NAVGATE®

LET THE CODING BEGIN!

User Guide
Navgate for Object Manager

Version: 1.0.10
Published: november 2015

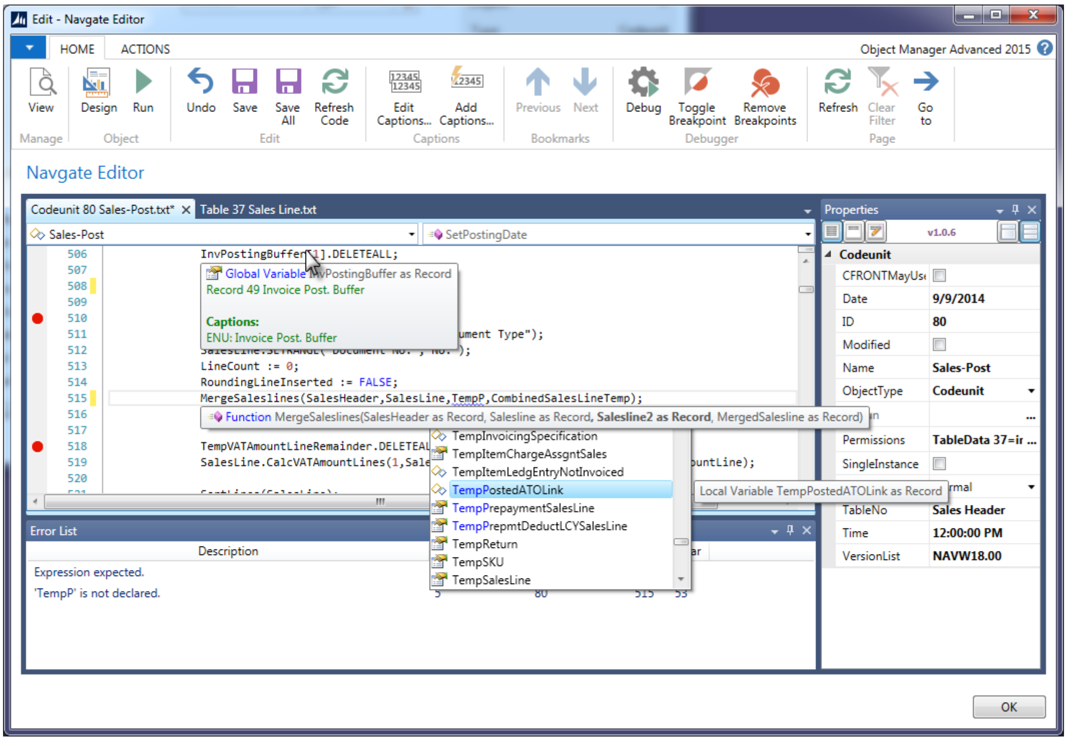
WICKITPRODUCTS.COM

CONTENTS

1 WHAT IS NAVGATE?

Navgate is a full development environment for **Microsoft Dynamics NAV**. It offers a syntax colored environment with all the features that modern development environment should have, like **IntelliSense** with **Code Completion, Parameter Info** and **Code Snippets**.

Navgate for Object Manager is specially build for **Object Manager** users. **Object Manager** features like **Where Used** and **Check Guidelines** are seamlessly integrated into **Navgate**.



```
InvPostingBuffer[1].DELETEALL;
LineCount := 0;
RoundingLineInserted := FALSE;
MergeSaleslines(SalesHeader,SalesLine,TempP,CombinedSalesLineTemp);
TempVATAmountLineRemainder.DELETEAL
SalesLine.CalcVATAmountLines(1,Sale
```

Global Variable InvPostingBuffer as Record
Record 49 Invoice Post. Buffer

Captions:
ENU: Invoice Post. Buffer

Function MergeSaleslines(SalesHeader as Record, Salesline as Record, **Salesline2 as Record**, MergedSalesline as Record)

- TempInvoicingSpecification
- TempItemChargeAssgntSales
- TempItemLedgEntryNotInvoiced
- TempPostedATOLink
- TempPrepaymentSalesLine
- TempPrepmtDeductLCYSalesLine
- TempReturn
- TempSKU
- TempSalesLine

Local Variable TempPostedATOLink as Record

Expression expected.
'TempP' is not declared.

Codeunit	
CFRONTMayUs	☐
Date	9/9/2014
ID	80
Modified	☐
Name	Sales-Post
ObjectType	Codeunit
Permissions	TableData 37=ir ...
SingleInstance	☐
TableNo	Sales Header
Time	12:00:00 PM
VersionList	NAVW18.00

Figure 1-1: Navgate

2 INSTALLATION

Navgate for Object Manager can be downloaded from wickitproducts.com/download. **Navgate for Object Manager** consists of two different parts: The ***NAV Add-in*** that has to be copied to your **Microsoft Dynamics NAV Client** folder and the ***Navgate Control Panel*** that will help you copy the **NAV Add-in** to the **Microsoft Dynamics Client** folder.

2.1 INSTALLING THE NAVGATE CONTROL PANEL

As you can see on the following picture you can use the **Navgate Control Panel** for different tasks. You can for example activate your license key and install the **NAV Add-in**.

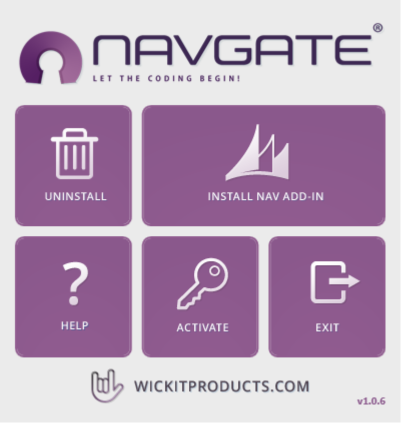


Figure 2-1: Navgate Control Panel

To install the **Control Panel** run the **Navgate Setup** executable.

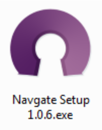
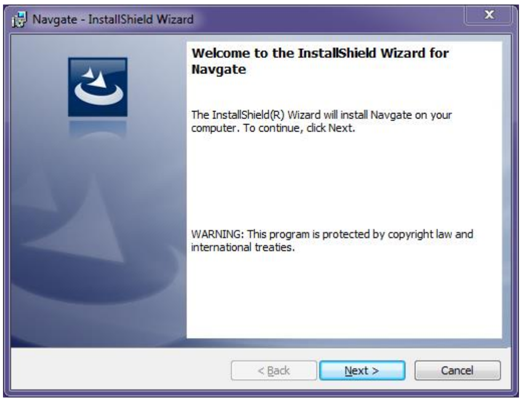


Figure 2-2: Navgate Setup

The **Installation Wizard** starts. Confirm acknowledgement of copyright protection and click **Next**.



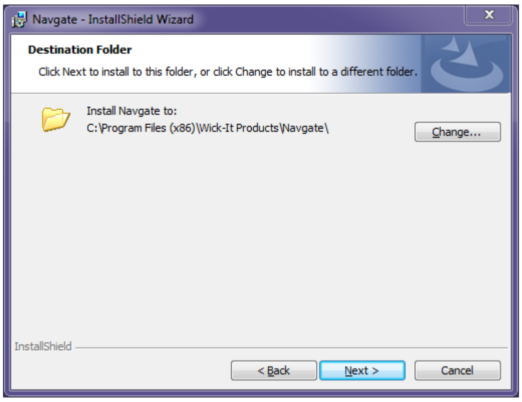
Welcome to the InstallShield Wizard for Navgate

The InstallShield(R) Wizard will install Navgate on your computer. To continue, click Next.

WARNING: This program is protected by copyright law and international treaties.

Figure 2-3: Navgate Installation Wizard 1

Click **Change** to install to a different folder, click **Next** to continue installation.



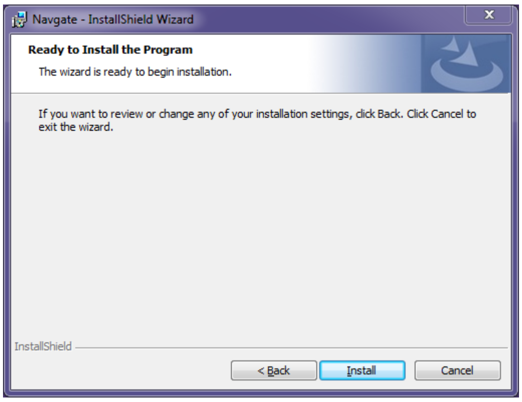
Destination Folder

Click Next to install to this folder, or click Change to install to a different folder.

Install Navgate to:
C:\Program Files (x86)\Wick-It Products\Navgate\

Figure 2-4: Navgate Installation Wizard 2

At next step the **Installation Wizard** is ready, click **Install** to begin installation.



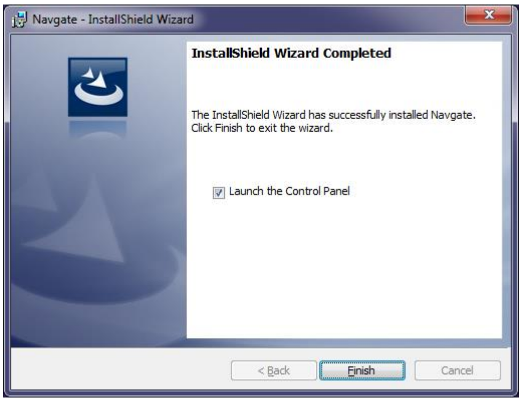
Ready to Install the Program

The wizard is ready to begin installation.

If you want to review or change any of your installation settings, click Back. Click Cancel to exit the wizard.

Figure 2-5: Navgate Installation Wizard 3

When **Navgate** was successfully installed click **Finish**.



InstallShield Wizard Completed

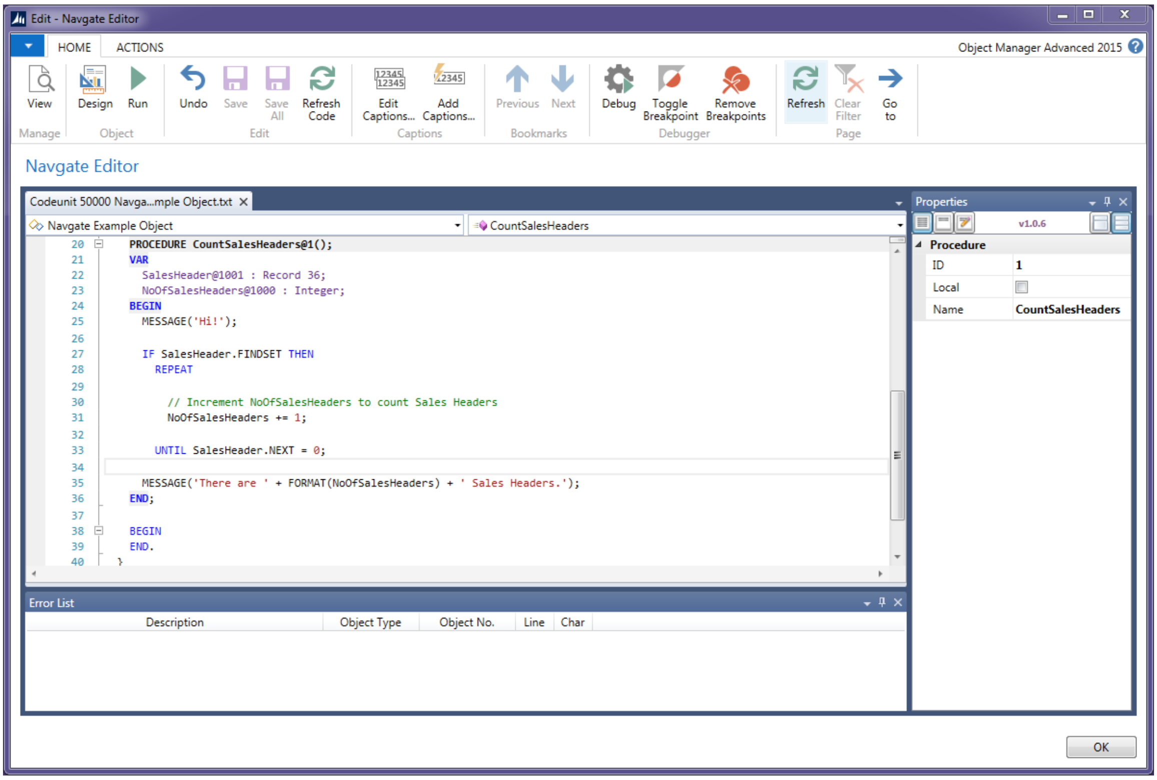
The InstallShield Wizard has successfully installed Navgate. Click Finish to exit the wizard.

☑ Launch the Control Panel

Figure 2-6: Navgate Installation Wizard 4

2.2 INSTALLING THE NAV ADD-IN

The **NAV Add-in** is an add-in that runs in the **Microsoft Dynamics NAV Client for Windows**. The add-in can be installed with the **Navgate Control Panel** but it is also possible to copy the files to the **Add-ins** directory of your **Microsoft Dynamics NAV** installation. This chapter describes the differences between these two options.



```
  PROCEDURE CountSalesHeaders@1();
  VAR
    SalesHeader@1001 : Record 36;
    NoOfSalesHeaders@1000 : Integer;
  BEGIN
    MESSAGE('Hi!');

    IF SalesHeader.FINDSET THEN
      REPEAT

        // Increment NoOfSalesHeaders to count Sales Headers
        NoOfSalesHeaders += 1;

      UNTIL SalesHeader.NEXT = 0;

    MESSAGE('There are ' + FORMAT(NoOfSalesHeaders) + ' Sales Headers.');
  END;

  BEGIN
  END.
}
```

Procedure
ID 1
Local ☐
Name CountSalesHeaders

Figure 2-7 NAV Add-in

2.2.1 Install with the Control Panel

To install the **NAV Add-in** open the **Navgate Control Panel**. You can find it in the start menu. Click **Install NAV Add-in.**

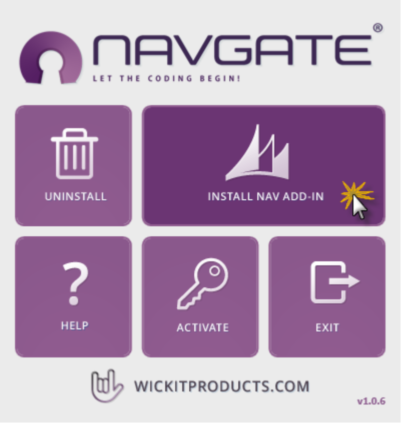
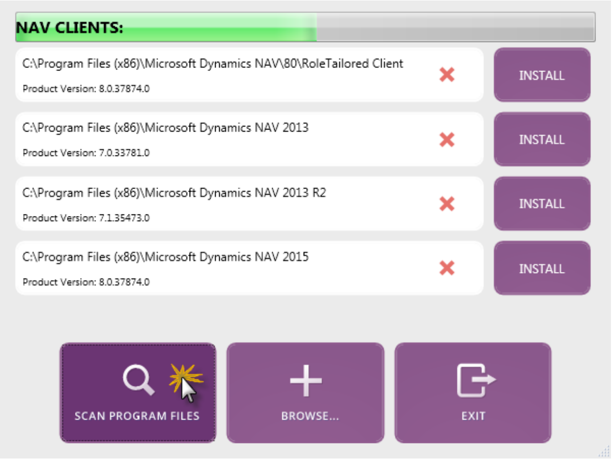


Figure 2-8: Navgate Control Panel

In the window that opens click **Scan Program Files** to scan this folder and automatically find the **NAV Clients**, or click **Browse** to locate the **NAV Clients** manually.



