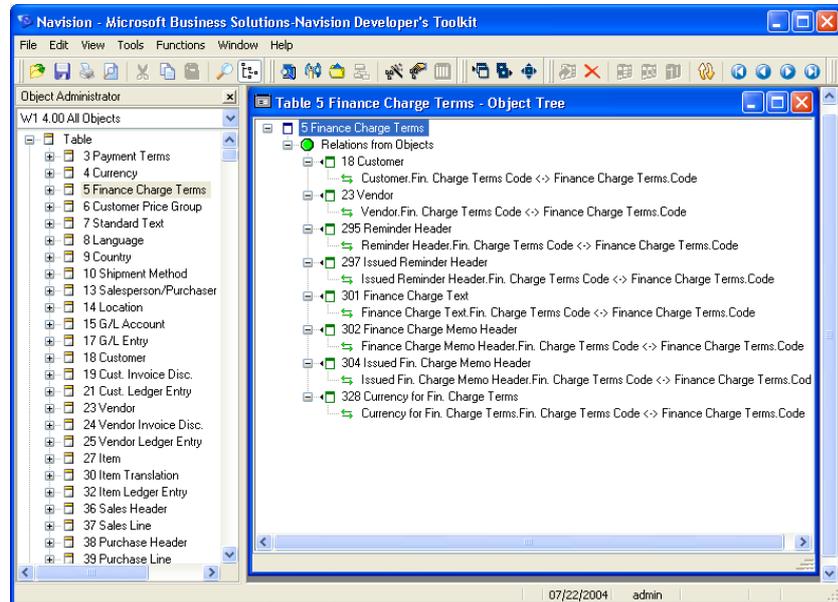
Using Relations from Objects

To see the objects that have table relations that point to the **Finance Charge Terms** table, follow this procedure:

- 1 Click the **Finance Charge Terms** table in the **Object Administrator** window to mark this table.
- 2 On the Functions menu, click Relations from Objects or use the shortcut menu to start the function. The **Object Tree** window appears showing you which objects have table relations that point to the **Finance Charge Terms** table:



- 3 Click the **Finance Charge Terms** table in the **Object Tree** window to mark this table and expand all the lower levels.



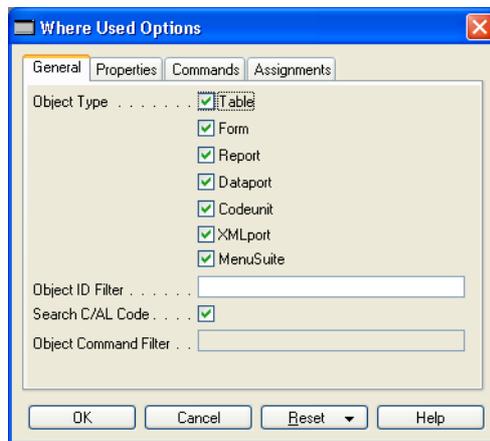
Where Used

The Where Used function shows all the places where an object or a part of an object is used. The Where Used function searches in properties and C/AL code of all object types.

Setting up Where Used Options

You can use the **Where Used Options** window to define object types, object ID's, C/AL code, properties, commands and assignments to be searched by the Where Used function.

To set up the Where Used options, click the Where Used Options command on the Tools menu. The **Where Used Options** window appears:



When you use the Where Used function the first time, the program uses the default setup values. After you have changed the values in the **Where Used Options** window, click OK to save these options. The next time you start the Where Used function, the program will use the new settings. Click the Reset button and click Default to reset all options to their default values.

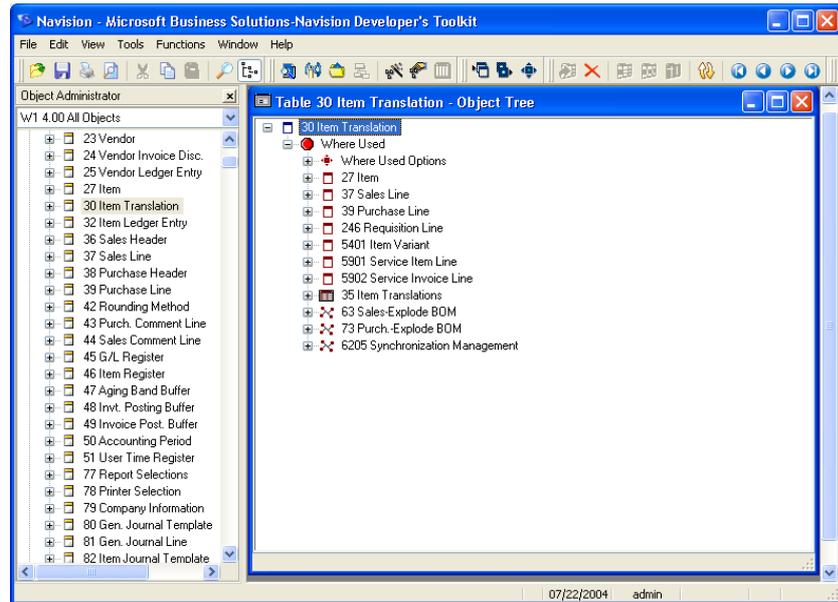
Note

.....
The program saves the changed values in the **Where Used Options** window when you click OK in this window. The program also uses these settings after you restart Navision Developer's Toolkit.
.....

Using Where Used

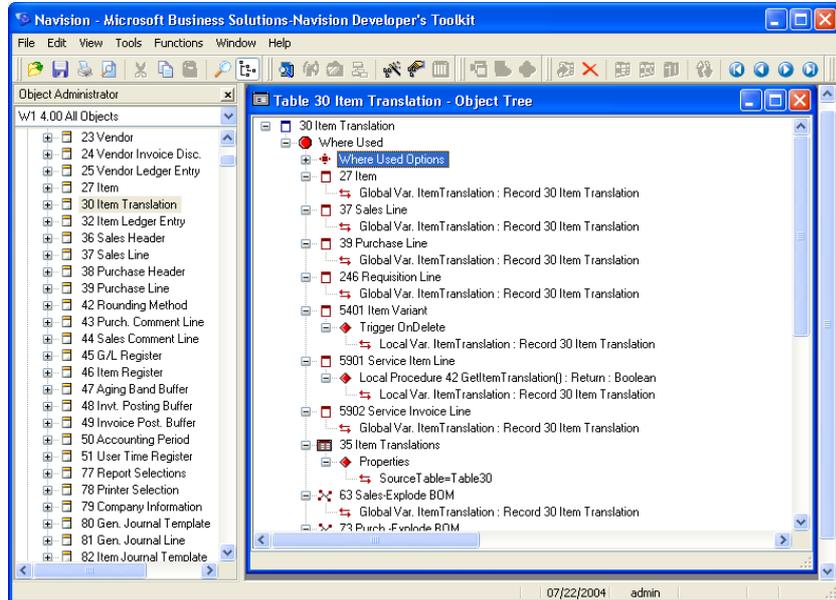
To see where the **Item Translation** table is used, follow this procedure:

- 1 Click the **Item Translation** table in the **Object Administrator** window to mark this table.
- 2 On the Functions menu, click Where Used or click the Where Used command on the shortcut menu to start the function. The **Object Tree** window appears showing you where the **Item Translation** table is used:



- 3 Click the **Item Translation** table in the **Object Tree** window to mark this table. Expand all levels below the table.

- 4 Click the Where Used Options entry in the **Object Tree** window to mark this entry and click Collapse All in the View menu, or click the Collapse All command on the shortcut menu, to collapse all levels.



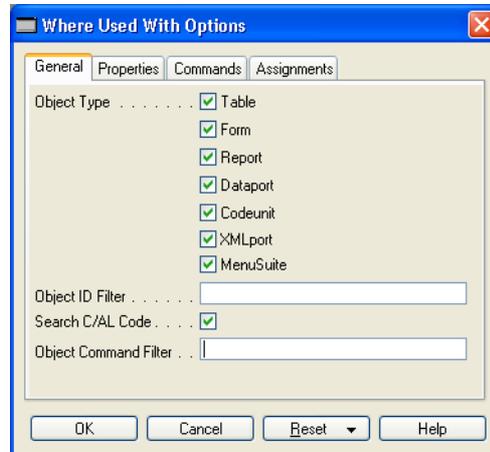
Where Used With

The Where Used With function is based on the Where Used function. When you start the Where Used With function, the **Where Used With Options** window appears automatically. You can change these options specifically for the current search.

Setting up Where Used With Options

You can use the **Where Used With Options** window to define object types, object ID's, C/AL code, properties, commands and assignments that the Where Used function should search. If you start the Where Used With function on an object, you can also set an object command filter.

The **Where Used With Options** window appears automatically when you start the Where Used With function:



After you fill the options and click OK, the program runs the Where Used function. However, the program does not save these options in the general Where Used options.

If you want to change the general Where Used options so that they apply every time you run the Where Used function, click the Where Used Options command on the Tools menu. These general Where Used options are used as presettings for the Where Used With function.

Note

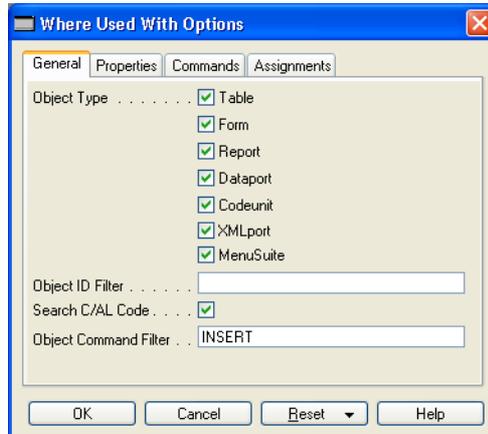
.....
 The changes in the **Where Used With Options** window do not change the general Where Used options.

Using Where Used With

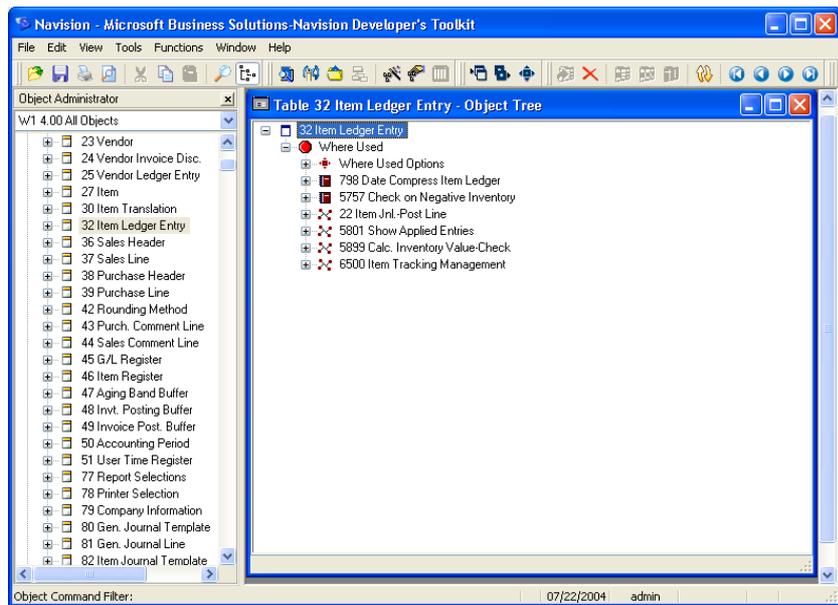
To see where the **Item Ledger Entry** table is inserted, follow the procedure below:

- 1 Click the **Item Ledger Entry** table in the **Object Administrator** window to mark this table.
- 2 On the Functions menu, click Where Used With, or click the Where Used With command on the shortcut menu, to start the function. The **Where Used With Options** window appears.

3 Enter *INSERT* in the **Object Command Filter** field:

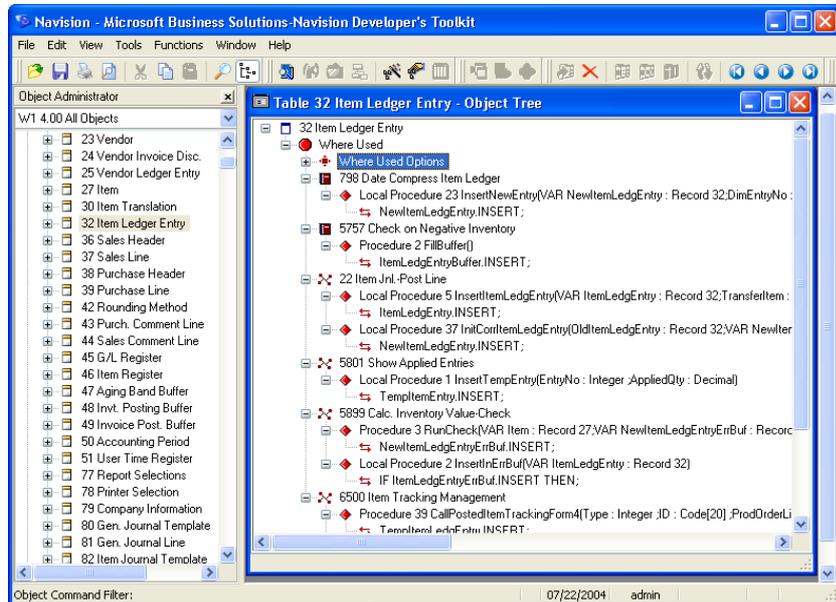


4 Click OK to start the **Where Used With** function. The **Object Tree** window appears showing you where the **Item Ledger Entry** table is inserted:



5 Click the **Item Ledger Entry** table in the **Object Tree** window to mark this table. Expand all levels.

- Click the Where Used Options entry in the **Object Tree** window and collapse all levels below.



3.4 OBJECT VIEWS

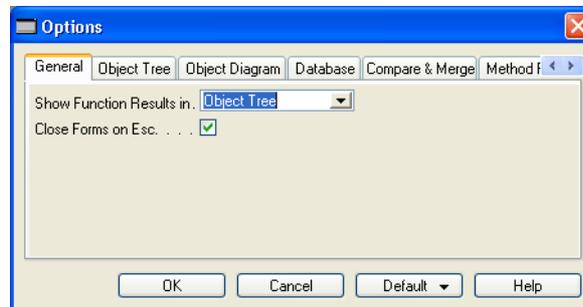
Usually, Object Views show the results of the Source Analyzer functions. There are two different views:

- Object Tree window
- Object Diagram window

Setting up the Default Object View

To select either an **Object Tree** window or an **Object Diagram** window for all new windows that show the results of functions, do the following:

- 1 Click the Tools menu and click Options. The **Options** window appears:



- 2 Click the **Show Function Results in** field, click the AssistButton ▾, and select the Object View you want to use.

Object Tree Window

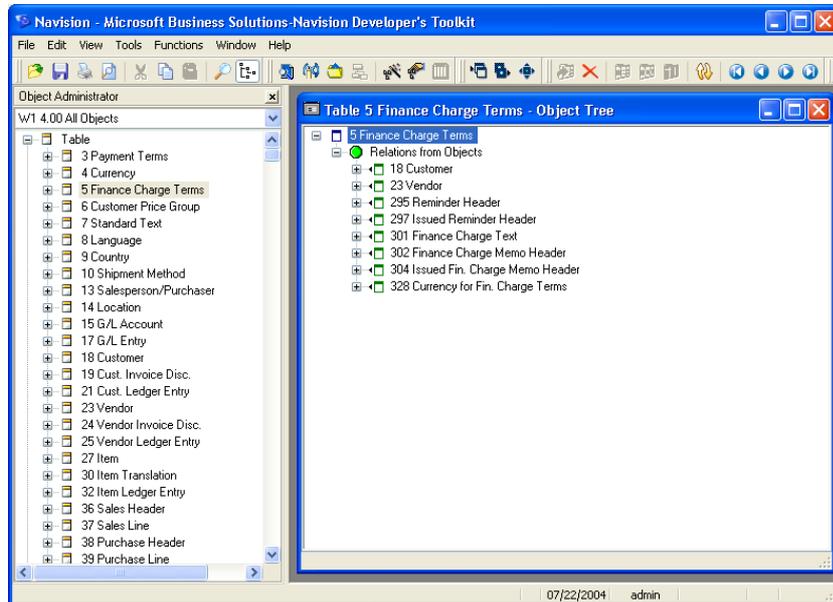
An **Object Tree** window shows the objects in a tree structure similar to the tree structure used in the Windows Explorer. You can also activate functions for an object in the **Object Tree** window. The result of these function operations are shown in the same **Object Tree** window.

Using the Object Tree Window

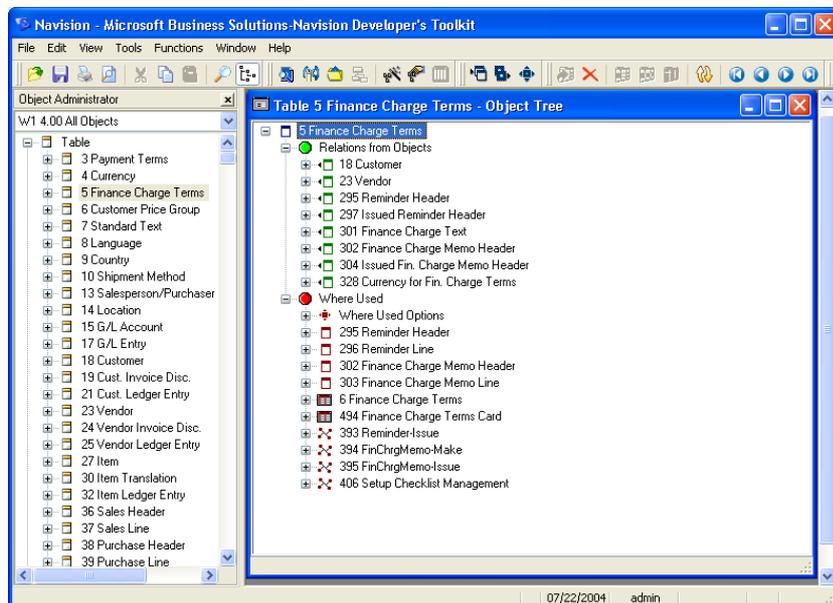
If you want to see the result of the Relations from Objects and Where Used functions for the **Finance Charge Terms** table, follow this procedure:

- 1 Click the **Finance Charge Terms** table in the **Object Administrator** window to mark this table.

- Click the Relations From Objects command in the Functions menu or start the function from the shortcut menu. The **Object Tree** window appears.



- Click the **Finance Charge Terms** table in the **Object Tree** window and click the Where Used command in the Functions menu. The **Object Tree** window appears with the result of both functions:



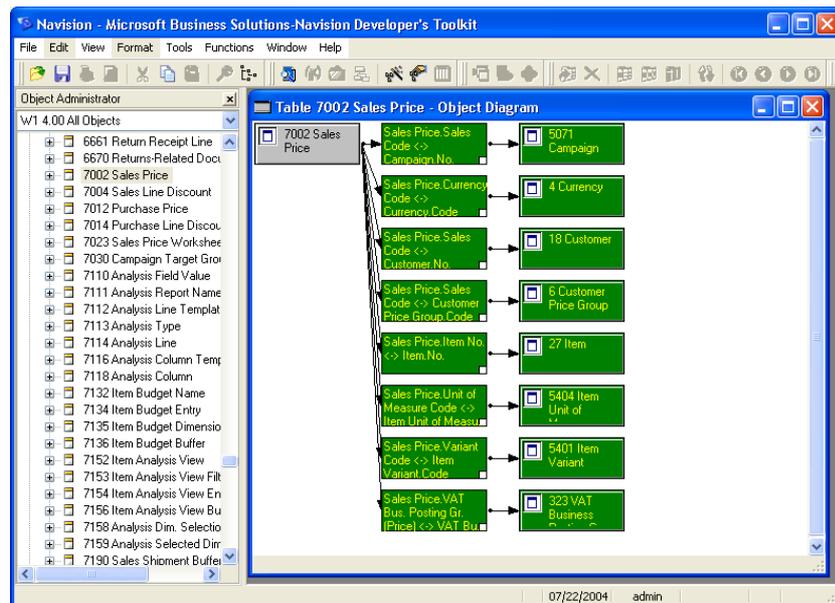
Object Diagram Window

An **Object Diagram** window shows objects in a graphical view. You can also activate any of the functions for an object in an **Object Diagram** window. The results are shown in the same **Object Diagram** window.

Using the Object Diagram Window

If you want to see the result of the Relations to Tables function based on the **Sales Price** table in an **Object Diagram** window, follow this procedure:

- 1 On the Tools menu, click Options change the option in the **Show Function Results in** field to **Object Diagram**. Click OK. You can also click the Treeview On/Off button in the toolbar.
- 2 Click on the **Sales Price** table in the **Object Administrator** window to mark this table.
- 3 On the Functions menu, click Relations to Tables. The **Object Diagram** window appears:



Note

The **Object Diagram** window shows relations graphically. Later versions of Navision Developer's Toolkit will include additional application areas. When the **Object Diagram** window is used with these application areas, it will be possible to take full advantage of its functionality and also view business flows.

3.5 OBJECT TOOLS

The Object Tools give you additional access to object data. Depending on the current object, you can use these tools in different ways. The following tools are available:

- Object Bin
- Code Viewer
- Method Flow
- Source Finder

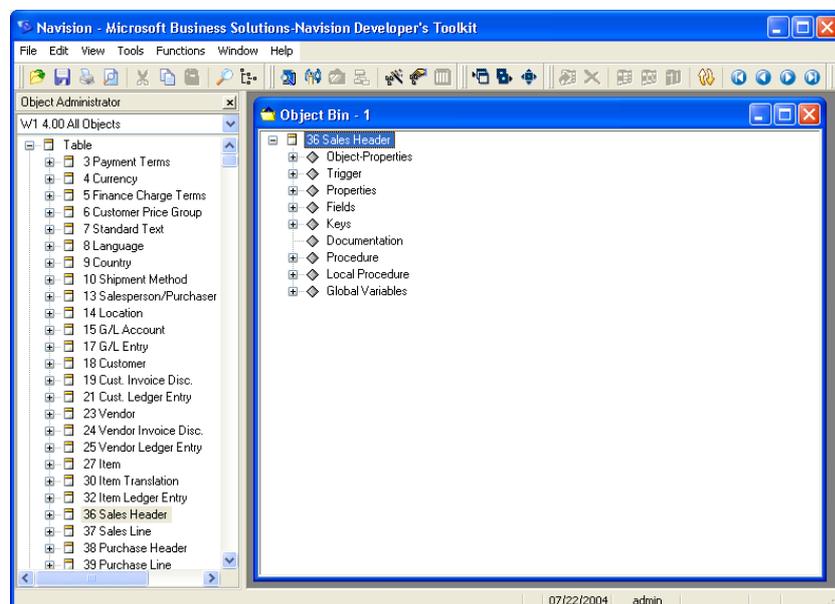
Object Bin Window

The **Object Bin** window shows an object in the same way as the **Object Administrator** window. You can use the **Object Bin** window to collect different objects in one window.

Using the Object Bin Window

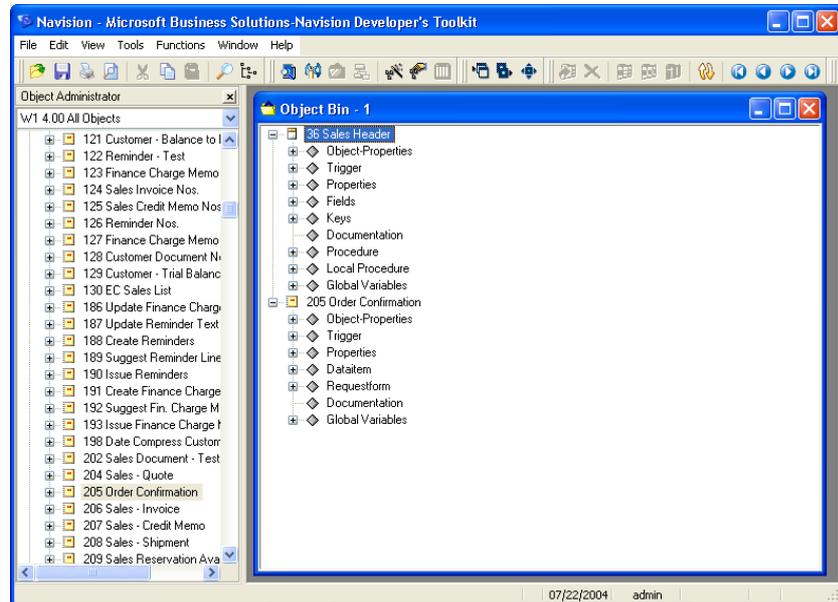
If you want to collect the **Sales Header** table and the **Order Confirmation** report in one **Object Bin** window, follow this procedure:

- 1 Click on the **Sales Header** table in the **Object Administrator** window to mark this table.
- 2 On the Tools menu, click Object Bin to copy this object in a new **Object Bin** window. The **Object Bin** window appears:



- 3 Click report 205, **Order Confirmation** in the **Object Administrator** window to mark this report. Hold down the left mouse button and drag & drop this report to the **Object Bin** window.

- 4 Click the **Order Confirmation** report in the **Object Bin** window and click the "+" sign in front of this report to expand the first level below:



Code Viewer Window

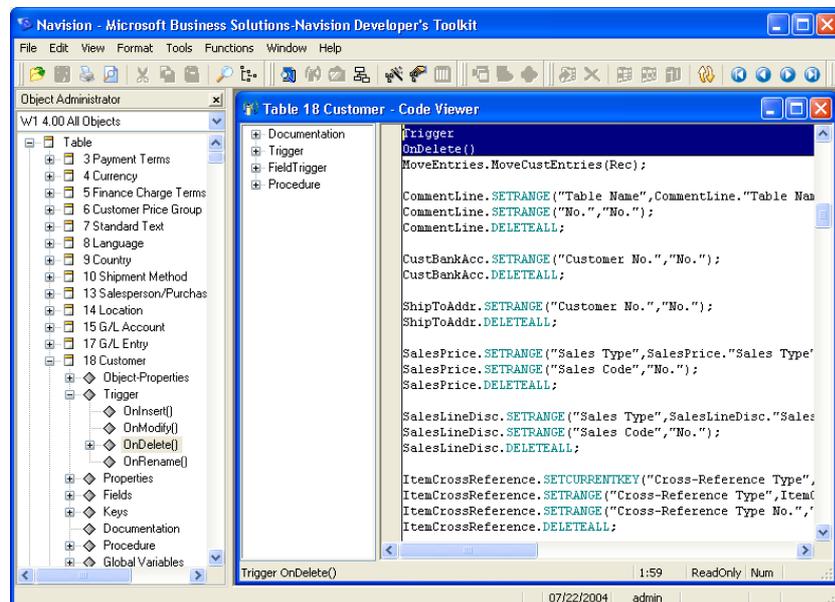
The **Code Viewer** window displays the C/AL code lines for an object. The **Code Viewer** window is divided into two parts. The left side shows the code structure of the object. For example, you can see the list of object triggers, field triggers and procedures. On the right side, you can see the C/AL code lines corresponding to the marked element on the left side. Keywords such as commands are colored to improve the readability of the code lines.

When you start the **Code Viewer** window from an object or field trigger in the **Object Administrator** window, the **Code Viewer** window automatically displays the corresponding trigger.

Using the Code Viewer Window

If you want to see code in the OnDelete trigger of the **Customer** table, follow this procedure:

- 1 Click the "+" sign in front of the **Customer** table entry to expand the first level of this object.
- 2 Click the "+" sign in front of the Trigger entry.
- 3 Click the OnDelete trigger.
- 4 On the Tools menu, click Code Viewer. The **Code Viewer** window appears:

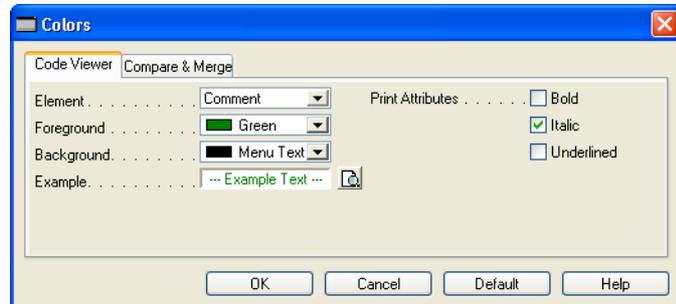


You can expand the levels on the left side of the **Code Viewer** window and click, for example, on the AssistEdit procedure to see the C/AL code in this procedure.