NAV CLIENTS:

C:\Program Files (x86)\Microsoft Dynamics NAV\80\RoleTailored Client
Product Version: 8.0.37874.0

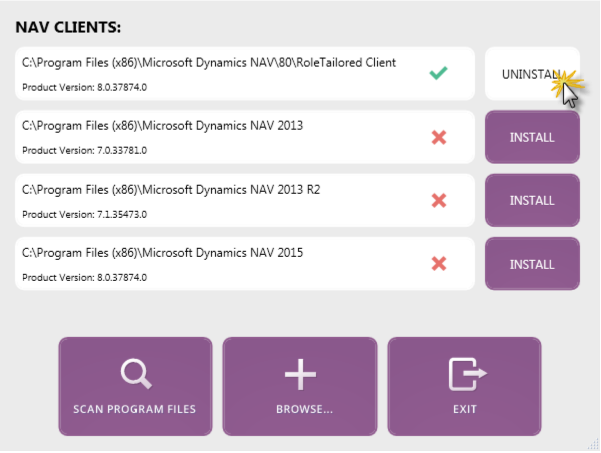
C:\Program Files (x86)\Microsoft Dynamics NAV 2013
Product Version: 7.0.33781.0

C:\Program Files (x86)\Microsoft Dynamics NAV 2013 R2
Product Version: 7.1.35473.0

C:\Program Files (x86)\Microsoft Dynamics NAV 2015
Product Version: 8.0.37874.0

Figure 2-9: Searching for Dynamics NAV Clients

After the **NAV Clients** were located, choose the **NAV Client** you wish to install and click **Install.** After successful installation the **Install** action will change to **Uninstall** action indicating that the add-in is present.



NAV CLIENTS:

C:\Program Files (x86)\Microsoft Dynamics NAV\80\RoleTailored Client
Product Version: 8.0.37874.0

C:\Program Files (x86)\Microsoft Dynamics NAV 2013
Product Version: 7.0.33781.0

C:\Program Files (x86)\Microsoft Dynamics NAV 2013 R2
Product Version: 7.1.35473.0

C:\Program Files (x86)\Microsoft Dynamics NAV 2015
Product Version: 8.0.37874.0

Figure 2-10: Install Dynamics NAV Add-in

NOTE: If you do not have write permissions on your Dynamics NAV Client folder launch the Control Panel with administrator rights. You can access the 'Run as Administrator' menu option by clicking Shift + Right Click of you mouse on the executable or shortcut.

2.2.2 Install by copying files

It is not necessary to use the control panel to install the **NAV Add-in**. It is also possible to copy all the needed files to the **Microsoft Dynamics NAV Client** folder manually. Open the **Navgate** folder in the **Program Files**. The default location is ***C:\Program Files (x86)\Wick-It Products\Navgate***.

Select the folder **RoleTailored Client** and copy it to your clipboard with **Ctrl+C**.

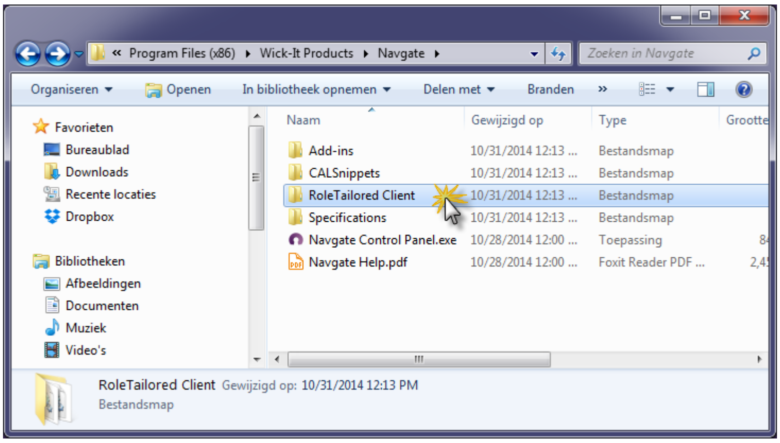


Figure 2-11: Copying RoleTailered Client Folder

Open the **Microsoft Dynamics NAV Client** folder where you want to install the **NAV Add-in** and paste the **RoleTailored Client** folder with **Ctrl+V**.

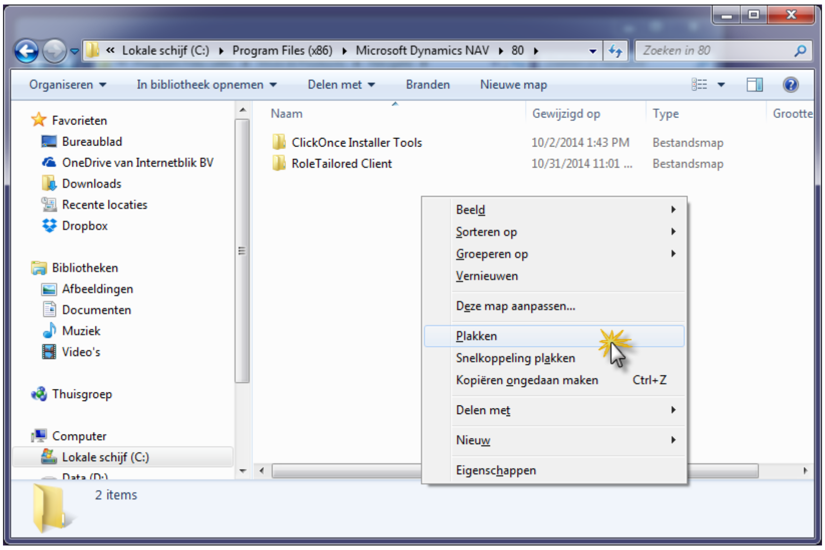
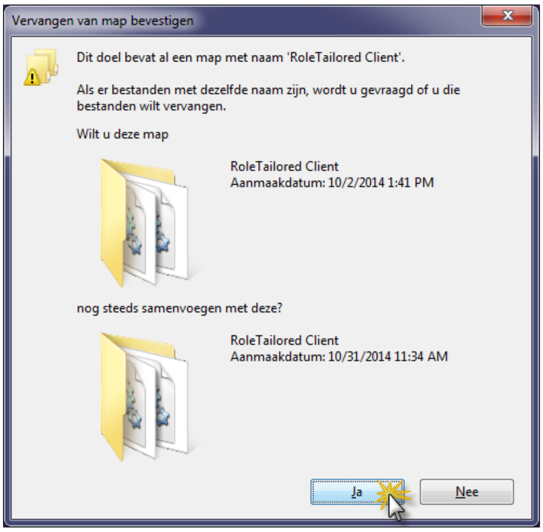


Figure 2-12: Pasting RoleTailerd Client Folder

Choose **Yes** to copy all the files to the **Microsoft Dynamics NAV Client** folder:


Vervangen van map bevestigen

Dit doel bevat al een map met naam 'RoleTailored Client'.

Als er bestanden met dezelfde naam zijn, wordt u gevraagd of u die bestanden wilt vervangen.

Wilt u deze map

RoleTailored Client
Aanmaakdatum: 10/2/2014 1:41 PM

nog steeds samenvoegen met deze?

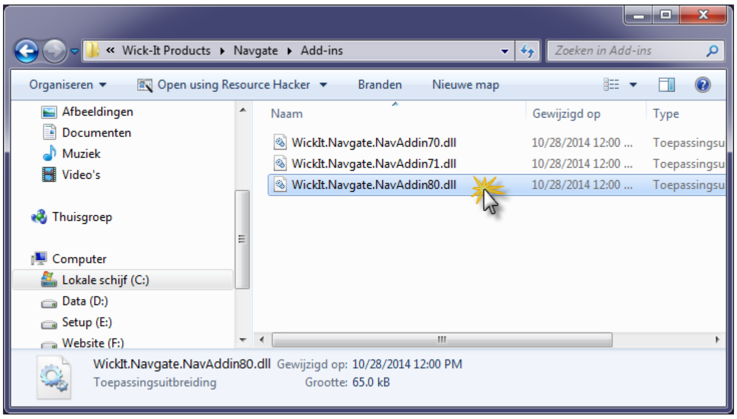
RoleTailored Client
Aanmaakdatum: 10/31/2014 11:34 AM

Figure 2-13: Confirm Pasting Files

The following files are copied to the **Microsoft Dynamics NAV Client** folder:

- \RoleTailored Client\Add-ins\NavgateEditor\ActiproSoftware.DataGrid.Contrib.Wpf.dll
- \RoleTailored Client\Add-ins\NavgateEditor\ActiproSoftware.Docking.Wpf.dll
- \RoleTailored Client\Add-ins\NavgateEditor\ActiproSoftware.PropertyGrid.Wpf.dll
- \RoleTailored Client\Add-ins\NavgateEditor\ActiproSoftware.Shared.Wpf.dll
- \RoleTailored Client\Add-ins\NavgateEditor\ActiproSoftware.SyntaxEditor.Wpf.dll
- \RoleTailored Client\Add-ins\NavgateEditor\ActiproSoftware.Text.Wpf.dll
- \RoleTailored Client\Add-ins\NavgateEditor\WickIt.Licensing.dll
- \RoleTailored Client\Add-ins\NavgateEditor\WickIt.Navgate.CALSyntaxLanguage.dll
- \RoleTailored Client\Add-ins\NavgateEditor\WickIt.Navgate.Editor.dll
- \RoleTailored Client\Add-ins\NavgateEditor\WickIt.Navgate.Parsing.dll

Finally you pick the necessary add-in from the **Add-ins** folder and copy it to the **\Add-ins\NavgateEditor** folder of your **Microsoft Dynamics NAV Client**.



Naam	Gewijzigd op	Type
WickIt.Navgate.NavAddin70.dll	10/28/2014 12:00 ...	Toepassingsu
WickIt.Navgate.NavAddin71.dll	10/28/2014 12:00 ...	Toepassingsu
WickIt.Navgate.NavAddin80.dll	10/28/2014 12:00 ...	Toepassingsu

WickIt.Navgate.NavAddin80.dll Gewijzigd op: 10/28/2014 12:00 PM
Toepassingsuitbreiding Grootte: 65.0 kB

Figure 2-14: Copy NAV Add-in

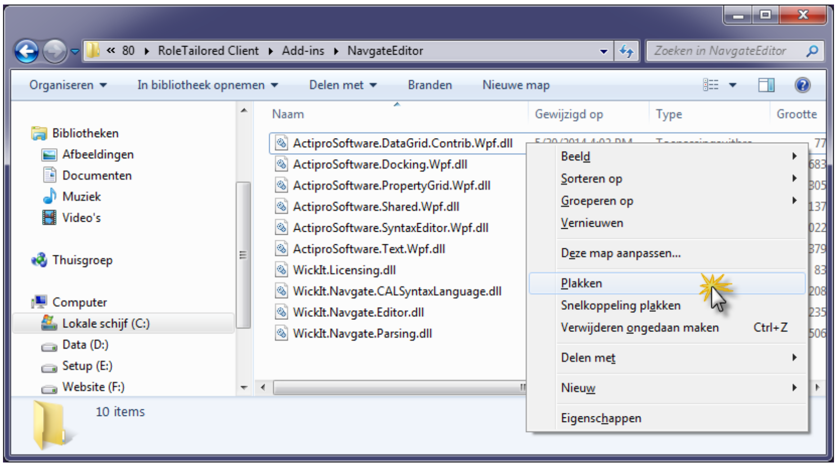


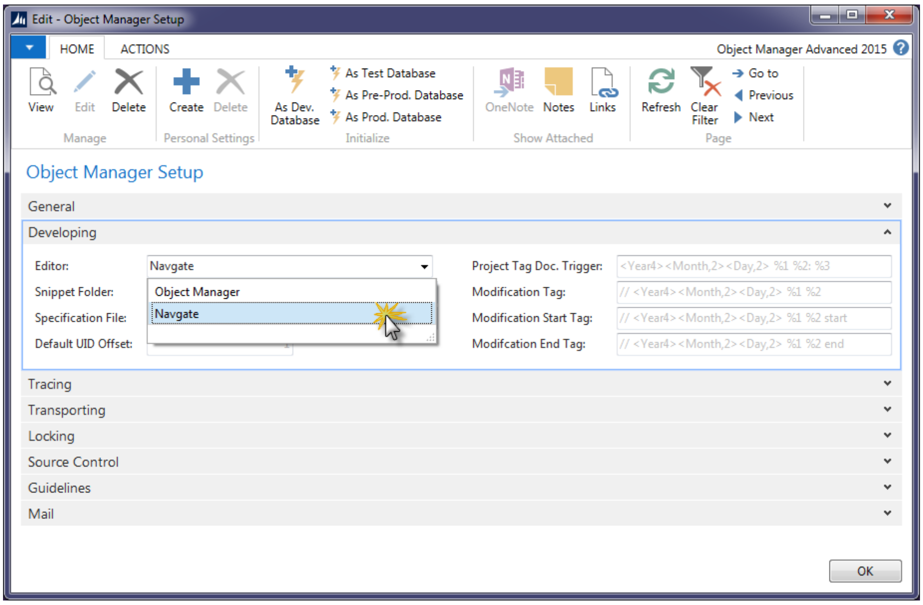
Figure 2-15: Paste NAV Add-in

2.3 ACTIVATING LICENSE KEY

This version of Navgate is shipped freely with Object Manager. Activating is therefore not needed.

3 OBJECT MANAGER CONFIGURATION

The **Object Manager** uses by default its own editor to modify objects. To use the **Navgate** editor choose **Navgate** as editor in the **Object Manager Setup**.