You can also start the Where Used function in the left side of the **Code Viewer** window. In the right side, you can mark code lines and open the **Object Bin** window to show all objects used in these code lines in an **Object Bin** window.

Setting Up Code Viewer Color Options

If you want to change the coloring in the **Code Viewer** window, click the Colors command on the Tools menu. The **Colors** window appears as below:



The color settings are always shown for the currently selected element. To see the coloring of all elements, click the Color Preview button next to the **Example** field.

Method Flow Window

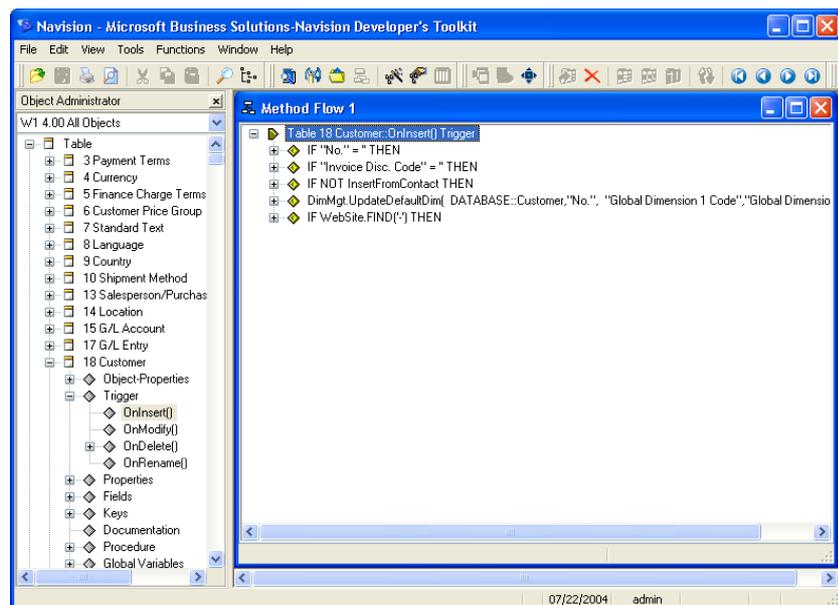
The **Method Flow** window shows C/AL code lines grouped into blocks by keywords for a trigger or a procedure. You can expand or collapse these blocks.

You can also expand a code line that contains a trigger or a procedure call to the same object or to another object. The **Method Flow** window shows a new node that shows the code lines of this trigger or procedure as indented lines in the same window.

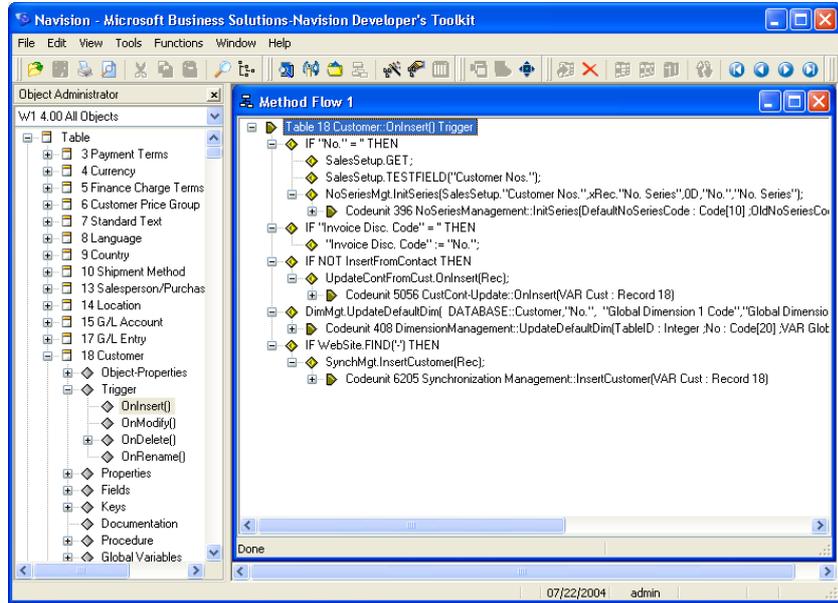
Using the Method Flow Window

To explore the OnInsert trigger in the **Customer** table, follow this procedure:

- 1 Click the "+" sign in front of the **Customer** table entry to expand the first level of this object in the **Object Administrator** window.
- 2 Click the "+" sign in front of the Trigger entry.
- 3 Click the OnInsert trigger.
- 4 On the Tools menu, click Method Flow. The **Method Flow** window appears:



- 5 Click the first line in the **Method Flow** window to mark this entry. Expand all lower levels.



If you click one of the nodes with the name Codeunit, you can expand this node to see the code lines from this codeunit.

Note

.....
The default settings specify that comments and BEGIN/END lines are not shown in the **Method Flow** window. On the Tools menu, click Options to change these settings.
.....

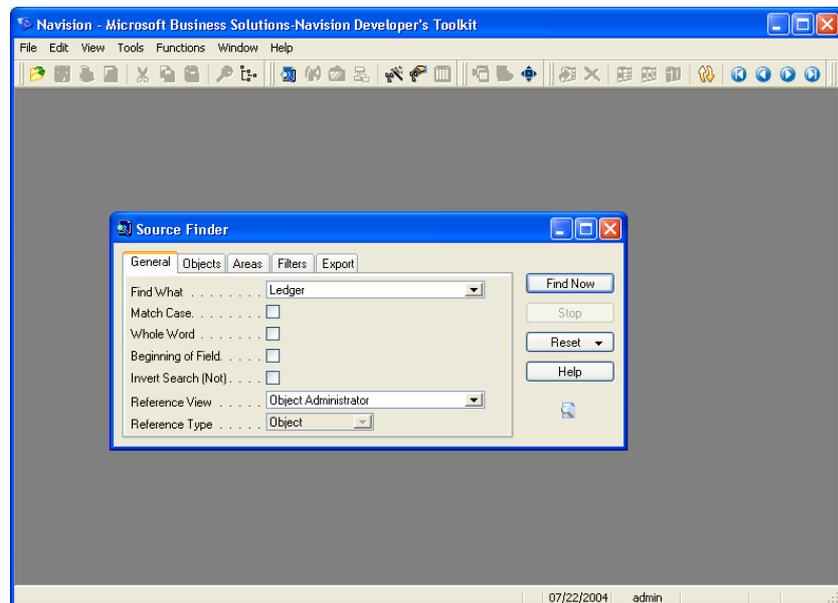
Source Finder Window

The Source Finder searches the database for a specific character string. You can use the settings on the different tabs in the **Source Finder** window to specify where in the database the Source Finder must search.

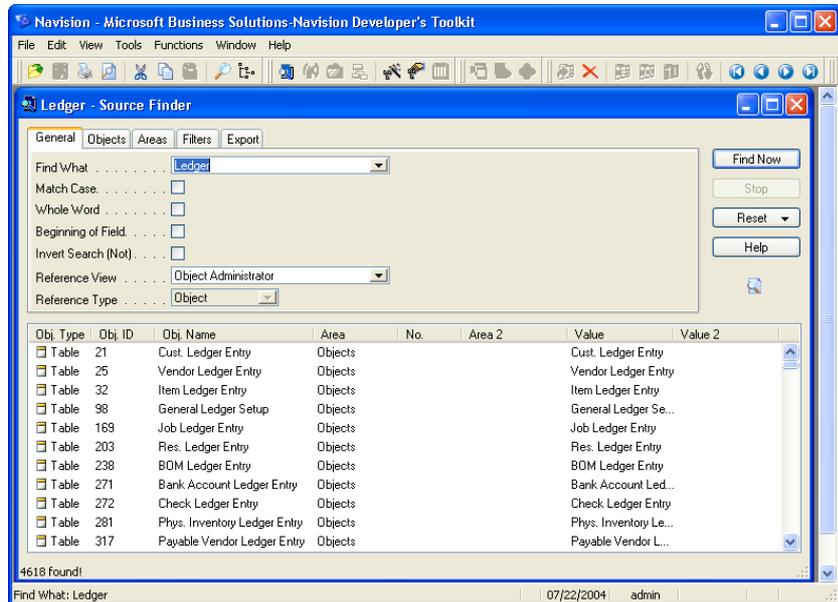
Using the Source Finder Window

To search for the string *Ledger* in all objects, follow this procedure:

- 1 On the View menu, click Object Administrator, or press F12 to hide the **Object Administrator** window. This gives you more space on your Navision Developer's Toolkit desktop.
- 2 Click the Tools menu and click the Source Finder command. The **Source Finder** window appears.
- 3 Enter *Ledger* in the **Find What** field.



- 4 Click Find Now in the **Source Finder** window to start the search. The **Source Finder** window shows the results of the search:



This **Source Finder** window shows you the occurrences of the expression *Ledger* in all objects and all areas. You can expand the width of the columns to see the complete entries.

Note

.....
 You can limit the search by changing the values on the **Objects**, **Areas** or **Filters** tabs.

Chapter 4

Examples of How to Use Source Analyzer

This chapter describes some typical cases from your daily work in using the Source Analyzer.

This chapter contains the following sections:

- Introduction
- Viewing Object Data
- Analyzing Objects
- Searching for Object Data

4.1 INTRODUCTION

The examples in this chapter are based on standard Navision version 4.00 objects. These objects are already imported in the Navision Developer's Toolkit demonstration database. Before you can do the exercises, you must start Navision Developer's Toolkit and open the Navision Developer's Toolkit demonstration database (for example, `DevTool.fdb`).

4.2 VIEWING OBJECT DATA

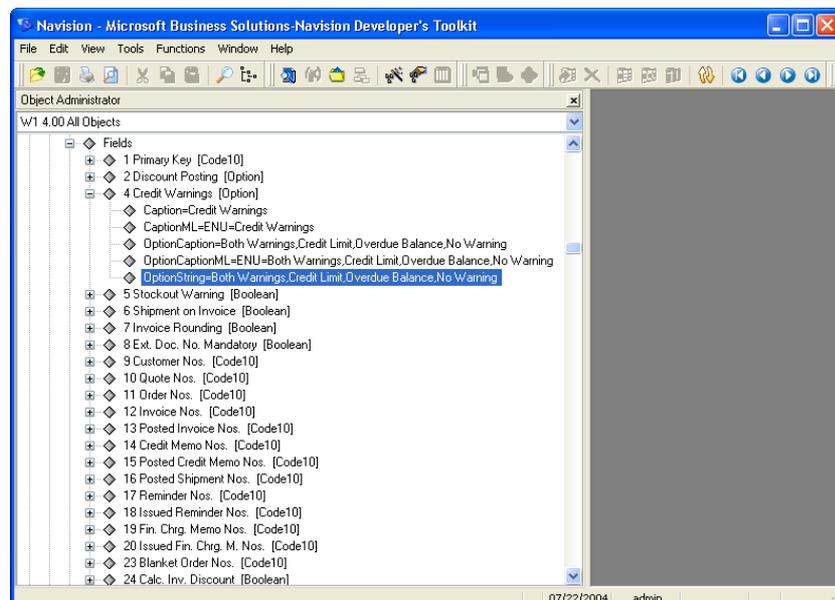
During the development of a Navision application, you always need information about other objects. You can open these objects in the Object Designer of Navision, but then you lose focus and hide your original designer window. If you use Navision Developer's Toolkit to look at the object data, you only have to switch the application and the focus of the designer window in Navision will not be changed.

This section explains how to view different types of object data. For the examples below, we recommend that the **Object Administrator** window is always open.

Finding the Option String of a Field

This example shows how to find the option string of the field 4, **Credit Warnings** in table 311, **Sales & Receivables Setup**:

- 1 Click the "+" sign in front of the Table object group in the **Object Administrator** window to expand this group.
- 2 Use the Find command on the Edit menu to find the expression **Sales & Rec.**
- 3 Click the "+" sign in front of the **Sales & Receivables Setup** table to expand the first level of this object.
- 4 Click the "+" sign in front of Fields. You see a list of all fields in this table.
- 5 Click the "+" sign in front of the field 4, **Credit Warnings** to expand the level below this field. Now you can see the OptionString property, as shown in the following:

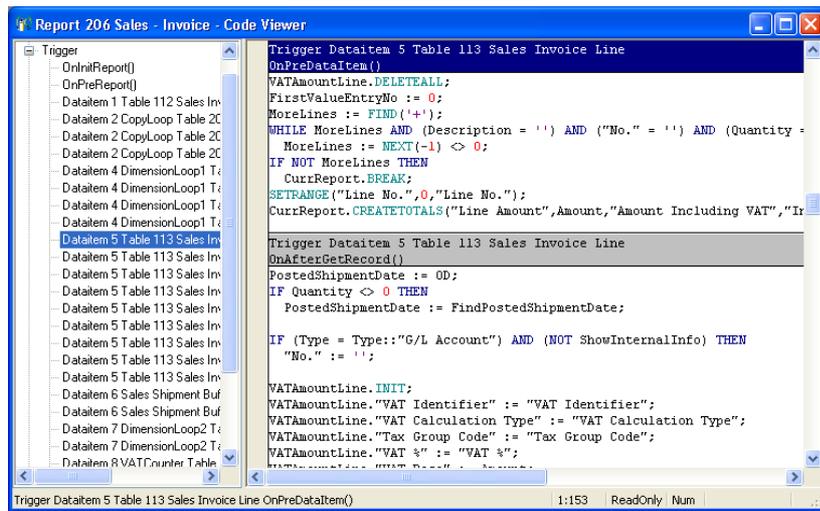


Viewing all C/AL code lines of a report

Sometimes you need to see the C/AL code lines in different triggers of an object. For example, in reports, you may have to select either a data item or a section to view the code lines.

To see all C/AL code lines of data item 5 in report 206, **Invoice** follow this procedure:

- 1 Expand the Report object group in the **Object Administrator** window.
- 2 Click report 206, **Invoice**.
- 3 On the Tools menu, click Code Viewer to open the **Code Viewer** window.
- 4 Expand all triggers on the left side of the **Code Viewer** window.
- 5 Click the first line that contains Dataitem 5. The program points automatically to the first trigger of the data item on the right side of the **Code Viewer** window:



The right part of the **Code Viewer** window contains all triggers of the current object. In the status bar of the **Code Viewer** window, you can always see the complete name of the current trigger. You can use the clipboard to copy code lines into the Navision C/AL editor.

4.3 ANALYZING OBJECTS

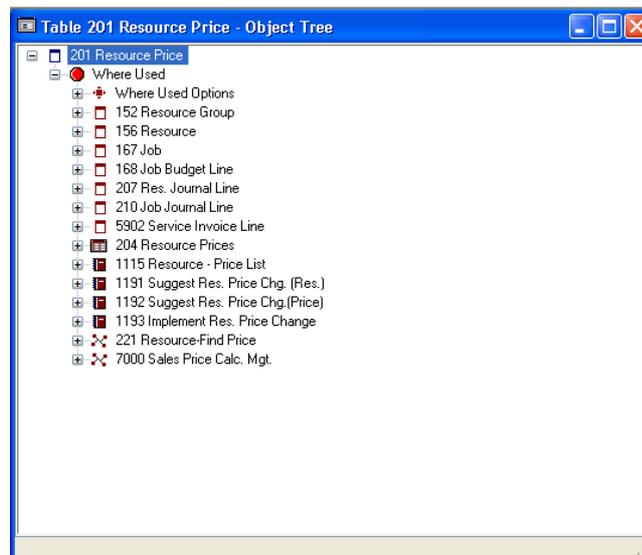
Whether you are designing a new application or supporting or updating an existing application, you often need to analyze the relations between objects or the usage of whole objects or parts of them.

This section shows you some examples of analyzing objects in the Source Analyzer.

Analyzing the Usage of a Table

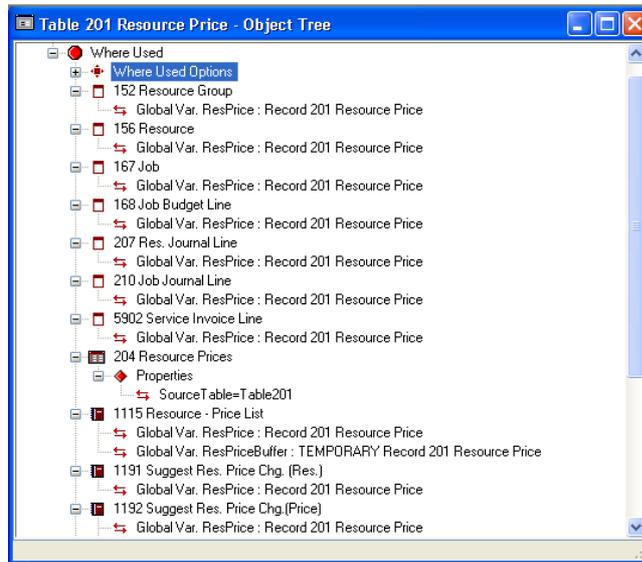
This example shows how to examine the usage of table 201, **Resource Price**:

- 1 Expand the Table object group in the **Object Administrator** window.
- 2 Click table 201, **Resource Price**.
- 3 On the Functions menu, click Where Used. The **Object Tree** window appears showing you where the **Resource Price** table is used:



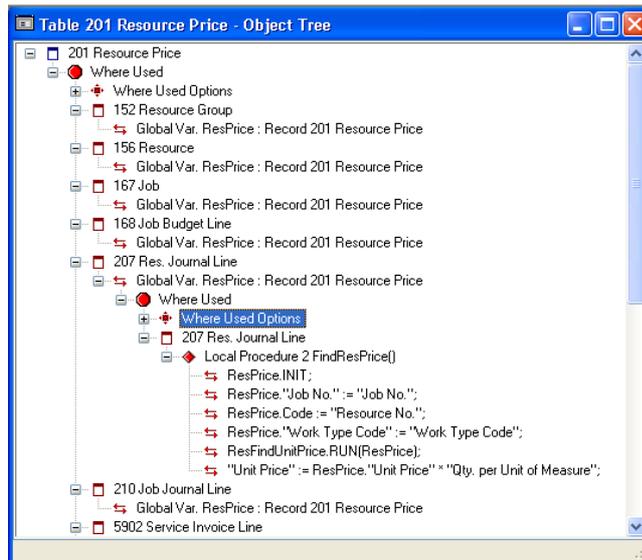
- 4 Click Where Used in the **Object Tree** window and expand all lines. Now you can see all levels in the **Object Tree** window.

- Click the Where Used Options entry in the **Object Tree** window and collapse all levels.



You can see that table 201, **Resource Price** is used in table 207, **Res. Journal Line** as a global record variable.

- Click the line below table 207, **Res. Journal Line** in the **Object Tree** window.
- On the Functions menu, click Where Used.
- Expand all levels below the currently marked line.
- Click on Where Used Options in the **Object Tree** window and collapse all levels.



You can see all places where table 201, **Resource Price** is used in table 207, **Res. Journal Line** as a global variable called **ResPrice**.

10 Click the line with the code `ResFindUnitPrice.RUN(ResPrice);`

11 On the Tools menu, click Code Viewer to see the corresponding code line in table 207, **Res. Journal Line**. The **Code Viewer** window appears:

```

ResFindUnitPrice.RUN(ResPrice);
"Unit Price" := ResPrice."Unit Price" * "Qty. per Unit of Measure";
VALIDATE("Unit Price");

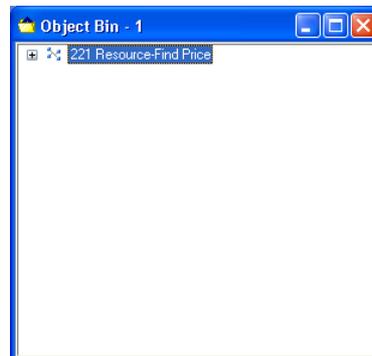
EmptyLine() : Return : Boolean
Procedure 5
EXIT(("Resource No." = '' AND {Quantity = 0});

SetUpNewLine(LastResJnlLine : Record 207)
Procedure 8
ResJnlTemplate.GET("Journal Template Name");
ResJnlBatch.GET("Journal Template Name","Journal Batch Name");
ResJnlLine.SETRANGE("Journal Template Name","Journal Template Name");
ResJnlLine.SETRANGE("Journal Batch Name","Journal Batch Name");
IF ResJnlLine.FIND('-') THEN BEGIN
    "Posting Date" := LastResJnlLine."Posting Date";
    "Document Date" := LastResJnlLine."Posting Date";
    "Document No." := LastResJnlLine."Document No.";
END ELSE BEGIN
    "Posting Date" := WORKDATE;
    "Document Date" := WORKDATE;
    IF ResJnlBatch."No. Series" <> '' THEN BEGIN
        CLEAR(NoSeriesMgt);
        "Document No." := NoSeriesMgt.TryGetNextNo(ResJnlBatch."No. Series","Posti
END;
  
```

The cursor is now in front of the corresponding line.

12 Double click the phrase `ResFindUnitPrice` in the current line to mark this expression.

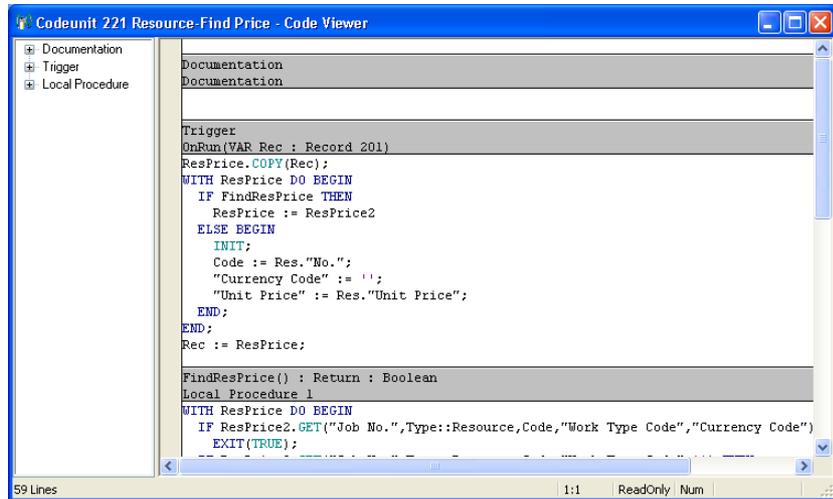
13 On the Tools menu, click Object Bin to copy this object in a new **Object Bin** window. The **Object Bin** window appears:



You can expand the different levels of this object to see all properties.

14 Click the first line in the **Object Bin** window.

15 On the Tools menu, click Code Viewer to see the code lines in this codeunit. The **Code Viewer** window appears:



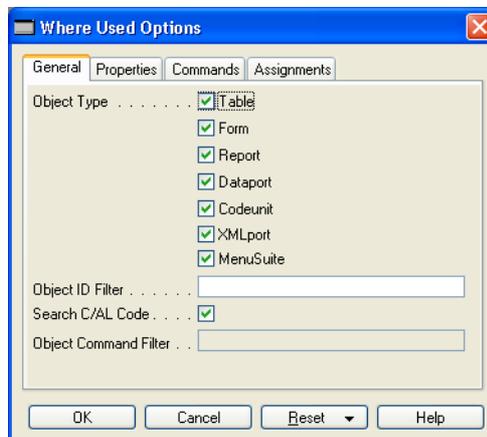
You can see all code lines in codeunit 221, **Resource-Find Price** to analyze what happens in this codeunit.

Finding Where a Field is Used

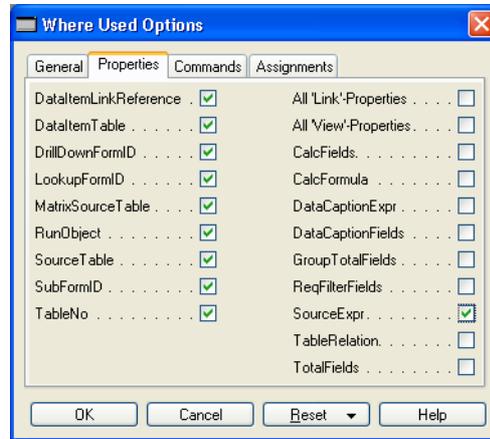
Suppose that you need to adjust the length of a field in Navision. Before you can change the field, you have to analyze the usage of this field. First of all, you need to know where the field is filled in or transferred to another field or used as a source expression of controls.

The example below shows how to analyze where the **Description** field in table 27, **Item** is used in C/AL code lines and as source expression of controls:

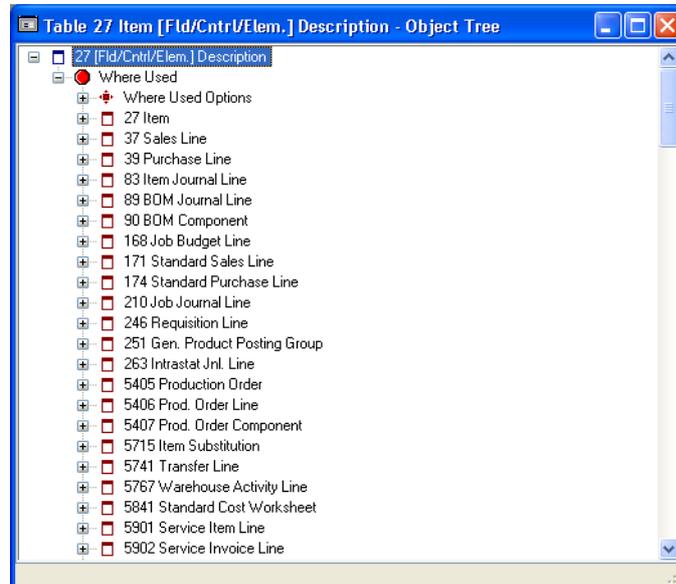
1 On the Tools menu, click Where Used Option. The **Where Used Options** window appears:



- Click the **Properties** tab and insert a ✓ in the **SourceExpr** field, as you see it in the **Where Used Options** window below:



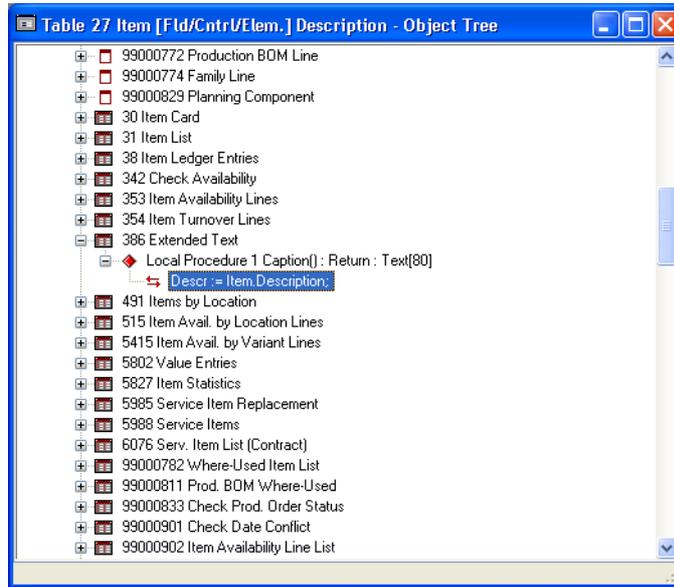
- Click OK.
- Click field 3, **Description** in table 27, **Item** in the **Object Administrator** window.
- On the Functions menu, click Where Used. The **Object Tree** window appears showing you where the **Item** table is used:



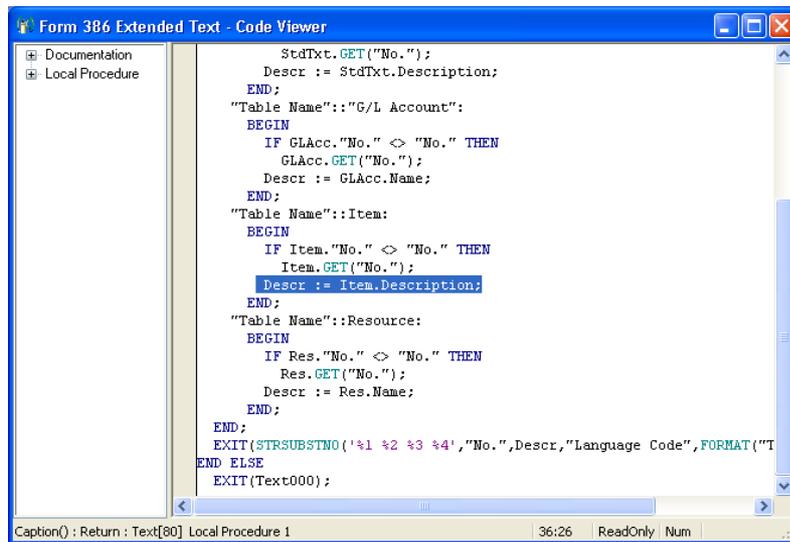
The **Object Tree** window shows all objects where the **Description** field is used.