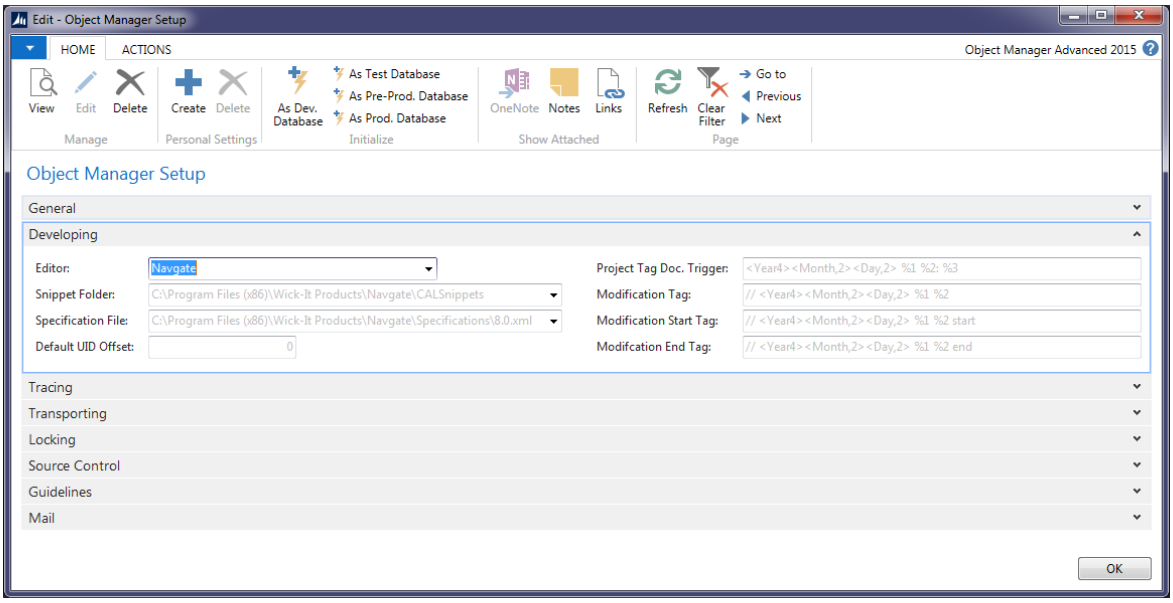
Editor: Navgate
- Object Manager
- Navgate

Snippet Folder:
Specification File:
Default UID Offset:

Project Tag Doc. Trigger: <Year4><Month,2><Day,2> %1 %2: %3
Modification Tag: // <Year4><Month,2><Day,2> %1 %2
Modification Start Tag: // <Year4><Month,2><Day,2> %1 %2 start
Modifcation End Tag: // <Year4><Month,2><Day,2> %1 %2 end

Figure 3-1: Choose Navgate in Object Manager Setup

If the active editor is **Navgate** all other settings on this fast tab become available as well:



Editor: Navgate
Snippet Folder: C:\Program Files (x86)\Wick-It Products\Navgate\CALSnippets
Specification File: C:\Program Files (x86)\Wick-It Products\Navgate\Specifications\8.0.xml
Default UID Offset: 0

Project Tag Doc. Trigger: <Year4><Month,2><Day,2> %1 %2: %3
Modification Tag: // <Year4><Month,2><Day,2> %1 %2
Modification Start Tag: // <Year4><Month,2><Day,2> %1 %2 start
Modifcation End Tag: // <Year4><Month,2><Day,2> %1 %2 end

Figure 3-2: Object Manager Setup

Snippet Folder: This is the folder where **Navgate** will look for code snippets. For more information about code snippets see the chapter ***Code Snippets***. If **Navgate** is installed with the installer this folder will be read from your register. If you installed **Navgate** by copying files to your **Dynamics NAV Client** directory you will have to enter this setting manually.

Specification File: This is the file where the definition of the **NAV** objects is specified. Every version of **NAV** has different functions so for each **NAV** version there is a separate specification file. Same as for the **Snippet Folder** this setting is automatically filled in if you installed **Navgate** with the installer. Otherwise you will have to set this field manually.

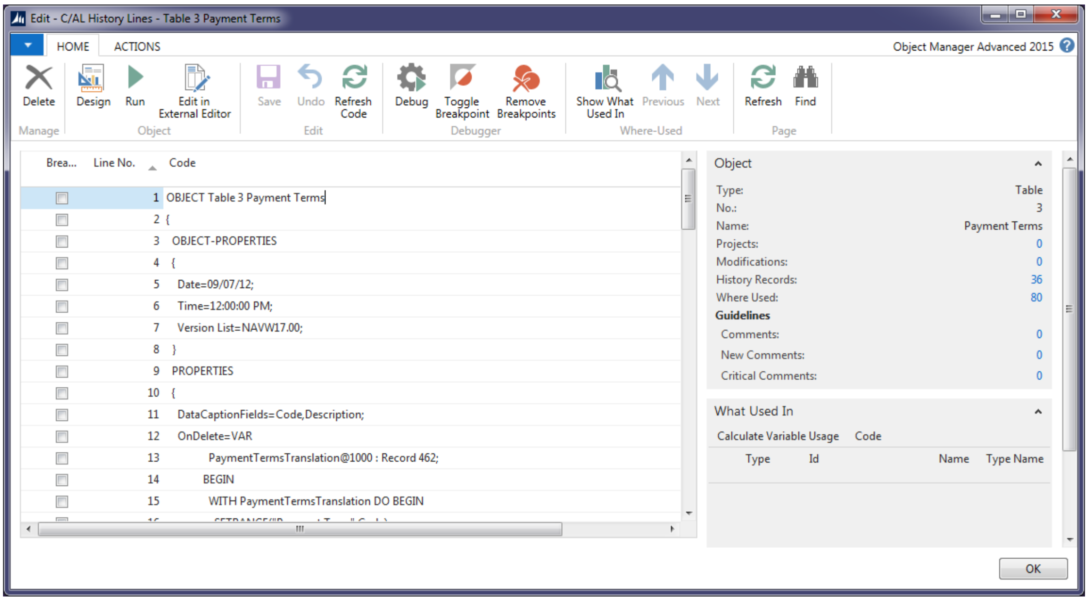
Default UID Offset: The **Default UID Offset** is used to give new elements in your objects an ID. This settings is read from your database. You can overrule it in the **Object Manager Setup** or in the **Project Type**. For more information about the UID Offset see the chapter ***Add Elements***.

Tag Fields: These settings are used for commenting your code. For more information about commenting your code see the chapter ***Commenting***.

4 BASIC FEATURES

4.1 OPEN OBJECTS

There are a lot of places in the **Object Manager** where you can go to the source code of an object. By default the **Object Manager** uses its own **C/AL** code editor which is not very convenient for reading and editing objects. It misses features like **Syntax Coloring** and **IntelliSense**. The following picture shows what it looks like.



```
OBJECT Table 3 Payment Terms
{
 OBJECT-PROPERTIES
 {
  Date=09/07/12;
  Time=12:00:00 PM;
  Version List=NAVW17.00;
 }
 PROPERTIES
 {
  DataCaptionFields=Code,Description;
  OnDelete=VAR
         PaymentTermsTranslation@1000 : Record 462;
        BEGIN
         WITH PaymentTermsTranslation DO BEGIN
```

Object

Type:	Table
No.:	3
Name:	Payment Terms
Projects:	0
Modifications:	0
History Records:	36
Where Used:	80
Guidelines	
Comments:	0
New Comments:	0
Critical Comments:	0

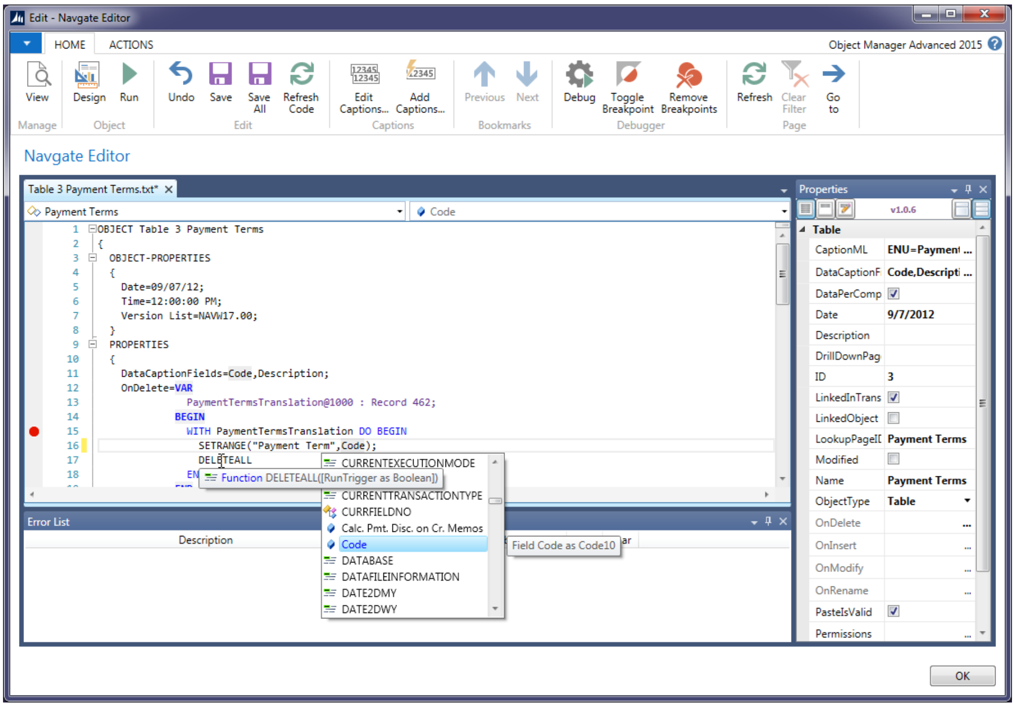
What Used In

Calculate Variable Usage Code

Type	Id	Name	Type Name

Figure 4-1: Object Manager's Code Editor

Choose **Navgate** in the **Object Manager Setup** as editor to show and edit the **C/AL** code of your objects.



```
OBJECT Table 3 Payment Terms
{
  OBJECT-PROPERTIES
  {
    Date=09/07/12;
    Time=12:00:00 PM;
    Version List=NAVW17.00;
  }
  PROPERTIES
  {
    DataCaptionFields=Code,Description;
    OnDelete=VAR
               PaymentTermsTranslation@1000 : Record 462;
             BEGIN
               WITH PaymentTermsTranslation DO BEGIN
                 SETRANGE("Payment Term",Code);
                 DELETEALL
               EN
```

Function DELETEALL([RunTrigger as Boolean])

- CURRENTEXECUTIONMODE
- CURRENTTRANSACTIONTYPE
- CURRFIELDNO
- Calc. Pmt. Disc. on Cr. Memos
- Code
- DATABASE
- DATAFILEINFORMATION
- DATE2DMY
- DATE2DWY

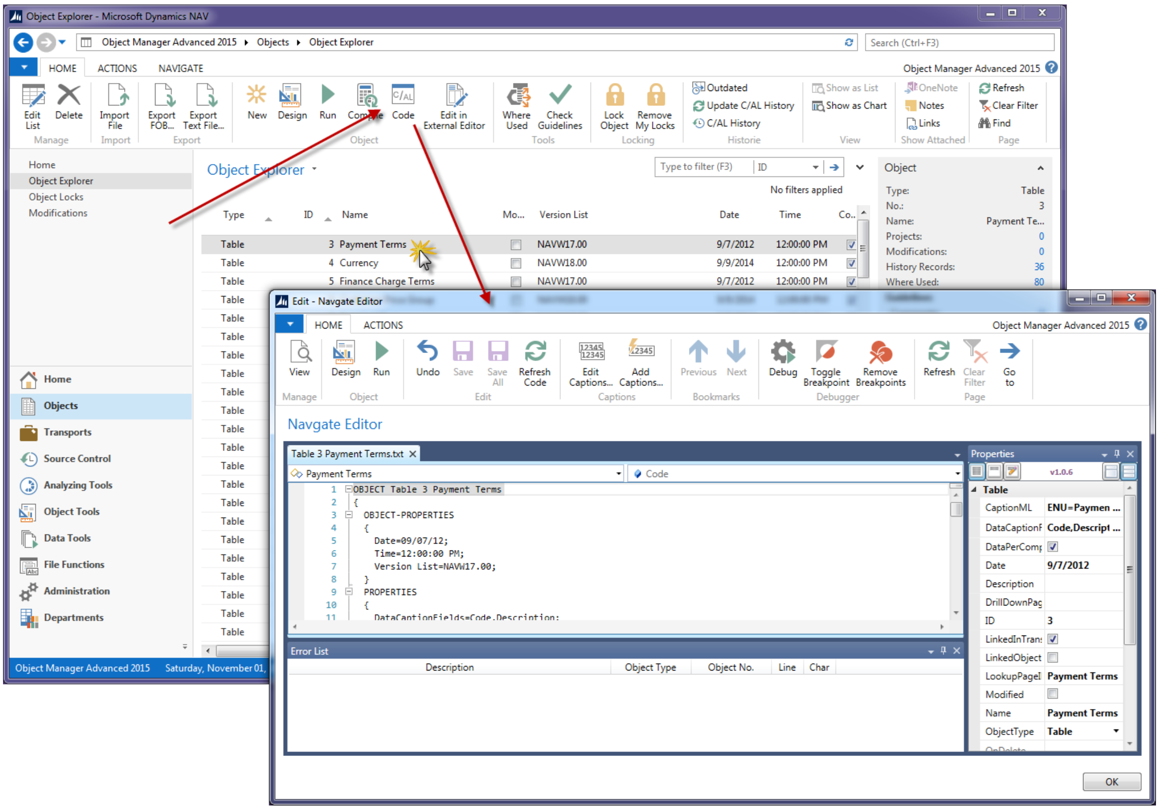
Field Code as Code10

Table

CaptionML	ENU=Paymen ...
DataCaptionF	Code,Descripti ...
DataPerComp	☑
Date	9/7/2012
Description	
DrillDownPag	
ID	3
LinkedInTrans	☑
LinkedObject	☐
LookupPageI[	Payment Terms
Modified	☐
Name	Payment Terms
ObjectType	Table
OnDelete	...
OnInsert	...
OnModify	...
OnRename	...
PasteIsValid	☑
Permissions	...

Figure 4-2: Navgate

Navgate can be opened wherever you see the following **Code** button on the ribbon.