- Click form 386, **Extended Text** and expand the levels below this form.

7 Click the indented line below form 386, **Extended Text**.



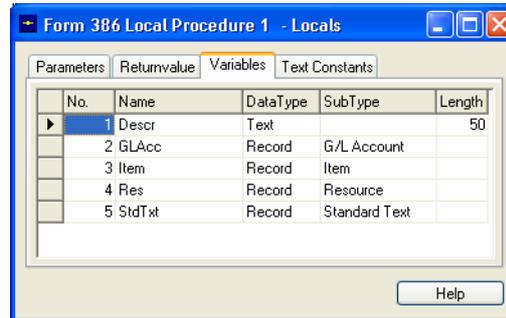
8 On the Tools menu, click Code Viewer to view all lines in this local procedure. The **Code Viewer** window appears:



The cursor is in front of the code line where the **Description** field is used. The value of the **Description** field is transferred to the **Descr.** variable.

9 Click the View menu and click C/AL Locals.

10 Click the **Variables** tab:



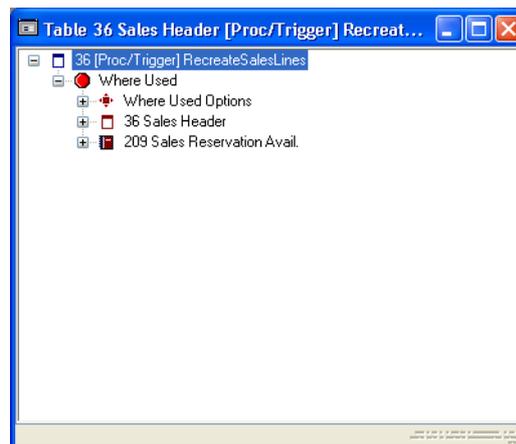
You can see the definition of the **Descr.** variable in the **Locals** window, which you also might have to change according to the changes of the **Description** field in table 27, **Item**.

Finding Where a Procedure is Called

During the support of an application, you might have to analyze, for example, where two procedures are called, to inform a user where functionality is processed.

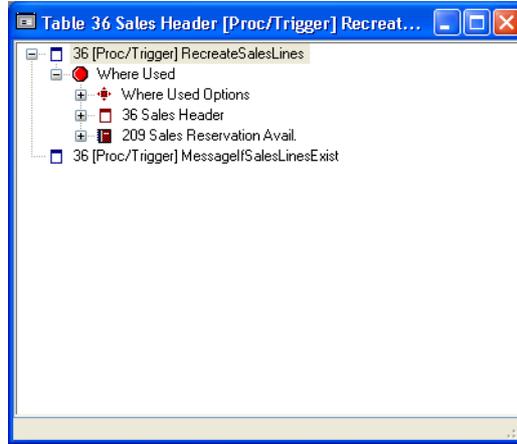
This example shows you how to find out where procedure 4, **RecreateSalesLines** and procedure 5, **MessageIfSalesLineExists** in table 36, **Sales Header** are called:

- 1 Click procedure 4, **RecreateSalesLines** in table 36, **Sales Header** in the **Object Administrator** window.
- 2 On the Functions menu, click Where Used. The **Object Tree** window appears:



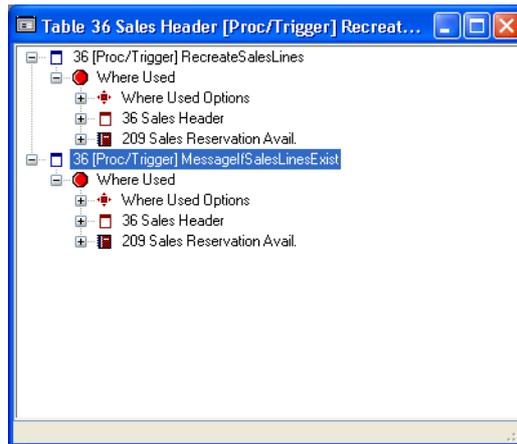
- 3 Click procedure 5, **MessageIfSalesLineExist** in table 36, **Sales Header** in the **Object Administrator** window.

4 Drag this procedure and drop it in the existing **Object Tree** window:

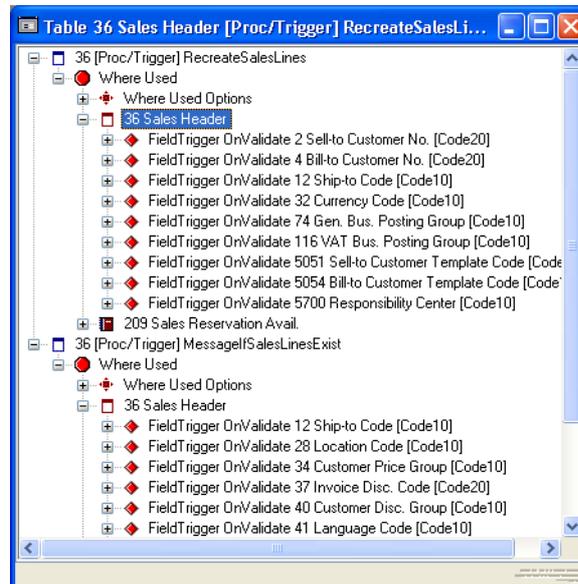


5 Click procedure **MessagelfSalesLineExist** in the **Object Tree** window.

6 On the Functions menu, click Where Used. The **Object Tree** window is updated:



7 Click the two lines that contain table 36, **Sales Header** and expand all lines.



You can see all OnValidate triggers where these procedures are called.

4.4 SEARCHING FOR OBJECT DATA

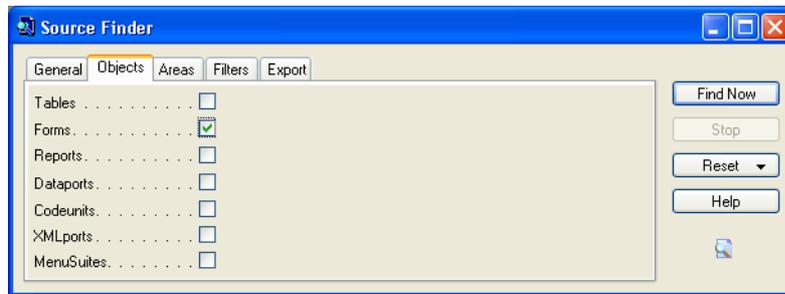
Searching for object data means searching for a text expression in different data areas of an object. The Source Finder provides you in your daily work with a powerful search engine to find various expressions in different objects, such as properties, commands, names and types.

This section shows you some examples of how to use the Source Finder.

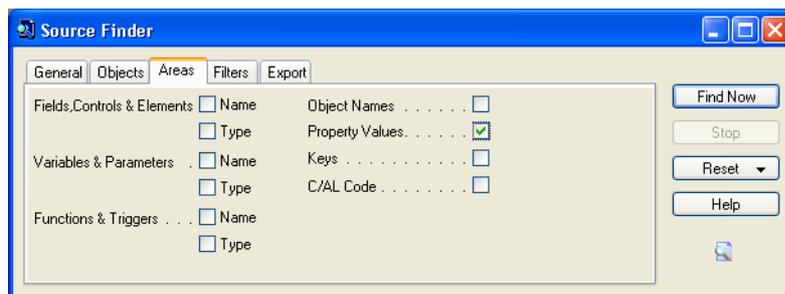
Searching for Properties

If you want to know in which forms the properties BlankZero and DecimalPlaces are changed, follow this procedure:

- 1 On the Tools menu, click Source Finder. The **Source Finder** window appears:
- 2 Click the **Objects** tab and insert a in the **Forms** field only:

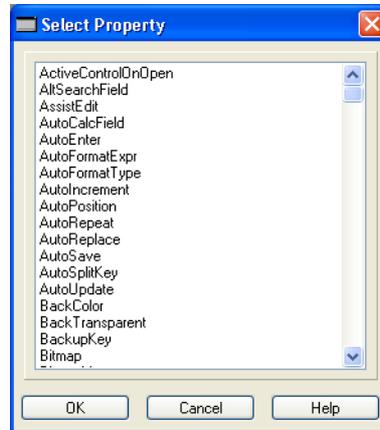


- 3 Click the **Areas** tab and insert a in the **Property Values** field only:



- 4 Click the **Filters** tab.

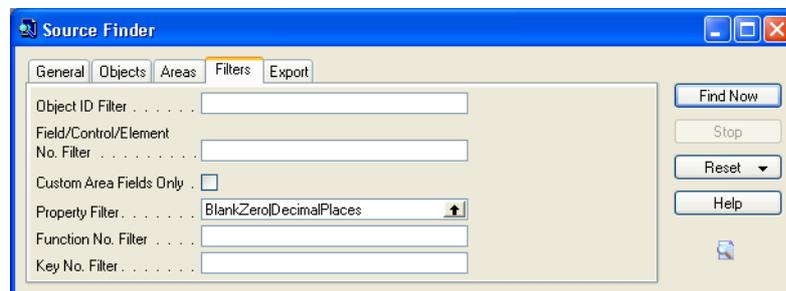
- Click the lookup button (↑) in the **Property Filter** field. The **Select Property** window appears:



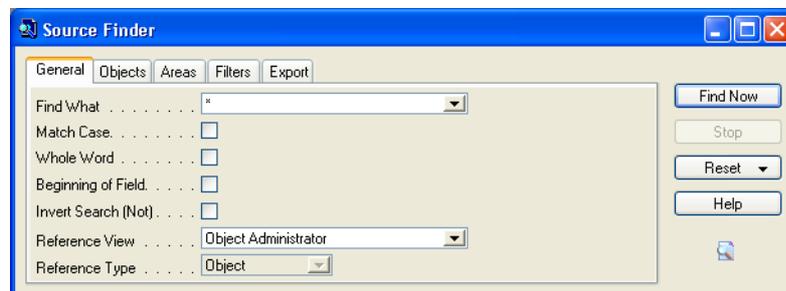
- Mark both the BlankZero and DecimalPlaces properties by holding down the SHIFT key as you click the properties.

- Click OK.

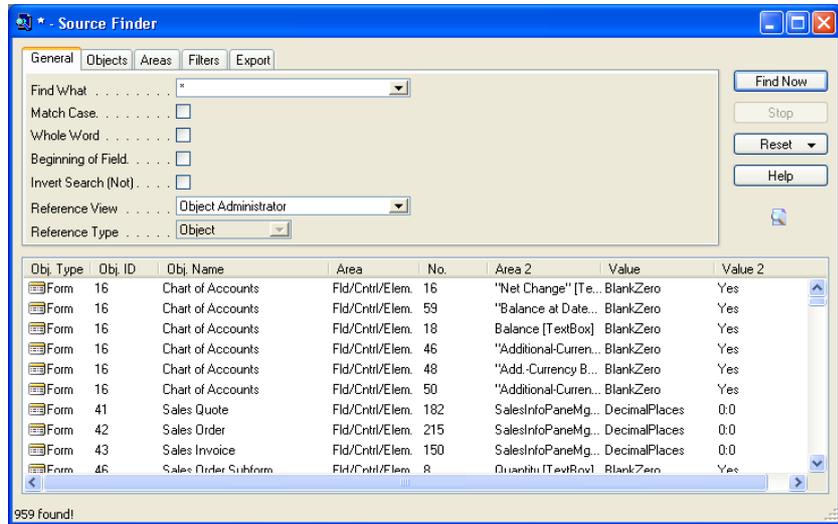
The program copies the properties to the **Property Filter** field in the **Source Finder** window:



- Click the **General** tab and enter * in the **Find What** field:



- Click the Find Now button to start the search. The **Source Finder** result window appears:

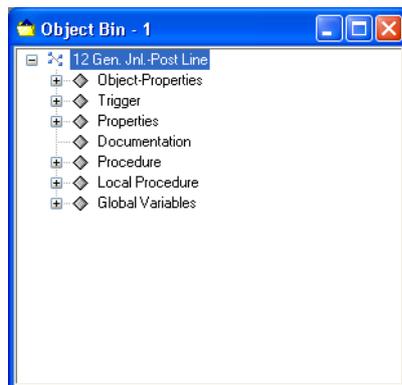


You can click the header of the **Value** column to sort the lines based on the values in this column.

Searching for Comments

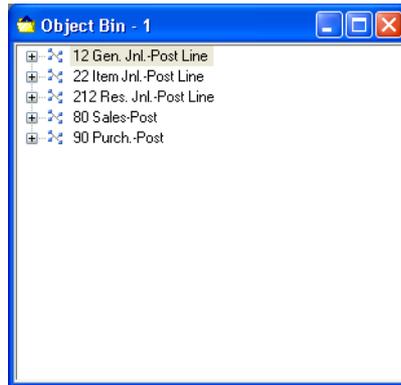
If you use comments to document the code lines, you can follow the procedure below to get a list of all comments in the major posting codeunits (codeunits 12, 22, 212, 80, 90):

- Click codeunit 12, **Gen. Jnl.-Post Line** in the **Object Administrator** window.
- On the Tools menu, click Object Bin to copy this object to a new **Object Bin** window. The **Object Bin** window appears:

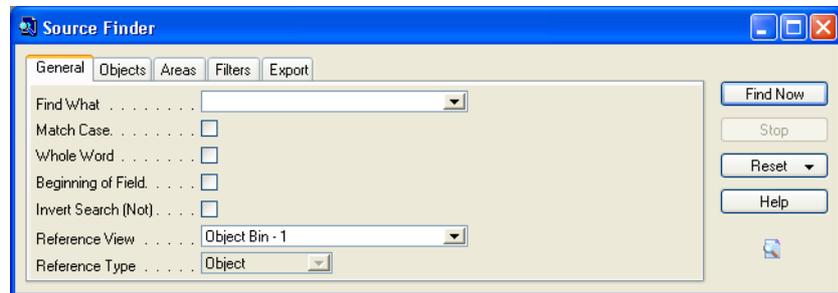


- Collapse all lines in the **Object Bin** window.
- Click codeunit 22, **Item Jnl.-Post Line** in the **Object Administrator** window and drag and drop it into the **Object Bin** window.