Type	ID	Name	Version List	Date	Time
Table	3	Payment Terms	NAVW17.00	9/7/2012	12:00:00 PM
Table	4	Currency	NAVW18.00	9/9/2014	12:00:00 PM
Table	5	Finance Charge Terms	NAVW17.00	9/7/2012	12:00:00 PM

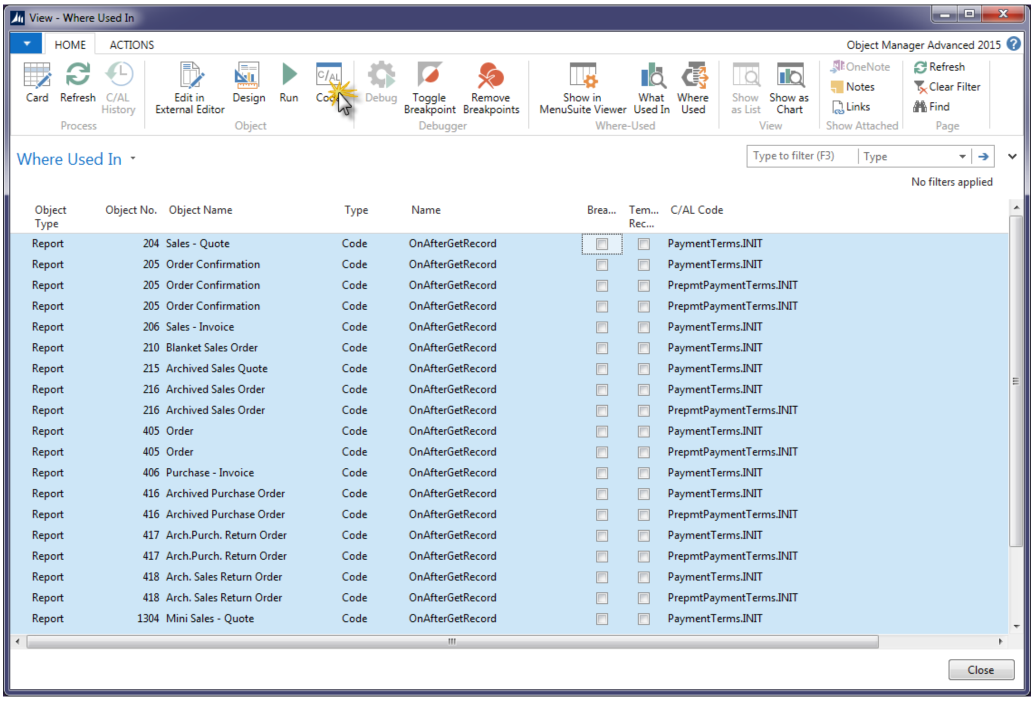
Object
Type: Table
No.: 3
Name: Payment Te...
Projects: 0
Modifications: 0
History Records: 36
Where Used: 80

```
OBJECT Table 3 Payment Terms
{
  OBJECT-PROPERTIES
  {
    Date=09/07/12;
    Time=12:00:00 PM;
    Version List=NAVW17.00;
  }
  PROPERTIES
  {
    DataCaptionFields=Code,Description;
```

Properties v1.0.6
Table
CaptionML: ENU=Paymen ...
DataCaptionF: Code,Descript ...
DataPerComp: ☑
Date: 9/7/2012
Description:
DrillDownPag:
ID: 3
LinkedInTran: ☑
LinkedObject: ☐
LookupPageI: Payment Terms
Modified: ☐
Name: Payment Terms
ObjectType: Table

Figure 4-3: Open Navgate with the Code Button

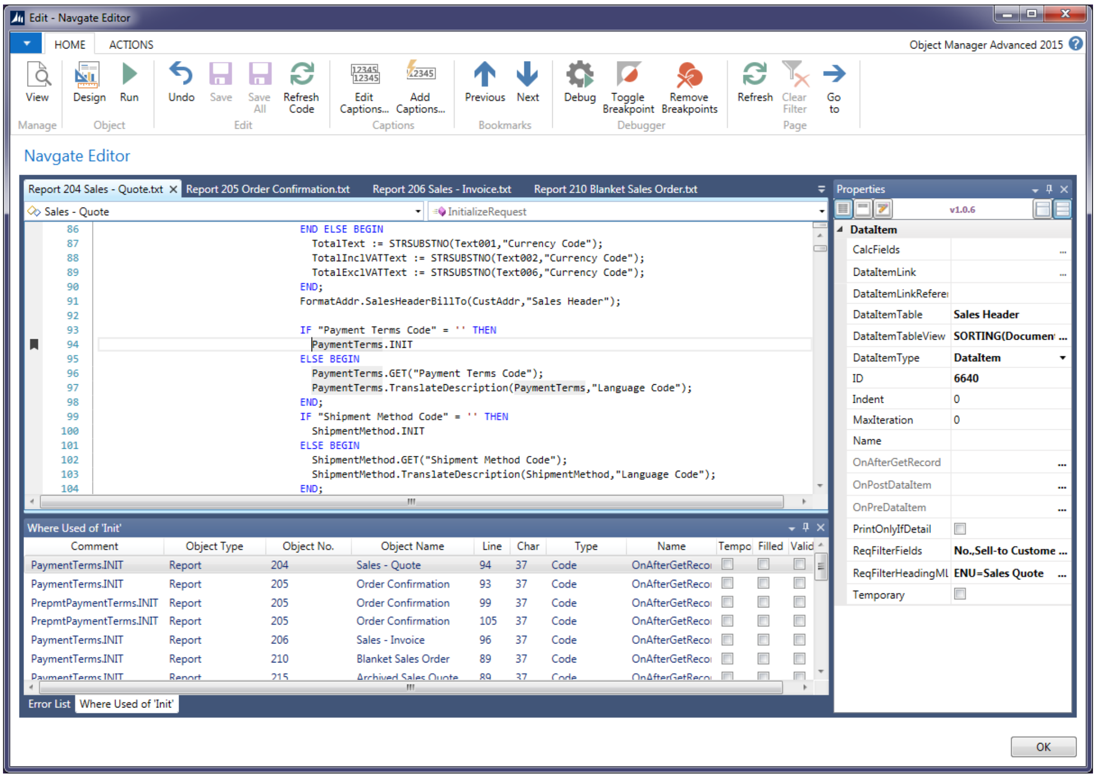
In the **Where Used** and **Check Guidelines** tool, **Navgate** can be opened with more than one object at once. For example, to see where the **PaymentTable.INIT** is called select all entries in the **Where Used** tool and click the **Code** button.



Object Type	Object No.	Object Name	Type	Name	Brea...	Tem... Rec...	C/AL Code
Report	204	Sales - Quote	Code	OnAfterGetRecord	☐	☐	PaymentTerms.INIT
Report	205	Order Confirmation	Code	OnAfterGetRecord	☐	☐	PaymentTerms.INIT
Report	205	Order Confirmation	Code	OnAfterGetRecord	☐	☐	PrepmtPaymentTerms.INIT
Report	205	Order Confirmation	Code	OnAfterGetRecord	☐	☐	PrepmtPaymentTerms.INIT
Report	206	Sales - Invoice	Code	OnAfterGetRecord	☐	☐	PaymentTerms.INIT
Report	210	Blanket Sales Order	Code	OnAfterGetRecord	☐	☐	PaymentTerms.INIT
Report	215	Archived Sales Quote	Code	OnAfterGetRecord	☐	☐	PaymentTerms.INIT
Report	216	Archived Sales Order	Code	OnAfterGetRecord	☐	☐	PaymentTerms.INIT
Report	216	Archived Sales Order	Code	OnAfterGetRecord	☐	☐	PrepmtPaymentTerms.INIT
Report	405	Order	Code	OnAfterGetRecord	☐	☐	PaymentTerms.INIT
Report	405	Order	Code	OnAfterGetRecord	☐	☐	PrepmtPaymentTerms.INIT
Report	406	Purchase - Invoice	Code	OnAfterGetRecord	☐	☐	PaymentTerms.INIT
Report	416	Archived Purchase Order	Code	OnAfterGetRecord	☐	☐	PaymentTerms.INIT
Report	416	Archived Purchase Order	Code	OnAfterGetRecord	☐	☐	PrepmtPaymentTerms.INIT
Report	417	Arch.Purch. Return Order	Code	OnAfterGetRecord	☐	☐	PaymentTerms.INIT
Report	417	Arch.Purch. Return Order	Code	OnAfterGetRecord	☐	☐	PrepmtPaymentTerms.INIT
Report	418	Arch. Sales Return Order	Code	OnAfterGetRecord	☐	☐	PaymentTerms.INIT
Report	418	Arch. Sales Return Order	Code	OnAfterGetRecord	☐	☐	PrepmtPaymentTerms.INIT
Report	1304	Mini Sales - Quote	Code	OnAfterGetRecord	☐	☐	PaymentTerms.INIT

Figure 4-4: Selecting Where Used Entries

All selected objects will be opened in **Navgate**. Every object in its own tab.



```
END ELSE BEGIN
  TotalText := STRSUBSTNO(Text001,"Currency Code");
  TotalInclVATText := STRSUBSTNO(Text002,"Currency Code");
  TotalExclVATText := STRSUBSTNO(Text006,"Currency Code");
END;
FormatAddr.SalesHeaderBillTo(CustAddr,"Sales Header");

IF "Payment Terms Code" = '' THEN
  PaymentTerms.INIT
ELSE BEGIN
  PaymentTerms.GET("Payment Terms Code");
  PaymentTerms.TranslateDescription(PaymentTerms,"Language Code");
END;
IF "Shipment Method Code" = '' THEN
  ShipmentMethod.INIT
ELSE BEGIN
  ShipmentMethod.GET("Shipment Method Code");
  ShipmentMethod.TranslateDescription(ShipmentMethod,"Language Code");
END;
```

Where Used of 'Init'

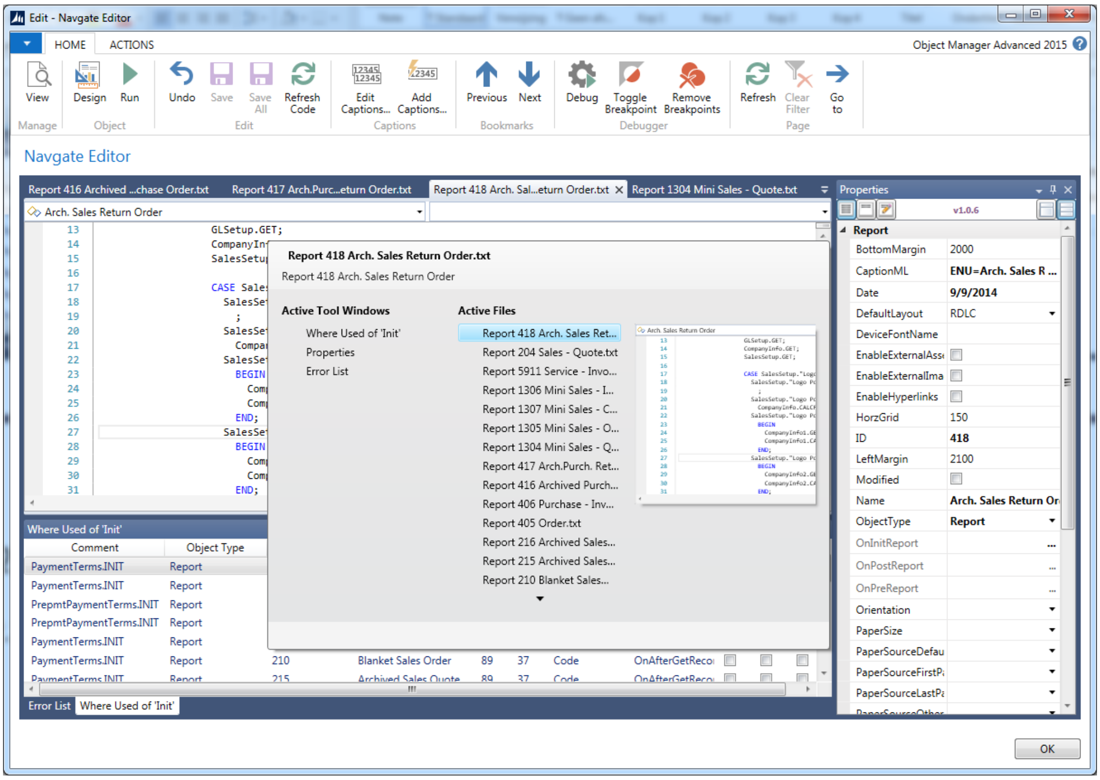
Comment	Object Type	Object No.	Object Name	Line	Char	Type	Name	Tempo	Filled	Valid
PaymentTerms.INIT	Report	204	Sales - Quote	94	37	Code	OnAfterGetReco	☐	☐	☐
PaymentTerms.INIT	Report	205	Order Confirmation	93	37	Code	OnAfterGetReco	☐	☐	☐
PrepmtPaymentTerms.INIT	Report	205	Order Confirmation	99	37	Code	OnAfterGetReco	☐	☐	☐
PrepmtPaymentTerms.INIT	Report	205	Order Confirmation	105	37	Code	OnAfterGetReco	☐	☐	☐
PaymentTerms.INIT	Report	206	Sales - Invoice	96	37	Code	OnAfterGetReco	☐	☐	☐
PaymentTerms.INIT	Report	210	Blanket Sales Order	89	37	Code	OnAfterGetReco	☐	☐	☐
PaymentTerms.INIT	Report	215	Archived Sales Quote	89	37	Code	OnAfterGetReco	☐	☐	☐

Properties – DataItem

CalcFields	...
DataItemLink	...
DataItemLinkRefere	
DataItemTable	Sales Header
DataItemTableView	SORTING(Documen ...
DataItemType	DataItem
ID	6640
Indent	0
MaxIteration	0
Name	
OnAfterGetRecord	...
OnPostDataItem	...
OnPreDataItem	...
PrintOnlyIfDetail	☐
ReqFilterFields	No.,Sell-to Custome ...
ReqFilterHeadingML	ENU=Sales Quote ...
Temporary	☐

Figure 4-5: Multiple Objects Opened in Navgate

With **Ctrl+TAB** you can switch between the different objects:



```
GLSetup.GET;
CompanyIn
SalesSetu

CASE Sale
  SalesSe
    ;
  SalesSe
    Compa
  SalesSe
    BEGIN
      Com
      Com
    END;
  SalesSe
    BEGIN
      Com
      Com
    END;
```

Report 418 Arch. Sales Return Order.txt

Report 418 Arch. Sales Return Order

Active Tool Windows

- Where Used of 'Init'
- Properties
- Error List

Active Files

- Report 418 Arch. Sales Ret...
- Report 204 Sales - Quote.txt
- Report 5911 Service - Invo...
- Report 1306 Mini Sales - I...
- Report 1307 Mini Sales - C...
- Report 1305 Mini Sales - O...
- Report 1304 Mini Sales - Q...
- Report 417 Arch.Purch. Ret...
- Report 416 Archived Purch...
- Report 406 Purchase - Inv...
- Report 405 Order.txt
- Report 216 Archived Sales...
- Report 215 Archived Sales...
- Report 210 Blanket Sales...

Where Used of 'Init'

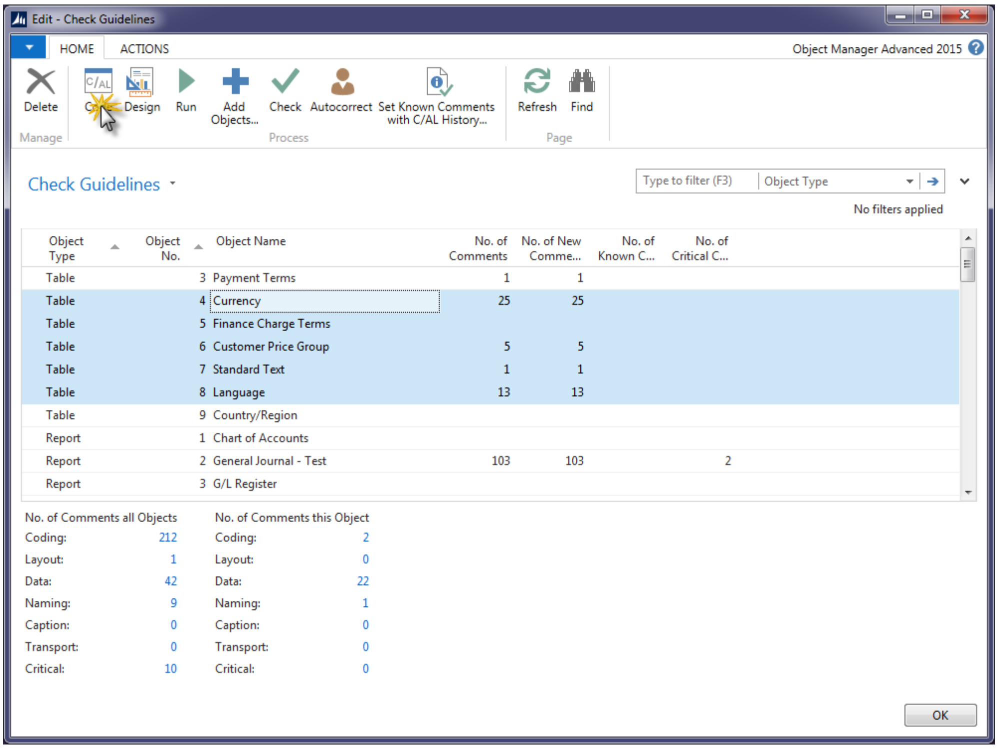
Comment	Object Type									
PaymentTerms.INIT	Report									
PaymentTerms.INIT	Report									
PrepmtPaymentTerms.INIT	Report									
PrepmtPaymentTerms.INIT	Report									
PaymentTerms.INIT	Report									
PaymentTerms.INIT	Report	210	Blanket Sales Order	89	37	Code	OnAfterGetReco	☐	☐	☐
PaymentTerms.INIT	Report	215	Archived Sales Quote	89	37	Code	OnAfterGetReco	☐	☐	☐

Properties – Report

BottomMargin	2000
CaptionML	ENU=Arch. Sales R ...
Date	9/9/2014
DefaultLayout	RDLC
DeviceFontName	
EnableExternalAss	☐
EnableExternalIma	☐
EnableHyperlinks	☐
HorzGrid	150
ID	418
LeftMargin	2100
Modified	☐
Name	Arch. Sales Return Or
ObjectType	Report
OnInitReport	...
OnPostReport	...
OnPreReport	...
Orientation	
PaperSize	
PaperSourceDefau	
PaperSourceFirstP	
PaperSourceLastPa	

Figure 4-6: Use Ctrl+TAB to Switch Objects

The same applies for the **Check Guidelines** Tool, you can select multiple objects and open them together at once in **Navgate**.



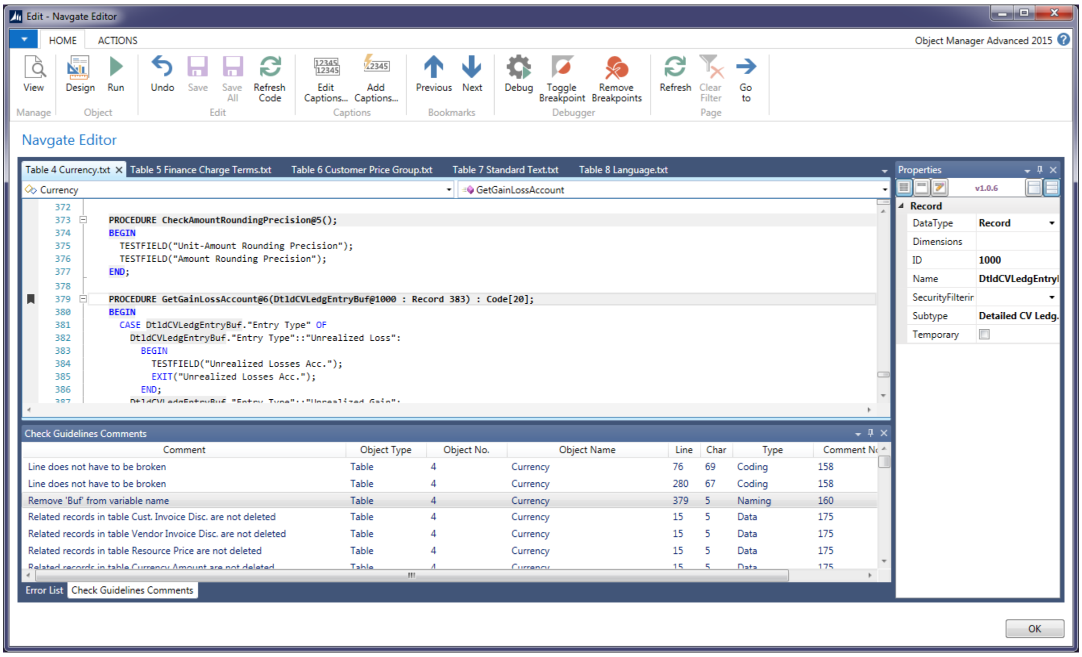
Check Guidelines

No filters applied

Object Type	Object No.	Object Name	No. of Comments	No. of New Comme...	No. of Known C...	No. of Critical C...
Table	3	Payment Terms	1	1		
Table	4	Currency	25	25		
Table	5	Finance Charge Terms				
Table	6	Customer Price Group	5	5		
Table	7	Standard Text	1	1		
Table	8	Language	13	13		
Table	9	Country/Region				
Report	1	Chart of Accounts				
Report	2	General Journal - Test	103	103		2
Report	3	G/L Register				

No. of Comments all Objects		No. of Comments this Object	
Coding:	212	Coding:	2
Layout:	1	Layout:	0
Data:	42	Data:	22
Naming:	9	Naming:	1
Caption:	0	Caption:	0
Transport:	0	Transport:	0
Critical:	10	Critical:	0

Figure 4-7: Check Guidelines Tool



Navgate Editor

```
    PROCEDURE CheckAmountRoundingPrecision@5();
    BEGIN
      TESTFIELD("Unit-Amount Rounding Precision");
      TESTFIELD("Amount Rounding Precision");
    END;

    PROCEDURE GetGainLossAccount@6(DtldCVLedgEntryBuf@1000 : Record 383) : Code[20];
    BEGIN
      CASE DtldCVLedgEntryBuf."Entry Type" OF
        DtldCVLedgEntryBuf."Entry Type"::"Unrealized Loss":
          BEGIN
            TESTFIELD("Unrealized Losses Acc.");
            EXIT("Unrealized Losses Acc.");
          END;
```

Record

DataType	Record
Dimensions	
ID	1000
Name	DtldCVLedgEntryI
SecurityFilterin	
Subtype	Detailed CV Ledg.
Temporary	☐

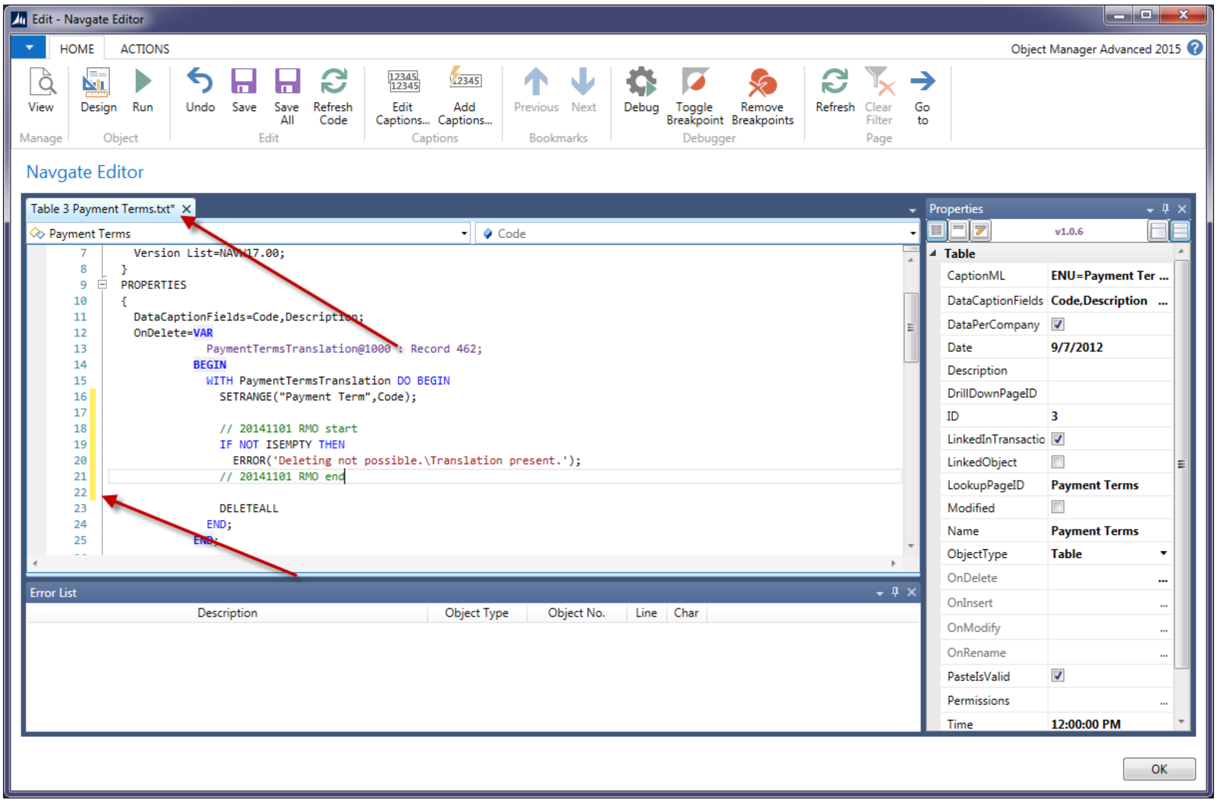
Check Guidelines Comments

Comment	Object Type	Object No.	Object Name	Line	Char	Type	Comment N
Line does not have to be broken	Table	4	Currency	76	69	Coding	158
Line does not have to be broken	Table	4	Currency	280	67	Coding	158
Remove 'Buf' from variable name	Table	4	Currency	379	5	Naming	160
Related records in table Cust. Invoice Disc. are not deleted	Table	4	Currency	15	5	Data	175
Related records in table Vendor Invoice Disc. are not deleted	Table	4	Currency	15	5	Data	175
Related records in table Resource Price are not deleted	Table	4	Currency	15	5	Data	175

Figure 4-8: Multiple Objects Opened in Navgate

4.2 SAVE OBJECTS

When you are editing your **C/AL** code, it is useful to understand which lines of code have been saved to the database and which have not. **Change Tracking** provides this functionality. A **yellow** indicator in the selection margin tells you that the changes you made are not saved to the database. Besides that an **asterisk** will be added to the title of the tab.

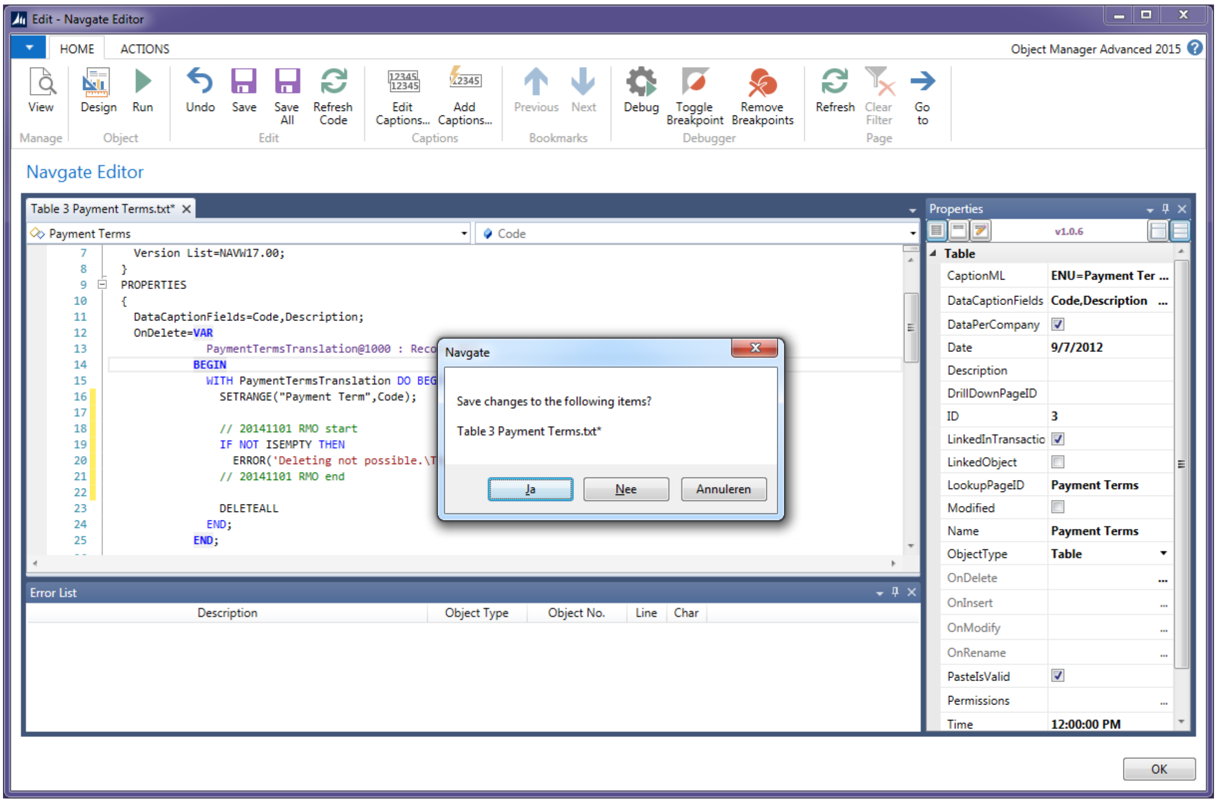


```
   Version List=NAVW17.00;
 }
 PROPERTIES
 {
   DataCaptionFields=Code,Description;
   OnDelete=VAR
              PaymentTermsTranslation@1000 : Record 462;
            BEGIN
              WITH PaymentTermsTranslation DO BEGIN
                SETRANGE("Payment Term",Code);

                // 20141101 RMO start
                IF NOT ISEMPTY THEN
                  ERROR('Deleting not possible.\Translation present.');
                // 20141101 RMO end

                DELETEALL
              END;
            END;
```

Table	
CaptionML	ENU=Payment Ter ...
DataCaptionFields	Code,Description ...
DataPerCompany	☑
Date	9/7/2012
Description	
DrillDownPageID	
ID	3
LinkedInTransactio	☑
LinkedObject	☐
LookupPageID	Payment Terms
Modified	☐
Name	Payment Terms
ObjectType	Table
OnDelete	...
OnInsert	...
OnModify	...
OnRename	...
PasteIsValid	☑
Permissions	...
Time	12:00:00 PM