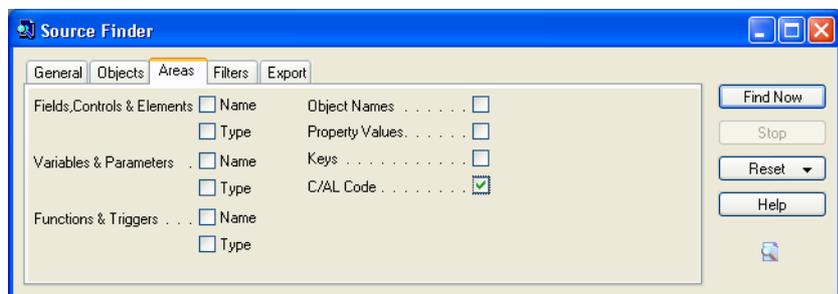
- 5 Click codeunit 212, **Res. Jnl.-Post Line** in the **Object Administrator** window and drag and drop it into the **Object Bin** window.
- 6 Click codeunit 80, **Sales-Post** in the **Object Administrator** window and drag and drop it into the **Object Bin** window.
- 7 Click codeunit 90, **Purch.-Post** in the **Object Administrator** window and drag and drop it into the **Object Bin** window.



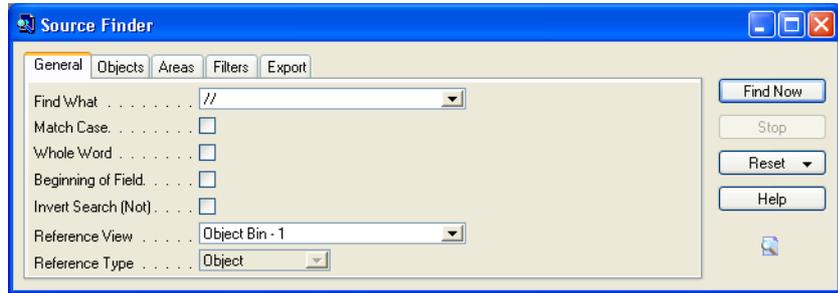
- 8 On the Tools menu, click Source Finder. The **Source Finder** window appears:
- 9 Click the dropdown button (▼) in the **Reference View** field and select **Object Bin - 1**:



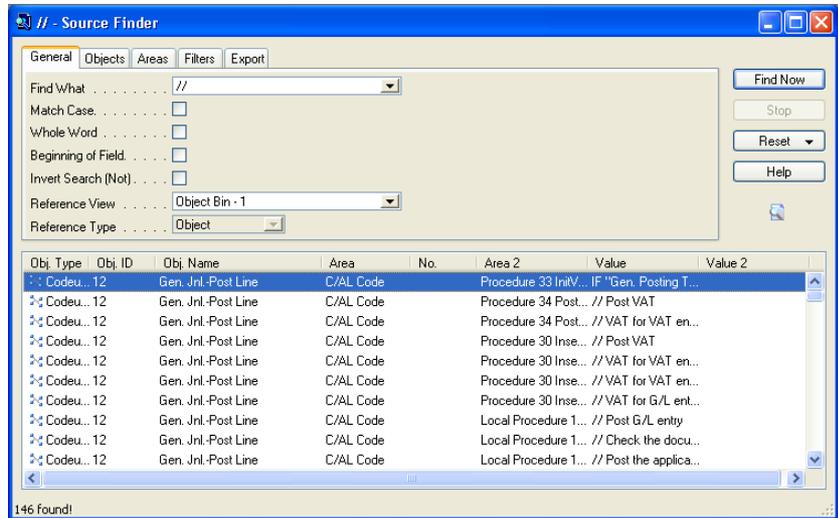
- 10 Click the **Areas** tab and insert a check mark (✓) in the **C/AL Code** field only:



11 Click the **General** tab and enter // in the **Find What** field:



12 Click the Find Now button to start the search. The **Source Finder** result window appears:



13 Click the first line in the **Source Finder** result window.

- 14 On the Tools menu, click Code Viewer to see the code lines in this local procedure.
The **Code Viewer** window appears:

The screenshot shows a window titled "Codeunit 12 Gen. Jnl.-Post Line - Code Viewer". On the left is a tree view with nodes for Documentation, Trigger, Procedure, and Local Procedure. The main area displays the following SQL code:

```

IF "Gen. Posting Type" <> 0 THEN BEGIN // None
    VATPostingSetup.GET("VAT Bus. Posting Group","VAT Prod. Posting G
    TESTFIELD("VAT Calculation Type",VATPostingSetup."VAT Calculation
CASE "VAT Posting" OF
    "VAT Posting"::"Automatic VAT Entry":
        BEGIN
            GLEEntry."Gen. Bus. Posting Group" := "Gen. Bus. Posting Gro
            GLEEntry."Gen. Prod. Posting Group" := "Gen. Prod. Posting G
            GLEEntry."VAT Bus. Posting Group" := "VAT Bus. Posting Group
            GLEEntry."VAT Prod. Posting Group" := "VAT Prod. Posting Gro
            GLEEntry."Tax Area Code" := "Tax Area Code";
            GLEEntry."Tax Liable" := "Tax Liable";
            GLEEntry."Tax Group Code" := "Tax Group Code";
            GLEEntry."Use Tax" := "Use Tax";
        CASE "VAT Calculation Type" OF
            "VAT Calculation Type"::"Normal VAT":
                BEGIN
                    IF "VAT Difference" <> 0 THEN BEGIN
                        GLEEntry.Amount := "VAT Base Amount (LCY)";
                        GLEEntry."VAT Amount" := "Amount (LCY)" - GLEEntry.Am
                        GLEEntry."Additional-Currency Amount" := "Source Cur
                        IF "Source Currency Code" = GLSetup."Additional Rep
                            AddCurrGLEEntryVATAmt := "Source Curr. VAT Amount"
                        ELSE
                            AddCurrGLEEntryVATAmt := CalcLCYToAddCurr(GLEEntry.
    END
    END

```

The status bar at the bottom of the window shows "InitVat() Procedure 33" on the left, "1:156" in the middle, and "ReadOnly Num" on the right.

If you go back to the **Source Finder** window and open the **Code Viewer** window based on another line again, the existing **Code Viewer** window will be updated.

Chapter 5

Compare & Merge Tool

The Navision Developer's Toolkit - Compare & Merge Tool is a utility designed to help document, compare, upgrade and maintain modified customer solutions. This tool can also be used for implementing or upgrading add-on solutions in a modified customer version.

This chapter contains basic information about the components of the Compare & Merge Tool.

This chapter contains the following sections:

- Overview of Compare & Merge Tool
- Merge Setup
- Compare & Merge
- Compare Two Versions
- Export Objects

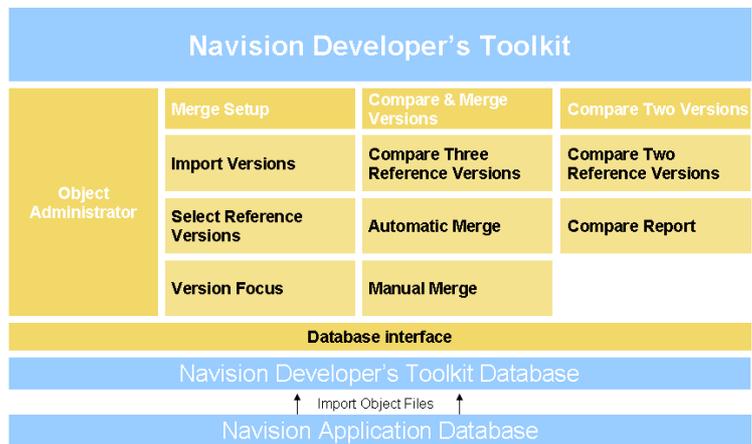
5.1 OVERVIEW OF COMPARE & MERGE TOOL

Navision Developer's Toolkit - Compare & Merge Tool is a collection of tools designed to help document, compare, upgrade and maintain modified customer solutions. This tool can also be used for implementing or upgrading add-on solutions in a modified customer version.

The basic idea behind Navision Developer's Toolkit is to have one database with all information about objects to be used during an upgrade and one program that provides the user with all functions needed during an upgrade process.

The picture below shows the structure and the elements of Navision Developer's Toolkit - Compare & Merge:

Compare & Merge - Overview



Navision Developer's Toolkit Database

The Compare & Merge Tool is based on the object data that is stored in the Navision Developer's Toolkit database. Before you can start a compare and merge process, you must import Navision objects from the application databases into the Navision Developer's Toolkit database either in text format or from a Navision client. You can import objects from different Navision application versions to be used as reference versions in the compare and merge process.

Reference Versions

Each version used in the compare and merge process is called a reference version. Basically, the Compare & Merge Tool compares three different versions of an object in order to create a new version:

Old Base Version

This version usually contains objects from a Navision standard version. For example, this could be Navision version W1 3.70.

Current Custom Version

This version is based on the Old Base Version and contains customized objects. For example, this could be Navision version W1 3.70 with changes for a customer.

New Base Version

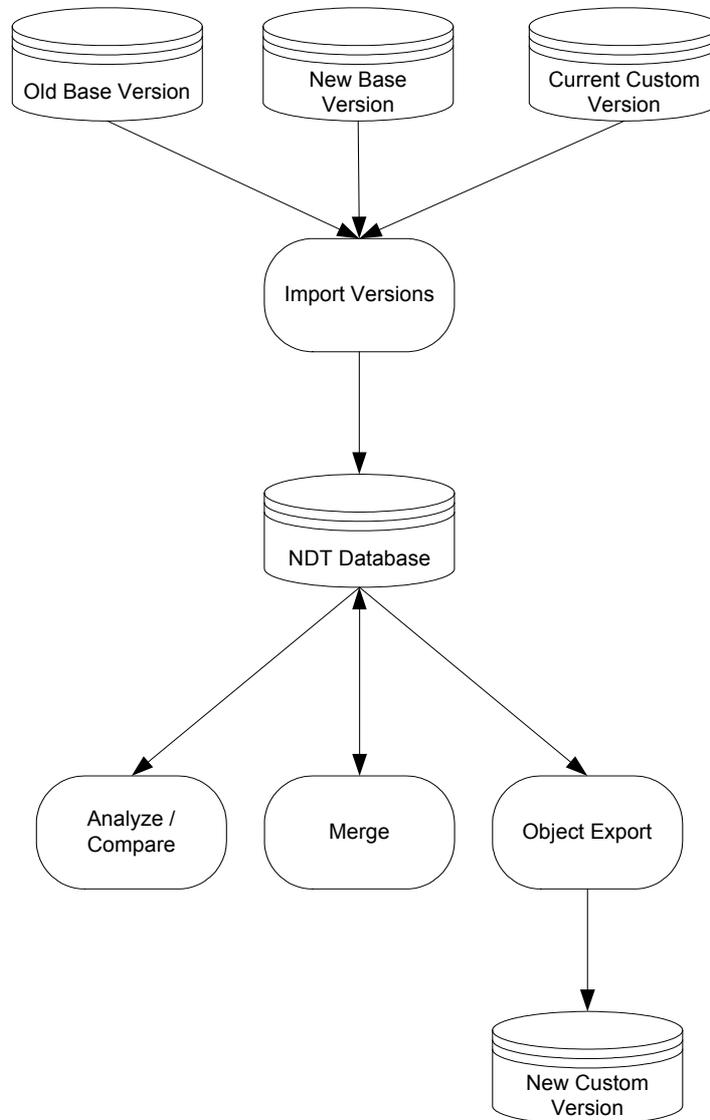
This version contains objects from a new Navision standard version and is based on the Old Base Version. For example, this could be Navision version W1 4.00.

New Custom Version

The Compare & Merge tool creates this version based on the three reference versions (Old Base, Current Custom, New Base).

Process Overview

The picture below shows how the program uses the reference versions and the basic steps of an upgrade process:



The update process can be split up in these major steps:

Define Reference Versions

The user must decide which versions they need during the upgrade process. For example, they might need the old Navision standard version, the old customer version and the new Navision standard version.

Export Objects from Navision Application Database

The objects from reference versions must be exported from the Navision application database in text format.

Import Objects in Navision Developer's Toolkit Database

All objects from reference versions must be available in the Navision Developer's Toolkit database. If a version does not exist in this database, the objects must be available in text format in order to import them to the Navision Developer's Toolkit database.

Analyze and Compare Objects

The user analyzes each reference version with the cross-reference functions in Navision Developer's Toolkit to explore the structure and the usage of objects and object details. They can first compare two versions, for example the Old Base Version and the New Base Version. The comparison of the three reference versions is the preparation of the merge process and leads to a suggestion for the action in the later merge process. All differences in the compared versions are marked in the corresponding version or in all conflicting versions, if there are conflicts.

Automatic Merge

Based on the defined reference versions, the user starts the automatic merge process. The system creates a new version as a merge result. All objects that could not be merged automatically are marked. These objects must then be merged manually.

All differences in the compared versions are colored in the corresponding version or in all conflicting versions, if there are conflicts.

Manual Merge of an Object

If an automatic merge is not possible due to conflicts between the reference versions, the user must decide manually which object version should be used as the new version. The manual merge could be necessary for the whole object or only for one or more areas in an object. After manual merge, the changes are saved in the new version.

Export Objects

All objects from the new version must be exported from the Navision Developer's Toolkit database in either a text format or directly to the Navision application database.