Figure 4-9: Modified Object

If your object has unsaved changes you will be warned if you try to close **Navgate**.



```
   Version List=NAVW17.00;
 }
 PROPERTIES
 {
   DataCaptionFields=Code,Description;
   OnDelete=VAR
              PaymentTermsTranslation@1000 : Reco
            BEGIN
              WITH PaymentTermsTranslation DO BEG
                SETRANGE("Payment Term",Code);

                // 20141101 RMO start
                IF NOT ISEMPTY THEN
                  ERROR('Deleting not possible.\T
                // 20141101 RMO end

                DELETEALL
              END;
            END;
```

Navgate

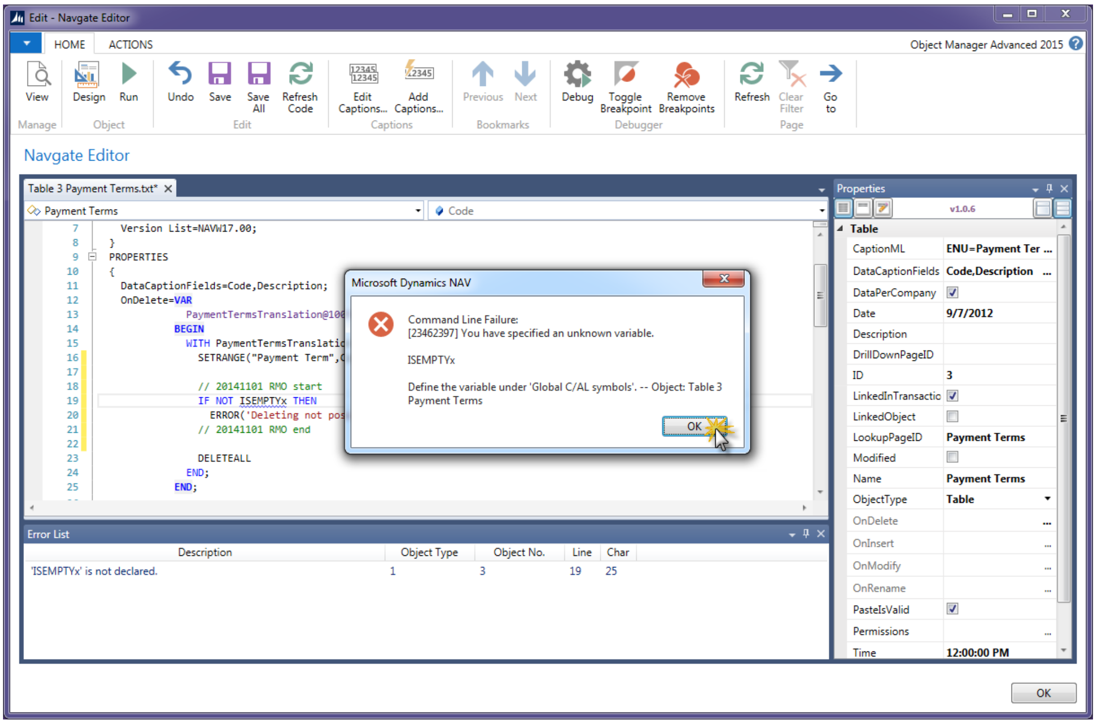
Save changes to the following items?

Table 3 Payment Terms.txt*

Table	
CaptionML	ENU=Payment Ter ...
DataCaptionFields	Code,Description ...
DataPerCompany	☑
Date	9/7/2012
Description	
DrillDownPageID	
ID	3
LinkedInTransactio	☑
LinkedObject	☐
LookupPageID	Payment Terms
Modified	☐
Name	Payment Terms
ObjectType	Table
OnDelete	...
OnInsert	...
OnModify	...
OnRename	...
PasteIsValid	☑
Permissions	...
Time	12:00:00 PM

Figure 4-10: Confirming That You Want to Discard Changes

The changes you make can be saved to the database in two ways. The save buttons will import and compile the object. It there is an error in your object the **Command Line**, which is responsible for compiling the object, will result with an error.



```
    Version List=NAVW17.00;
  }
  PROPERTIES
  {
    DataCaptionFields=Code,Description;
    OnDelete=VAR
               PaymentTermsTranslation@100
             BEGIN
               WITH PaymentTermsTranslatic
                 SETRANGE("Payment Term",C

                 // 20141101 RMO start
                 IF NOT ISEMPTYx THEN
                   ERROR('Deleting not pos
                 // 20141101 RMO end

                 DELETEALL
               END;
             END;
```

Command Line Failure:
[23462397] You have specified an unknown variable.

ISEMPTYx

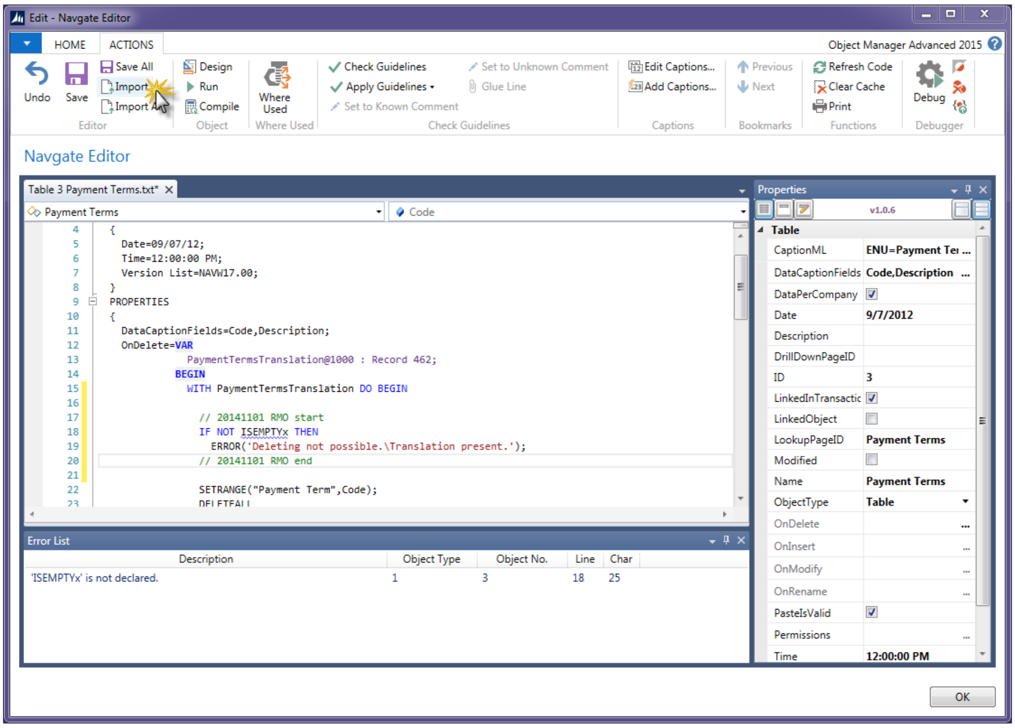
Define the variable under 'Global C/AL symbols'. -- Object: Table 3 Payment Terms

Description	Object Type	Object No.	Line	Char
'ISEMPTYx' is not declared.	1	3	19	25

Table	
CaptionML	ENU=Payment Ter ...
DataCaptionFields	Code,Description ...
DataPerCompany	☑
Date	9/7/2012
Description	
DrillDownPageID	
ID	3
LinkedInTransactic	☑
LinkedObject	☐
LookupPageID	Payment Terms
Modified	☐
Name	Payment Terms
ObjectType	Table
OnDelete	...
OnInsert	...
OnModify	...
OnRename	...
PasteIsValid	☑
Permissions	...
Time	12:00:00 PM

Figure 4-11: Error from the Command Line

To ignore this error and save the object to the database anyway, it is possible to use the **Import** buttons. This function will import your object without compiling it.



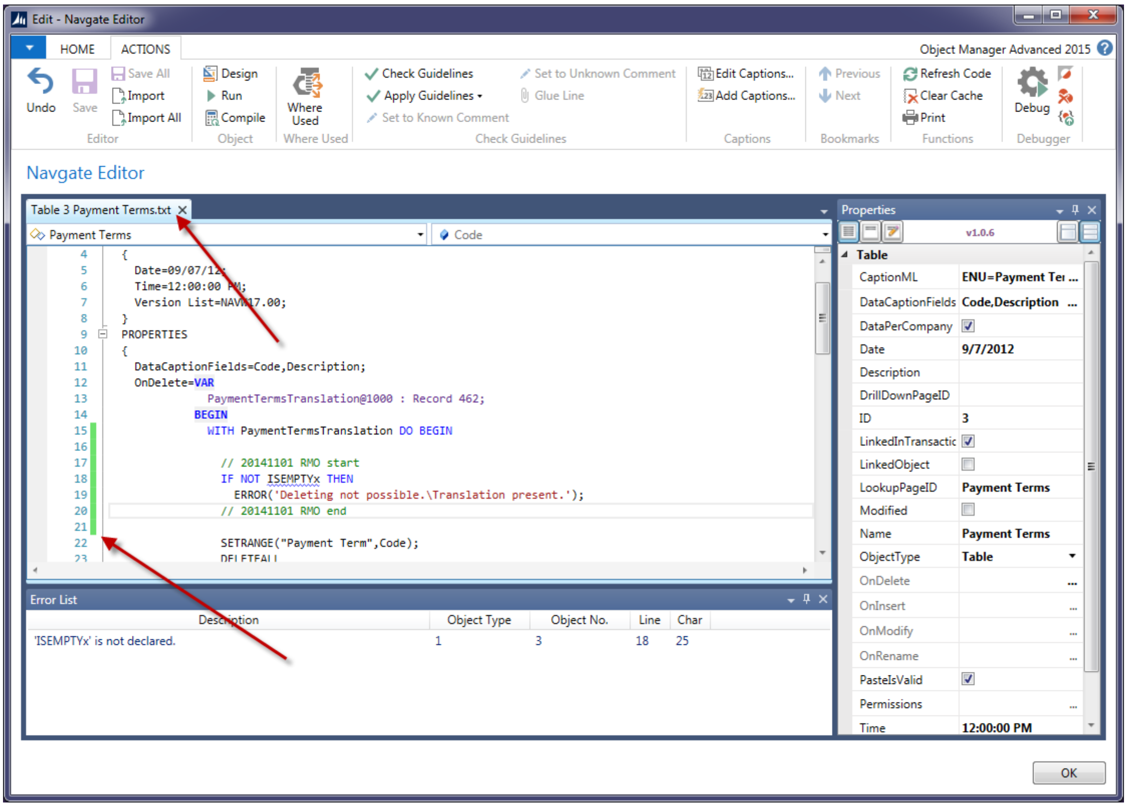
```
  {
    Date=09/07/12;
    Time=12:00:00 PM;
    Version List=NAVW17.00;
  }
  PROPERTIES
  {
    DataCaptionFields=Code,Description;
    OnDelete=VAR
               PaymentTermsTranslation@1000 : Record 462;
             BEGIN
               WITH PaymentTermsTranslation DO BEGIN

                 // 20141101 RMO start
                 IF NOT ISEMPTYx THEN
                   ERROR('Deleting not possible.\Translation present.');
                 // 20141101 RMO end

                 SETRANGE("Payment Term",Code);
                 DELETEALL
```

Description	Object Type	Object No.	Line	Char
'ISEMPTYx' is not declared.	1	3	18	25

Table	
CaptionML	ENU=Payment Ter ...
DataCaptionFields	Code,Description ...
DataPerCompany	☑
Date	9/7/2012
Description	
DrillDownPageID	
ID	3
LinkedInTransactic	☑
LinkedObject	☐
LookupPageID	Payment Terms
Modified	☐
Name	Payment Terms
ObjectType	Table
OnDelete	...
OnInsert	...
OnModify	...
OnRename	...
PasteIsValid	☑
Permissions	...
Time	12:00:00 PM

Figure 4-12: Import an Object without Compiling

After saving your object to the database the **asterisk** will be removed from the title and the **yellow** indicator in the selection margin will turn **green**.



```
{
  Date=09/07/12;
  Time=12:00:00 PM;
  Version List=NAVW17.00;
}
PROPERTIES
{
  DataCaptionFields=Code,Description;
  OnDelete=VAR
             PaymentTermsTranslation@1000 : Record 462;
           BEGIN
             WITH PaymentTermsTranslation DO BEGIN

               // 20141101 RMO start
               IF NOT ISEMPTYx THEN
                 ERROR('Deleting not possible.\Translation present.');
               // 20141101 RMO end

               SETRANGE("Payment Term",Code);
               DELETEALL
```

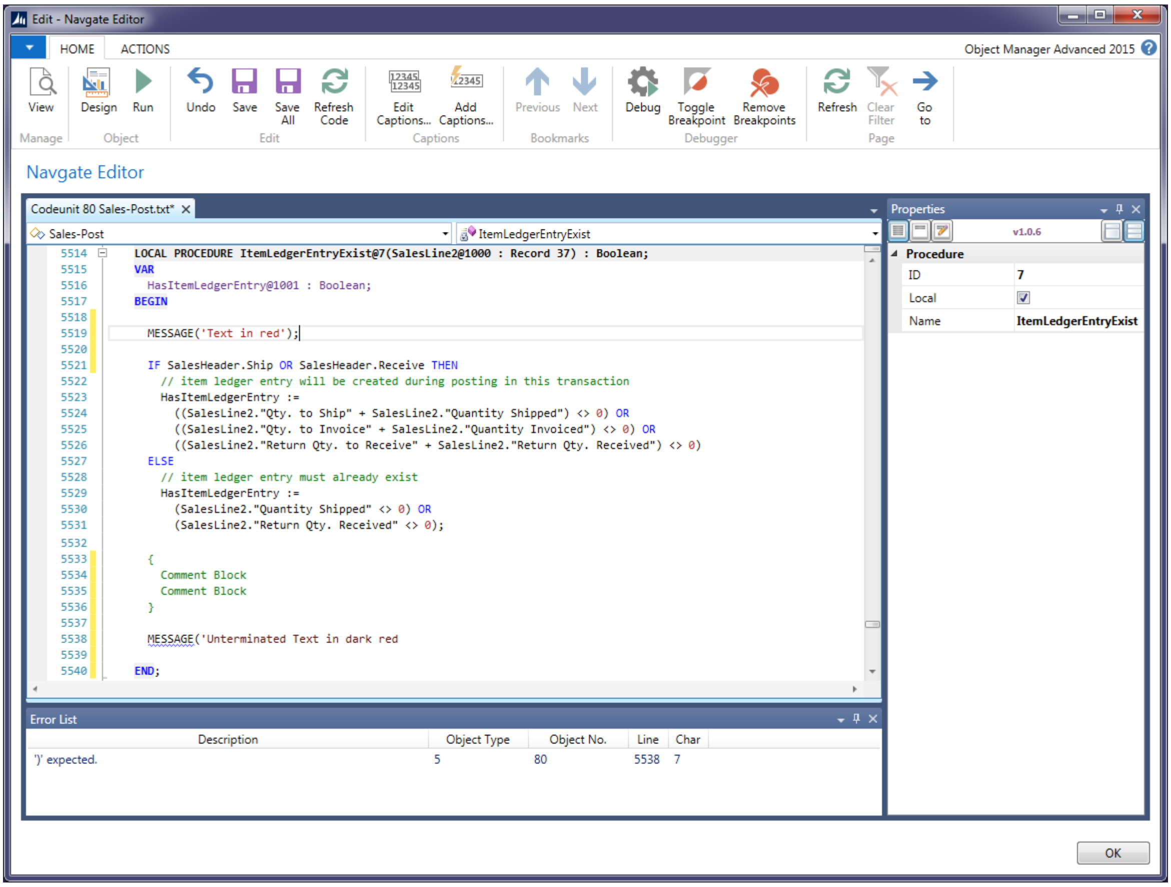
Description	Object Type	Object No.	Line	Char
'ISEMPTYx' is not declared.	1	3	18	25

Table	
CaptionML	ENU=Payment Tei ...
DataCaptionFields	Code,Description ...
DataPerCompany	☑
Date	9/7/2012
Description	
DrillDownPageID	
ID	3
LinkedInTransactic	☑
LinkedObject	☐
LookupPageID	Payment Terms
Modified	☐
Name	Payment Terms
ObjectType	Table
OnDelete	...
OnInsert	...
OnModify	...
OnRename	...
PasteIsValid	☑
Permissions	...
Time	12:00:00 PM

Figure 4-13: Modifications Committed to the Database

5 SYNTAX COLORING

Navgate can parse and color different code parts to make it much easier to identify on sight. As an example **Navgate** colors all comments green. Code identifiers are black, keywords are blue, string and values as date and integers are red and unterminated strings are colored dark red. A procedure header is made bold with a grey background and variables are colored purple.



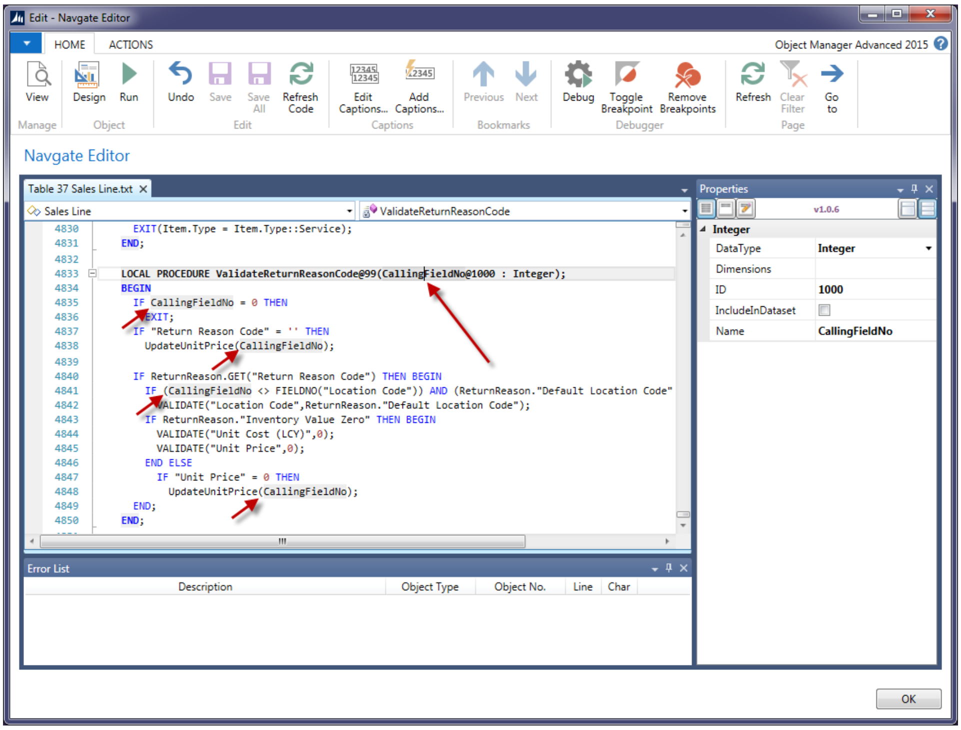
```
LOCAL PROCEDURE ItemLedgerEntryExist@7(SalesLine2@1000 : Record 37) : Boolean;
VAR
  HasItemLedgerEntry@1001 : Boolean;
BEGIN

  MESSAGE('Text in red');

  IF SalesHeader.Ship OR SalesHeader.Receive THEN
    // item ledger entry will be created during posting in this transaction
    HasItemLedgerEntry :=
      ((SalesLine2."Qty. to Ship" + SalesLine2."Quantity Shipped") <> 0) OR
      ((SalesLine2."Qty. to Invoice" + SalesLine2."Quantity Invoiced") <> 0) OR
      ((SalesLine2."Return Qty. to Receive" + SalesLine2."Return Qty. Received") <> 0)
  ELSE
    // item ledger entry must already exist
    HasItemLedgerEntry :=
      (SalesLine2."Quantity Shipped" <> 0) OR
      (SalesLine2."Return Qty. Received" <> 0);

  {
    Comment Block
    Comment Block
  }

  MESSAGE('Unterminated Text in dark red

END;
```

Description	Object Type	Object No.	Line	Char
')' expected.	5	80	5538	7

Procedure
ID 7
Local ☑
Name ItemLedgerEntryExist

Figure 5-1: Syntax Coloring

If you place your cursor on a keyword all instances of that keyword are highlighted. This helps you to see where for example a parameter or variable is used.



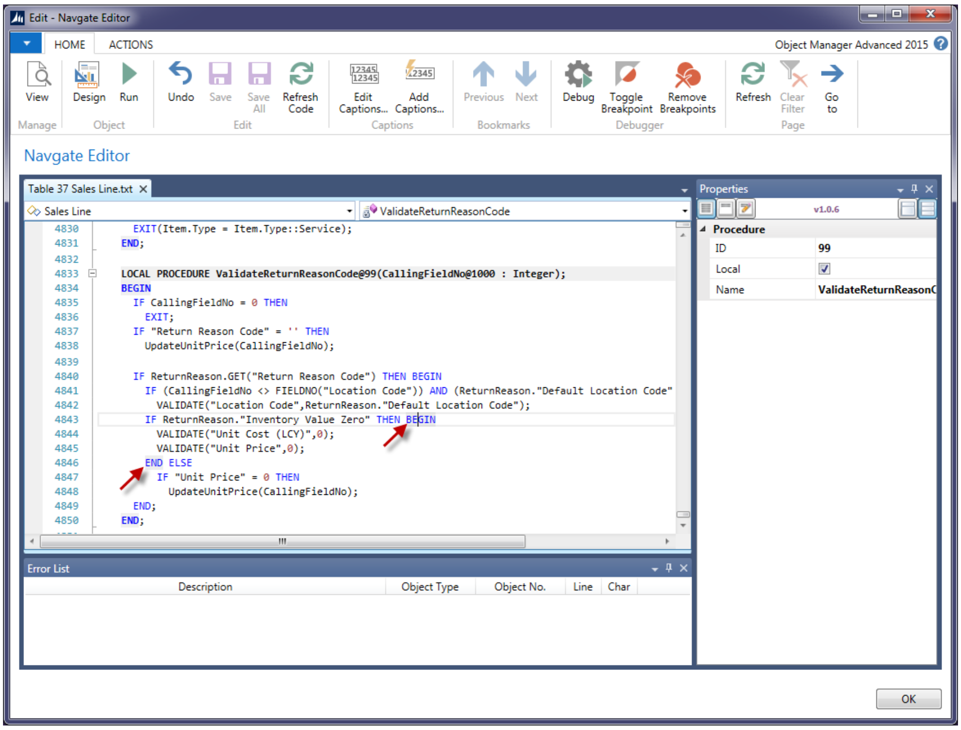
```
    EXIT(Item.Type = Item.Type::Service);
  END;

  LOCAL PROCEDURE ValidateReturnReasonCode@99(CallingFieldNo@1000 : Integer);
  BEGIN
    IF CallingFieldNo = 0 THEN
      EXIT;
    IF "Return Reason Code" = '' THEN
      UpdateUnitPrice(CallingFieldNo);

    IF ReturnReason.GET("Return Reason Code") THEN BEGIN
      IF (CallingFieldNo <> FIELDNO("Location Code")) AND (ReturnReason."Default Location Code"
        VALIDATE("Location Code",ReturnReason."Default Location Code");
      IF ReturnReason."Inventory Value Zero" THEN BEGIN
        VALIDATE("Unit Cost (LCY)",0);
        VALIDATE("Unit Price",0);
      END ELSE
        IF "Unit Price" = 0 THEN
          UpdateUnitPrice(CallingFieldNo);
    END;
  END;
```

Integer

DataType	Integer
Dimensions	
ID	1000
IncludeInDataset	☐
Name	CallingFieldNo

Figure 5-2: Keyword Highlighting

Place your cursor on keywords like **IF**, **ELSE**, **BEGIN**, etc., then the corresponding keywords will also be highlighted.



```
    EXIT(Item.Type = Item.Type::Service);
  END;

  LOCAL PROCEDURE ValidateReturnReasonCode@99(CallingFieldNo@1000 : Integer);
  BEGIN
    IF CallingFieldNo = 0 THEN
      EXIT;
    IF "Return Reason Code" = '' THEN
      UpdateUnitPrice(CallingFieldNo);

    IF ReturnReason.GET("Return Reason Code") THEN BEGIN
      IF (CallingFieldNo <> FIELDNO("Location Code")) AND (ReturnReason."Default Location Code"
        VALIDATE("Location Code",ReturnReason."Default Location Code");
      IF ReturnReason."Inventory Value Zero" THEN BEGIN
        VALIDATE("Unit Cost (LCY)",0);
        VALIDATE("Unit Price",0);
      END ELSE
        IF "Unit Price" = 0 THEN
          UpdateUnitPrice(CallingFieldNo);
    END;
  END;
```

Procedure

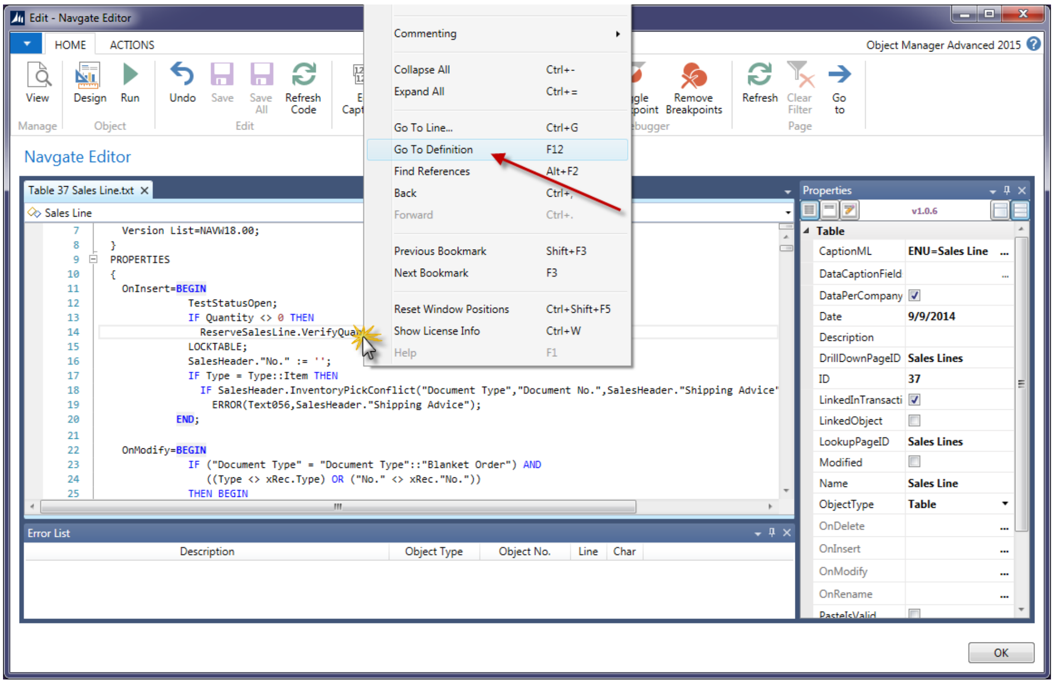
ID	99
Local	☑
Name	ValidateReturnReasonC

Figure 5-3: Highlighted BEGIN and END

6 NAVIGATION

6.1 GO TO DEFINITION

With the **Go To Definition** function (**F12**) you can move to the location where the keyword is defined in the database. If this keyword is in another object a new tab will be opened.