Import New Object Version in Navision Application Database

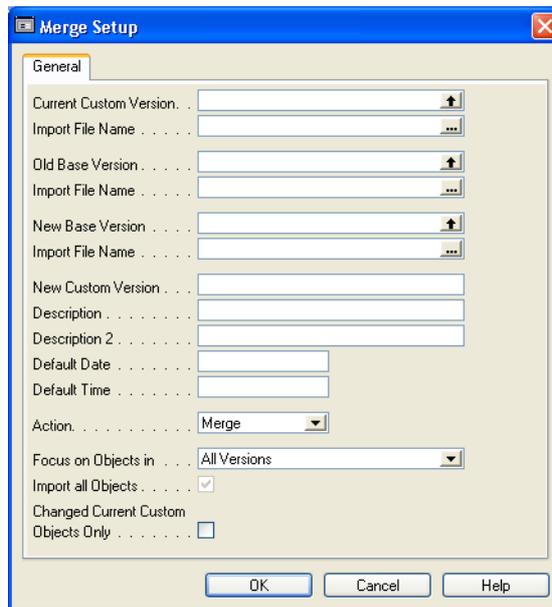
If the objects were exported from the Navision Developer's Toolkit database in text format, they must be imported into the Navision application database. After this, all objects must be compiled.

5.2 MERGE SETUP

The Merge Setup contains all information required to start a compare and merge process. You can use the **Merge Setup** window or the Merge Wizard to enter this information.

Entering Merge Setup

In the **Merge Setup** window, you can fill in the necessary parameters to merge the custom and base versions into a new version:



For each reference version (Current Custom, Old Base, New Base Versions) you can either select an already imported version, or you can enter an **Import File Name** that contains Navision objects in text format. If you do not have a version already imported you must also enter a code in the **Version** field. You must also enter a code in the field **New Custom Version**.

Depending on the merge scenario, you can set different options to speed up the compare and merge process. The following sections contain the setup for three different scenarios.

Merge a Full Version

The main goal of this scenario is to create a fully merged version that contains all objects from the reference versions. You can use the default setting *All Versions* in the **Focus on Objects in** field. The program handles all objects that are in at least one of the reference versions. If a version must be imported, the program imports all objects. During the compare and merge process, each object is either simply copied or merged into the New Custom Version. Merging requires the most time in this scenario.

Merge Only Customized Objects

In this scenario, you have exported only the customized objects from the Navision application database and you want to update these objects. In the field **Focus on Objects in**, you must select *Current Custom Version*. With this setting, the program only compares and merges objects that exist in the Current Custom Version. As a merge result, you receive all objects from the Current Custom Version merged with the same objects from the New Base Version. This is a fast way to do the merge process, if only your Current Custom Version contains changed objects.

If you do not check the field **Import all Objects**, the program only imports objects from reference version files that exist in the Current Custom Version.

Merge a Service Pack

If you receive a service pack for a Base Version, you can follow this scenario, where you merge only objects from the service pack. Select *New Base Version* in the field **Focus on Objects in** and do not place a check mark in **Import all Objects**. The program compares and merges only objects that exist in the New Base Version. As a merge result, you receive all objects from the Current Custom Version merged with the same objects from the New Base Version. This is fast way to do the merge process, if only your New Base Version contains some changed objects.

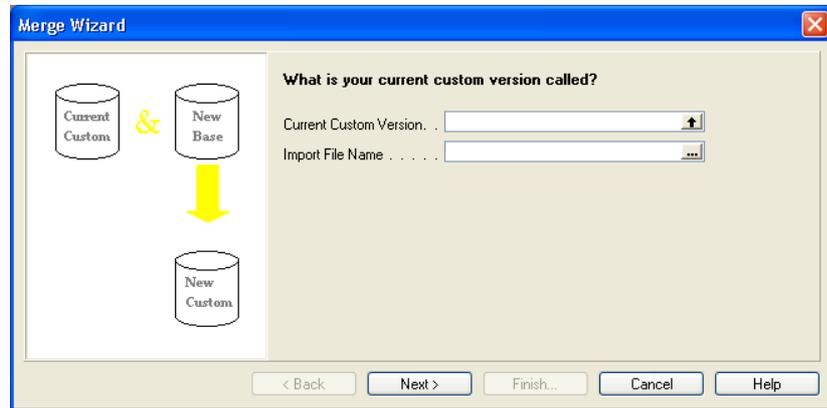
Please refer to the online Help if you need more information about the fields in the **Merge Setup**.

Note

.....
All versions of objects that you want to use in one compare and merge process must be exported with the same Navision client version (for example, Navision 4.00), because of changes in the Navision text file format between versions.
.....

Merge Wizard

You can use the **Merge Wizard** windows to enter the Merge Setup information. The Merge Wizard guides you through the setup procedure. In the first window of the wizard you select the current custom version:



All information that you enter in the Merge Wizard is saved in the Merge Setup. The fields are exactly the same as in the **Merge Setup** window.

Start Compare & Merge Process

Click OK in the **Merge Setup** window or Finish in the last window of the **Merge Wizard** to start the compare and merge process. The program first calculates the required database space and checks the available space in the Navision Developer's Toolkit database. A warning appears before the compare and merge process starts if you need to expand the database.

The program first imports the versions, if required, and then starts the compare and merge process. After this process is finished, the program shows a summary of the process. After you confirm this summary, the result of the automatic merge is shown in the **Compare & Merge** window.

5.3 COMPARE & MERGE

After you have started the compare and merge process and the versions have been imported, if required, the system first compares all three reference versions and tries to create the New Custom Version. This is called the automatic merge process.

Automatic Merge Process

During the automatic compare and merge process, the program compares the reference versions to find out which objects or parts of them are different and how this information can be merged.

During the import, for each object, the program creates a computed value called a checksum. The checksum includes all object data, except the fields **Date**, **Time**, **Version List** and **Modified**. In the first step, the checksum is compared for each object. Based on this checksum, the program decides whether an object can be easily copied to the New Custom Version or if a merge on the detail level is required, if the object is different in all three reference versions.

During the detail-level merge, the program first compares Current Custom Version and Old Base Version. After that, it compares the New Base Version and Old Base Version. The result of this comparison leads to an action to create the New Custom Version. If there is a conflict because a detail has been changed in the Current Custom Version and the New Base Version, the program copies the information from the version that is defined in the **Merge Options**.

If you want to know more about the merge rules, please read the online Help.

After the program has finished the automatic merge process, the status window displays the number of objects that have been copied and merged and how many objects have conflicts and overlapping controls. After you close this status window, the **Compare & Merge** window appears.

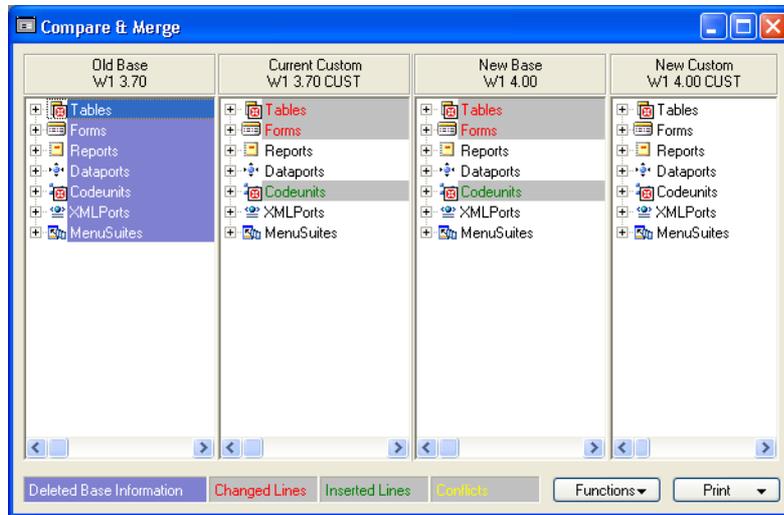
Manual Merge Process

After the automatic merge process is finished, you can start the manual merge process.

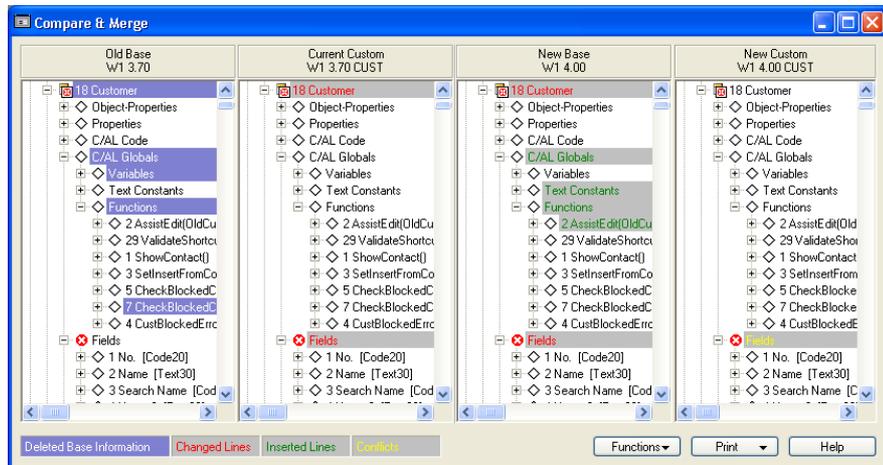
The manual merge process is always required when the automatic merge is not possible due to conflicts between the reference versions. You must decide manually which object version you want to use as a new version. The manual merge could be necessary for the whole object or only for one or more areas in an object. You can do this in the **Compare & Merge** window.

Compare & Merge Window

The **Compare & Merge** window below shows an example merge:



All columns are synchronized vertically and horizontally. You can expand objects to see the details and collapse objects to reduce the object details shown in the **Compare & Merge** window.



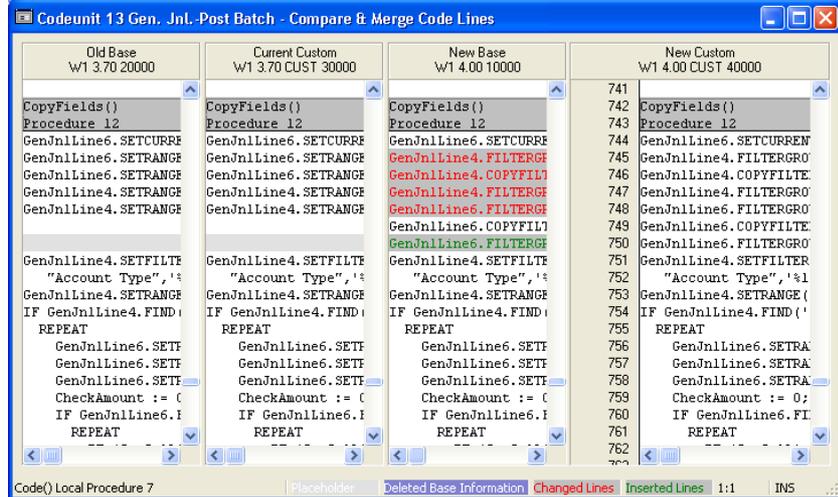
As an example, this window contains only some objects in each version.

The coloring is used to show the differences between the reference versions. A conflict bitmap indicates that the program could not merge automatically. You should always check the suggestion in the New Custom Version and either mark this suggestion as accepted or change it.

You can change all details of an object in the New Custom Version only. If you want to undo your changes in an object, you can click Rermerge in the shortcut menu on the object level. This rermerge will overwrite all your changes in this object.

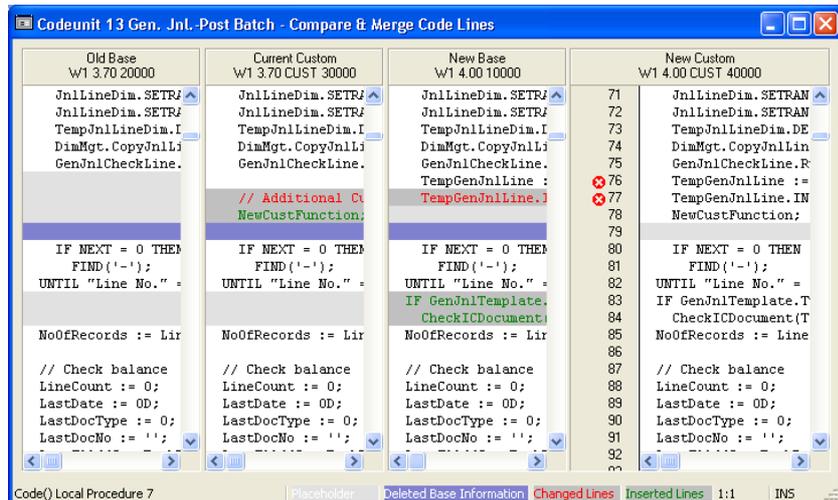
If you want to see the C/AL code for an object, open the **Compare & Merge Code Lines** window.

Compare & Merge Code Lines Window



The **Compare & Merge Code Lines** window shows all C/AL code lines for an object in each version that is used in the current compare and merge process. You can open the **Compare & Merge Code Lines** window from a selected object or a specific trigger.

The coloring indicates the status of each code line. The conflict bitmap is shown in front of the corresponding line in the New Custom Version:



You can change all lines of an object in the New Custom Version only. If you start remerging for an object, the program also remerges the code lines and your changes will be overwritten.

When you click the Tools menu and click C/AL Globals, the **Compare & Merge C/AL Globals** window appears. This window shows the global variables for the current object and is not editable. Click the Tools menu and click C/AL Locals to open the

Compare & Merge C/AL Locals window. This window shows the local variables for the current trigger or function and is also not editable.

Note

.....
The code line editor does not perform any syntax check in code lines. Syntax errors in code lines might cause problems during import or compilation in the application database.
.....

Interrupting a Merge Process

 If you want to interrupt the automatic merge process, you can click Stop in the status window. The program finishes the current object and saves all merge information created so far. After you reopen the **Compare & Merge** window, even if you have closed the program, the program reminds you that you have an unfinished merge process. You can decide whether you want to proceed with it now or later.

During the manual merge process, the program saves all changes as soon as you click on another object or when you close the **Compare & Merge** window. You can close the program and continue whenever you want.

Note

.....
It is only possible to have one merge project per Navision company. If you want to process more than one merge process in parallel, you must create a new company in the same database or in a new database.
.....

Check Overlapping Controls

 After finishing the merge of forms, the system automatically checks forms for overlapping controls. The result is shown in the status window after the automatic merge process. In the **Compare & Merge** window, click the Print button and click Overlapping Controls to see a list of overlapping controls in forms and request forms in reports.

These overlapping controls can be rearranged either directly in Navision or in Navision Developer's Toolkit by changing the properties.

Note

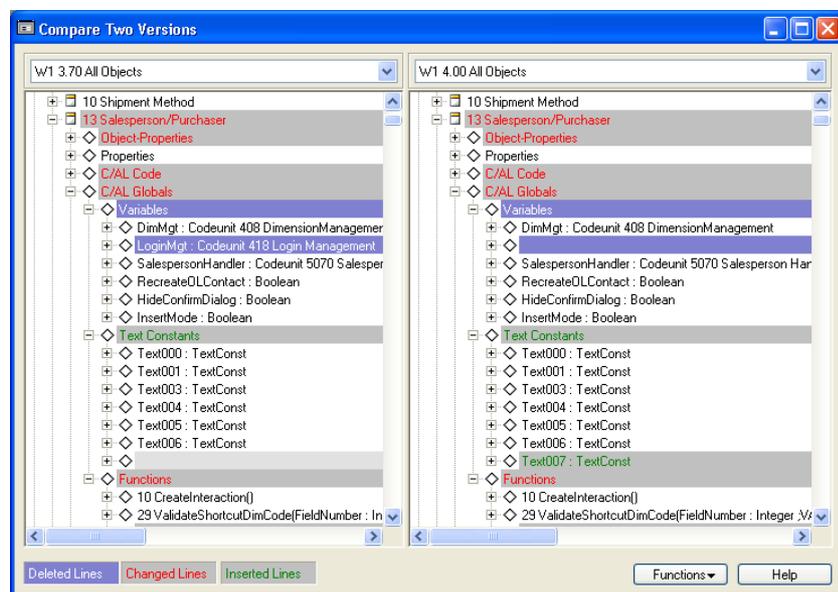
.....
Only forms and request forms in reports are checked for overlapping controls.
.....

5.4 COMPARE TWO VERSIONS

You can use the **Compare Two Versions** window to compare two versions that you have already imported in Navision Developer's Toolkit. The comparison is based on the object structure and gives you fast access to specific areas.

Compare Two Versions Window

The **Compare Two Versions** window shows the objects of both versions and highlights the differences between them:

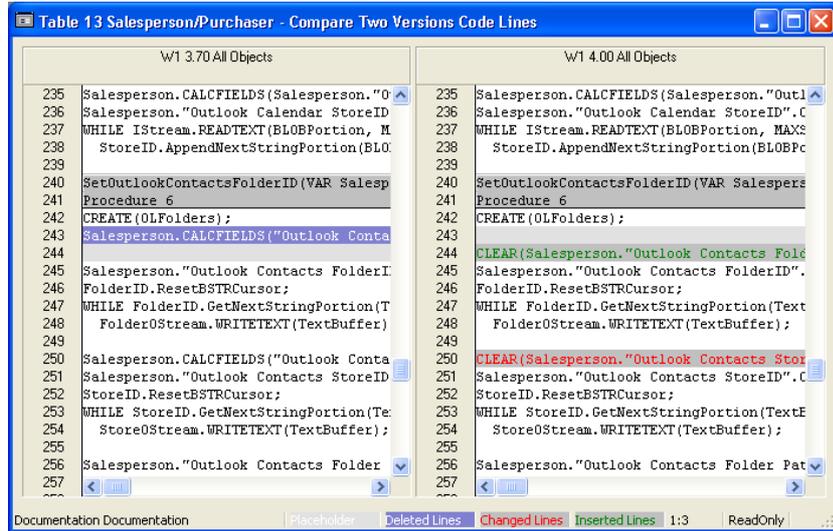


Each column displays all objects of the versions you have selected in each column header. When you select a version, the program starts the comparison automatically.

Both columns are synchronized vertically and horizontally and coloring is used to show the differences on all levels. When an item in a tree is not expanded, the item represents the coloring of details of the item. If you want to set a filter on object type or object ID, click on the Functions button in this window.

Compare Two Versions Code Lines

The **Compare Two Versions Code Lines** window displays all C/AL code lines for an object in each version that is used in the **Compare Two Versions** window. You can start the **Compare Two Versions Code Lines** window for a selected object by clicking the Tools menu and selecting Code Viewer, or by using the shortcut menu or by pressing the F9 key.



The coloring indicates the status of each code line. You can use the icons on the menu bar to jump to the previous or next difference.

Printing the Compare Two Versions Report

If you want to have a printout of the comparison of two versions, you can click Print in the File menu. A request form appears where you can set filter and other options for this report.

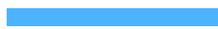
Attention

.....

The printout of this report and a jump to the last page in the preview may take some time, because both versions have to be compared again before the printing. You can reduce the time by printing differences only.

.....

Export Compare Two Versions

 You can export the information from the **Compare Two Version** window by clicking the File menu, pointing to Export, and clicking As Text File. A request form appears where you can set filter options and other options for this export.

Attention

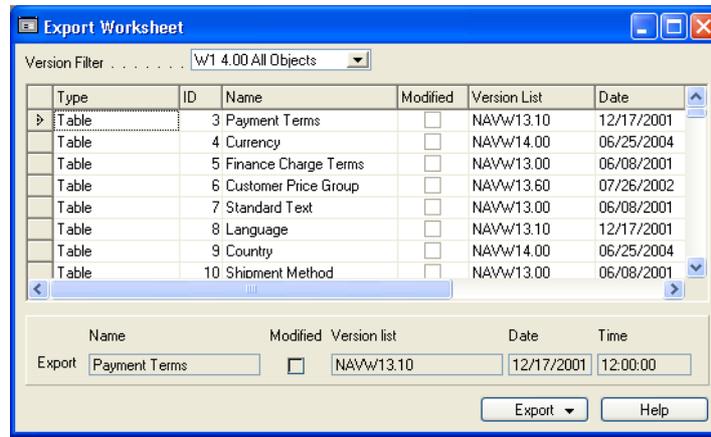
.....
This export may take some time, because both versions have to be compared again before the export. You can reduce the time by exporting differences only.
.....

5.5 EXPORT OBJECTS

Navision Developer's Toolkit allows you to export objects that have been imported before or objects that have been created during the merge process.

Export Worksheet

You can use the **Export Worksheet** to export objects from Navision Developer's Toolkit either directly to a Navision application database or to a file in Navision text format. On the File menu, point to Export, and click Navision Text File or Navision Client to open this window. The **Export Worksheet** appears:



After you have selected the object version in the header of the **Export Worksheet**, all objects from this version are shown. You can filter objects by setting field filters or by marking lines. Click the Export button and click To Navision Text File... to export all marked objects as a Navision text file, or click To Navision Client... to export all marked objects directly to a Navision database.

If you have chosen to export to a Navision text file, a request form appears in which you can select whether or not to print one file per object and specify the file name for the text file. Click OK to confirm your selection, and all the marked objects are exported to a Navision text file.

If you have chosen to export to a Navision client, the **Select Navision Client** window opens, from which you can select the appropriate Navision client. Click OK to confirm, and all marked objects are exported to the Navision database.

Chapter 6

Installing and Starting Navision Developer's Toolkit

Navision Developer's Toolkit is delivered with an installation program. This chapter will show you how to use it.

This chapter contains the following section:

- Installing and Uninstalling
- Starting Navision Developer's Toolkit and Connecting to a Database

6.1 INSTALLING AND UNINSTALLING NAVISION DEVELOPER'S TOOLKIT

You use a special installation program to install Navision Developer's Toolkit on a hard disk. You cannot simply copy the files because they are packed in a special way. The program guides you through the installation process, asking you questions you must answer along the way.

Attention

.....
Before you begin installing the program from the program CD, make a copy of it. Then use the copy to install the program and save the original as a backup.
.....

Installing the Navision Developer's Toolkit

Before you start the installation procedure, make sure that you have checked the following requirements to install and run Navision Developer's Toolkit:

- Operating system Windows 98 or Windows NT/2000/XP.
- Computer with Pentium processor.
- Microsoft Business Solutions-Navision 4.00 or higher installed.
- C/FRONT W1 4.00 or higher.
- Microsoft Business Solutions-Navision license file (`fin.flf`) with permission to the granules C/FRONT and Navision Developer's Toolkit available.

To install the program, follow this procedure:

- 1 Start the operating system.

If you are installing the Navision Developer's Toolkit from a network drive or a CD-ROM, make sure that you are connected to the network server or CD-ROM drive.

- 2 Click the Start Button on the task bar and then click Run.
- 3 Type the path and name of the installation program. If you are installing from CD, for example, the path is `d:\setup`, where `d:` is the drive of your CD-ROM. If you install from a network drive, type the path and name of the installation program on the network. Click OK to start the installation program.

Under Windows 98/2000/XP or Windows NT version 4.0 or later, you can also use the Add/Remove Programs function in the Control Panel.

After you have started the installation program, the **Welcome** window appears.

- 4 To continue, click the Next button at the bottom of the window. If you decide not to continue with the installation, click Cancel. If you continue, the **Disclaimer** window appears.
- 5 Click Yes in the **Disclaimer** window to accept this disclaimer. The setup continues. Click No, if you do not accept this disclaimer. The setup closes.

- 6 If you accepted the disclaimer, the **Latest Release Information** window appears. In this window, you see the latest information for the current release. Click Cancel to cancel the installation program or click Next to continue.
- 7 The **Choose Destination Location** window appears next. Here, you must specify the folder where Navision Developer's Toolkit should be installed. You can accept the default (in this case, c:\Program Files\Microsoft Business Solutions-Navision Developers Toolkit) or you can click Browse to select a different location.
- 8 Click Next in the **Choose Destination Location** window. The **Select Copy Options** window appears.

In this window, you can decide if you want to select a Microsoft Business Solutions-Navision license file (for example `fin.flf`) and a Navision STX file (for example `fin.stx`). Both files are copied to the directory where you are installing the program. The STX file is used as default for keywords that are used during the import. You can change these keywords later. If you do not select an STX file, the program uses the default STX file from the setup program.

- 9 Click Next.

If you have checked one of the file options, the **Select File** window appears. After you have selected all files the **Start Copying Files** window appears.

If you did not check a file type option, the **Start Copying Files** window appears right away. This window shows a summary of all information that has been collected during the previous steps.

- 10 Click Next to start the actual installation. You can follow the progress of the installation in the **Setup Status** window.
- 11 As soon as the installation program has finished, a message appears. Click Finish. The window closes.

Uninstalling the Navision Developer's Toolkit

To uninstall Navision Developer's Toolkit in a Windows 98/2000/XP or Windows NT (version 4.0 or later) installation, use the Add/Remove Programs function in the Control Panel.

6.2 STARTING THE PROGRAM AND CONNECTING TO A DATABASE

The Navision Developer's Toolkit installation program automatically places Navision Developer's Toolkit in the program folder of your Windows Start menu.

Start Navision Developer's Toolkit to gain access to all the analyzing and development tools. In order to work with a database, you must open a local database or connect to a server database.

Note

.....
When you use Navision Developer's Toolkit the first time, you must setup a database. Please refer to chapter 2, "Navision Developer's Toolkit Database" or read the online Help to get more information about the setup of a Navision Developer's Toolkit database.
.....

To access a local database or to connect to a server database:

- 1 Click the File menu, point to Database, and click Open. The **Open Database** window appears.
- 2 In the **Path to C/Front** field, enter the location of the `cfront.dll` file or the `cfrontsql.dll` file, depending on the server type you are using. The default value in this field points to the installation directory. You can click the AssistButton to open a dialog window from which you can browse and select the file.
- 3 If you want to connect to a server database, select the server name and the net type. Make sure that the server has already been started.
- 4 Enter the database name and specify whether you want to use the Windows authentication or the database server authentication when opening the database. If you select database server authentication, enter the user ID and password as well.
- 5 Click OK to open the database.

INDEX

- C**
- Code Viewer window 41
- Color Options
 - setting up 42
- Compare & Merge 75
 - Automatic Merge Process 75
 - Code Lines 77
 - Compare & Merge Window 76
 - interrupting a Merge Process 78
 - Manual Merge Process 75
 - Merge Setup 72
 - Merge Wizard 74
 - Overview 68
 - Start Process 74
- Compare & Merge Tool
 - Features 6
- Compare Two Versions 79
 - Code Lines 80
 - Export 81
 - Print Report 80
- Concept of Navision Developer's Toolkit 2
- converting a database 19
- copying a database 10
- creating a new database 10
- creating an import version 12
- Current Custom Version 69
- D**
- database
 - connecting to a server 86
 - copying 10
 - creating a new 10
 - opening a local 86
- demonstration database
 - setting up 18
- E**
- example
 - analyzing usage of a table 51
 - finding usage of a field 54
 - finding usage of a procedure 57
 - searching for comments 62
 - searching for properties 60
 - viewing C/AL code lines 49
 - viewing object properties 49
- Export Objects 82
- Export Worksheet 82
- exporting application objects 11
- F**
- Functions 26
- I**
- Import Version Card 12
- Import Worksheet 16
- installing the program 84
- M**
- Merge Setup 72
- Merge Wizard 74
- Method Flow window 43
- N**
- New Base Version 69
- New Custom Version 69
- O**
- Object Administrator
 - using 25
- Object Administrator window 24
- Object Bin window 39
- object data
 - importing application objects 12
- Object Diagram window 38
- object export 82
- object file
 - exporting application objects 11
- Object Functions 26
- object import
 - creating an import version 12
 - importing object data 12
 - starting 15
 - using an import version 15
- Object Tools 39
- Object Tree window 36
- Object View
 - setting up the default 36
- Object Views 36
- Old Base Version 69
- Overlapping Controls
 - Print Report 78
- R**
- Reference Versions 69
- Relations from Objects 28
- Relations to Tables 26
- S**
- Source Analyzer
 - Features 4
 - overview 22

Index

| | |
|--------------------------------|----|
| Source Finder window | 45 |
| SQL Server | 19 |
| starting the program | 86 |
| U | |
| uninstalling the program | 85 |
| upgrading a database | 19 |
| W | |
| Where Used | 30 |
| Where Used Options | |
| setting up | 30 |
| Where Used With | 32 |
| Where Used With Options | |
| setting up | 32 |