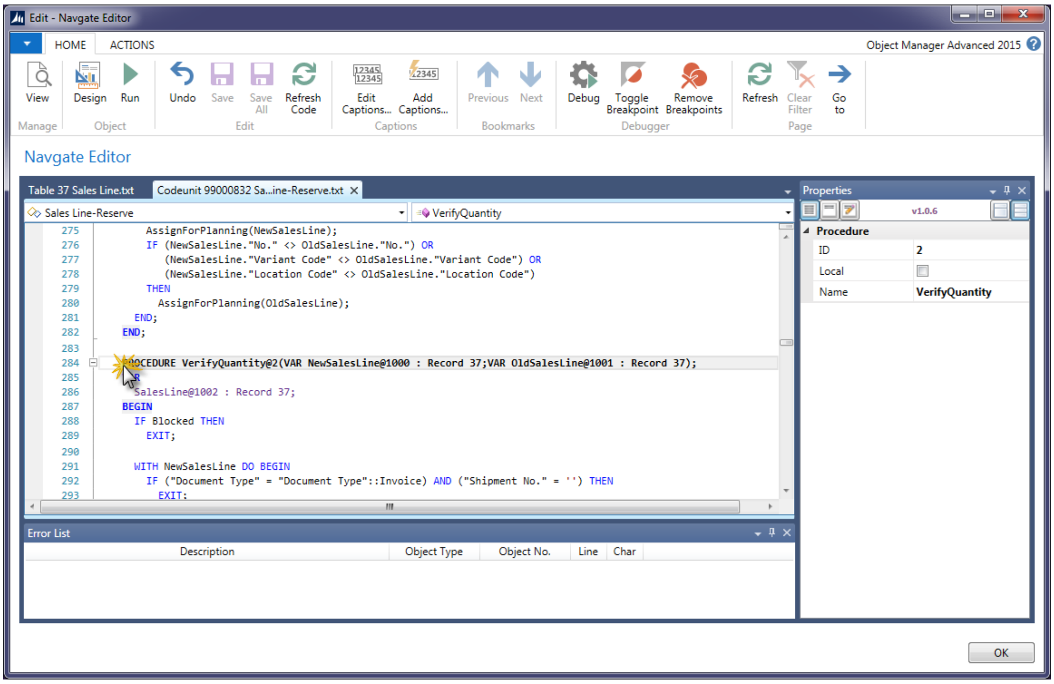


```
    Version List=NAVW18.00;
  }
  PROPERTIES
  {
    OnInsert=BEGIN
               TestStatusOpen;
               IF Quantity <> 0 THEN
                 ReserveSalesLine.VerifyQua
               LOCKTABLE;
               SalesHeader."No." := '';
               IF Type = Type::Item THEN
                 IF SalesHeader.InventoryPickConflict("Document Type","Document No.",SalesHeader."Shipping Advice"
                   ERROR(Text056,SalesHeader."Shipping Advice");
             END;

    OnModify=BEGIN
               IF ("Document Type" = "Document Type"::"Blanket Order") AND
                  ((Type <> xRec.Type) OR ("No." <> xRec."No."))
               THEN BEGIN
```

Table

CaptionML	ENU=Sales Line
DataCaptionField	
DataPerCompany	☑
Date	9/9/2014
Description	
DrillDownPageID	Sales Lines
ID	37
LinkedInTransacti	☑
LinkedObject	☐
LookupPageID	Sales Lines
Modified	☐
Name	Sales Line
ObjectType	Table
OnDelete	
OnInsert	
OnModify	
OnRename	
PasteIsValid	☐

Figure 6-1: Go To Definition



```
        AssignForPlanning(NewSalesLine);
        IF (NewSalesLine."No." <> OldSalesLine."No.") OR
           (NewSalesLine."Variant Code" <> OldSalesLine."Variant Code") OR
           (NewSalesLine."Location Code" <> OldSalesLine."Location Code")
        THEN
          AssignForPlanning(OldSalesLine);
      END;
    END;

    PROCEDURE VerifyQuantity@2(VAR NewSalesLine@1000 : Record 37;VAR OldSalesLine@1001 : Record 37);
    VAR
      SalesLine@1002 : Record 37;
    BEGIN
      IF Blocked THEN
        EXIT;

      WITH NewSalesLine DO BEGIN
        IF ("Document Type" = "Document Type"::Invoice) AND ("Shipment No." = '') THEN
          EXIT;
```

Procedure

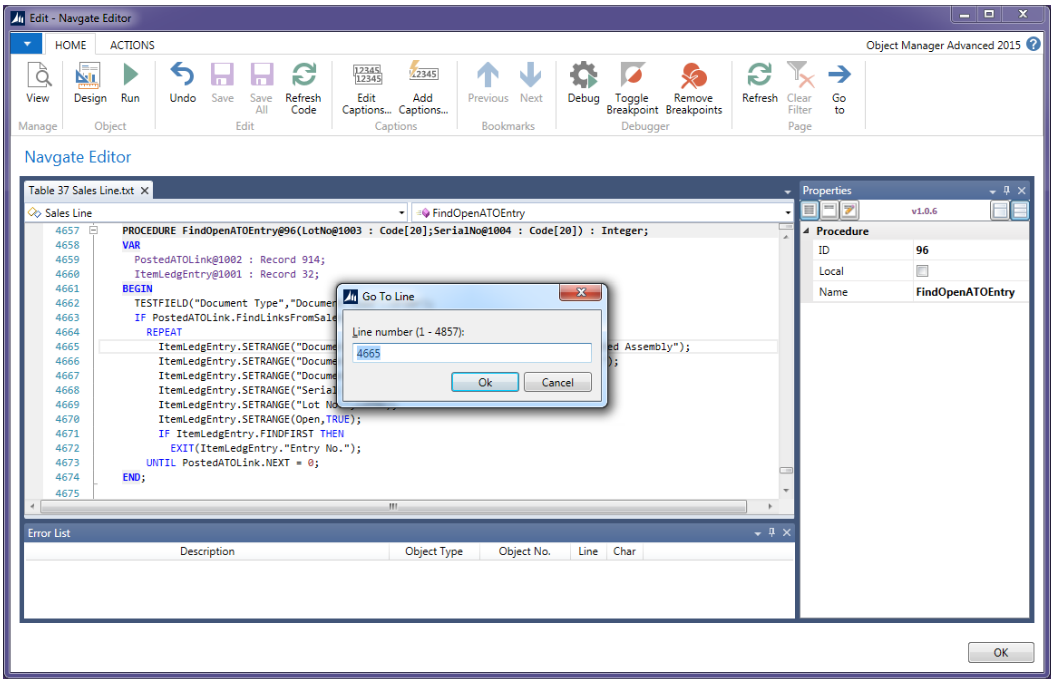
ID	2
Local	☐
Name	VerifyQuantity

Figure 6-2: Definition Opened in New Tab

It is possible to go to the definition of an object, field, function, variable and control.

6.2 GO TO LINE

To jump right to a specific line you can use the **Go To Line** function (**Ctrl+G**).



```
PROCEDURE FindOpenATOEntry@96(LotNo@1003 : Code[20];SerialNo@1004 : Code[20]) : Integer;
VAR
  PostedATOLink@1002 : Record 914;
  ItemLedgEntry@1001 : Record 32;
BEGIN
  TESTFIELD("Document Type","Documen
  IF PostedATOLink.FindLinksFromSale
    REPEAT
      ItemLedgEntry.SETRANGE("Docume                                    ed Assembly");
      ItemLedgEntry.SETRANGE("Docume                                    );
      ItemLedgEntry.SETRANGE("Docume
      ItemLedgEntry.SETRANGE("Serial
      ItemLedgEntry.SETRANGE("Lot No
      ItemLedgEntry.SETRANGE(Open,TRUE);
      IF ItemLedgEntry.FINDFIRST THEN
        EXIT(ItemLedgEntry."Entry No.");
    UNTIL PostedATOLink.NEXT = 0;
END;
```

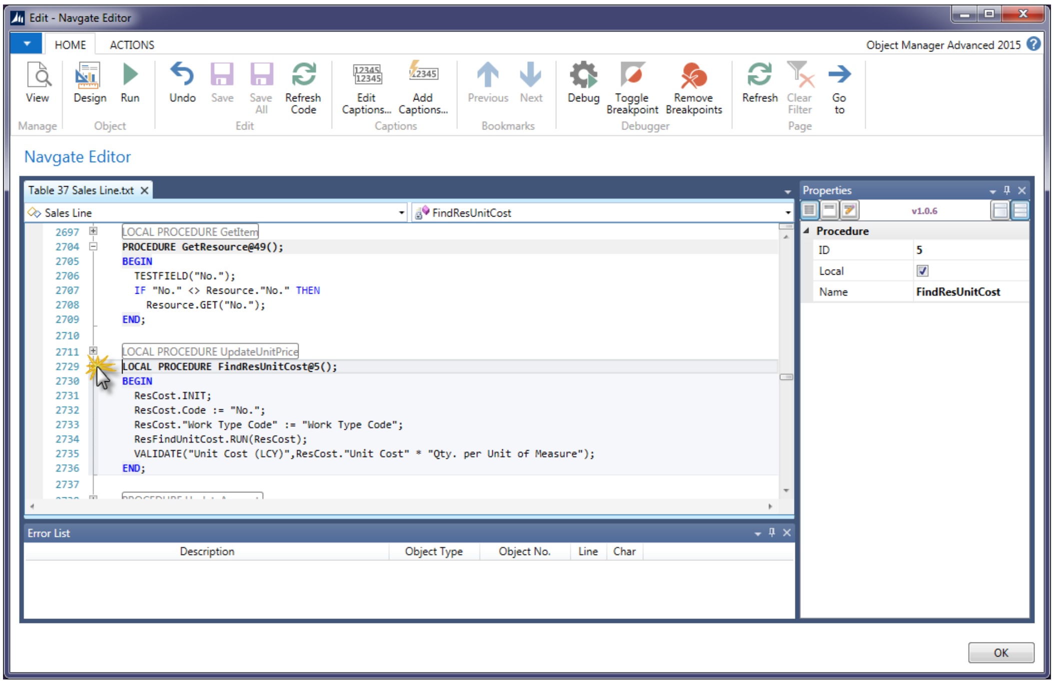
Line number (1 - 4857): 4665

Procedure
ID: 96
Local: ☐
Name: FindOpenATOEntry

Figure 6-3: Go To Line

6.3 CODE OUTLINING

Code Outlining allows you to collapse or expand regions of code along the container boundaries. A series of grouping lines and expand/collapse boxes are drawn in the selection margin. These expand/collapse boxes are clickable, enabling you to hide or display lines of code based on the logical grouping.

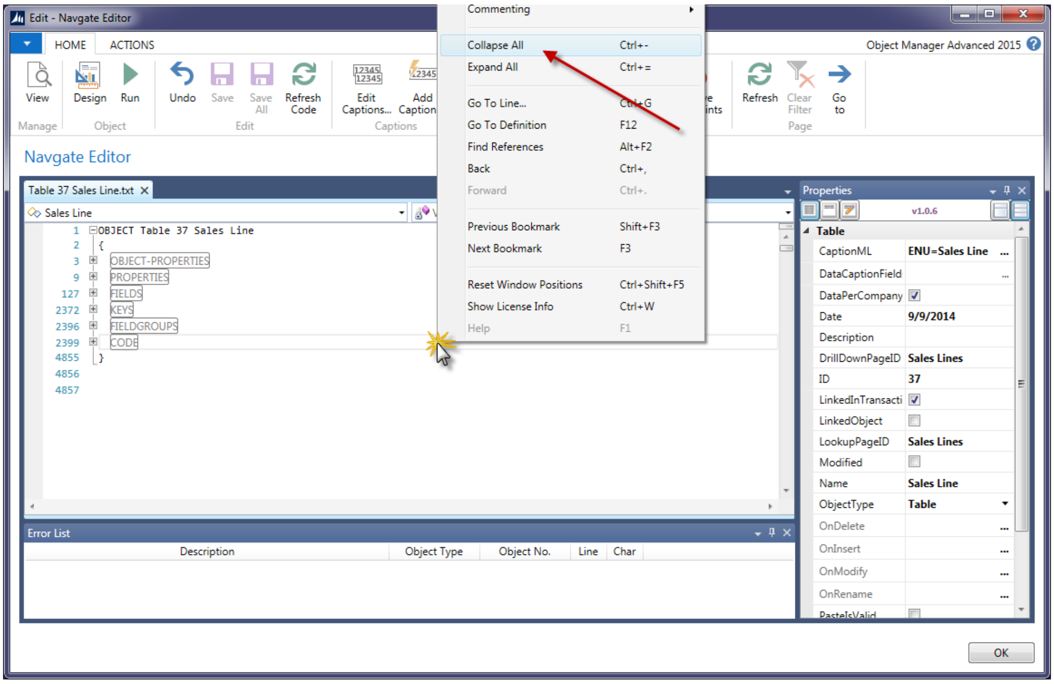


```
LOCAL PROCEDURE GetItem
PROCEDURE GetResource@49();
BEGIN
  TESTFIELD("No.");
  IF "No." <> Resource."No." THEN
    Resource.GET("No.");
END;

LOCAL PROCEDURE UpdateUnitPrice
LOCAL PROCEDURE FindResUnitCost@5();
BEGIN
  ResCost.INIT;
  ResCost.Code := "No.";
  ResCost."Work Type Code" := "Work Type Code";
  ResFindUnitCost.RUN(ResCost);
  VALIDATE("Unit Cost (LCY)",ResCost."Unit Cost" * "Qty. per Unit of Measure");
END;

```

Procedure
ID: 5
Local: ☑
Name: FindResUnitCost

Figure 6-4: Procedure Collapsed

Two code outlining commands are available in the **Context Menu** which can be opened with right click of the mouse. **Ctrl+Minus Sign (Ctrl+-)** will hide all outline regions. **Ctrl+EqualSign (Ctrl+=)** will unhide all outline regions.



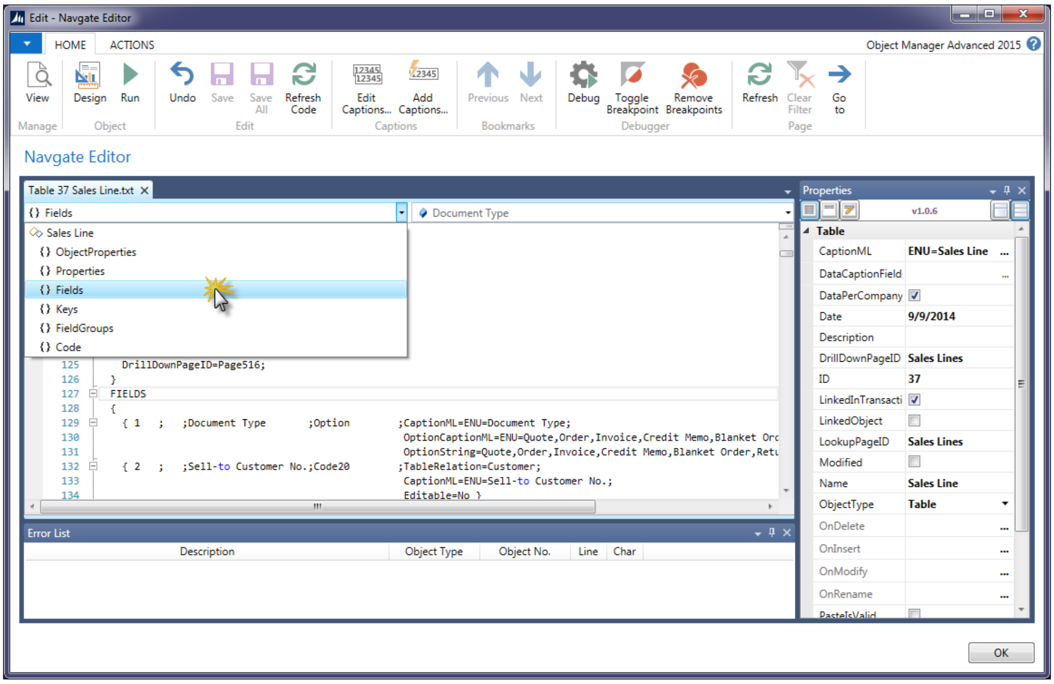
```
OBJECT Table 37 Sales Line
{
  OBJECT-PROPERTIES
  PROPERTIES
  FIELDS
  KEYS
  FIELDGROUPS
  CODE
}
```

Table	
CaptionML	ENU=Sales Line
DataCaptionField	
DataPerCompany	☑
Date	9/9/2014
Description	
DrillDownPageID	Sales Lines
ID	37
LinkedInTransacti	☑
LinkedObject	☐
LookupPageID	Sales Lines
Modified	☐
Name	Sales Line
ObjectType	Table
OnDelete	
OnInsert	
OnModify	
OnRename	
PasteIsValid	☐

Figure 6-5: Outlining Command in the Context Menu

6.4 NAVIGABLE SYMBOL SELECTOR

At the top of the editor are two dropdown selectors which make it easy to navigate to a section or element in the object. The left dropdown contains all the sections that are defined in the document.

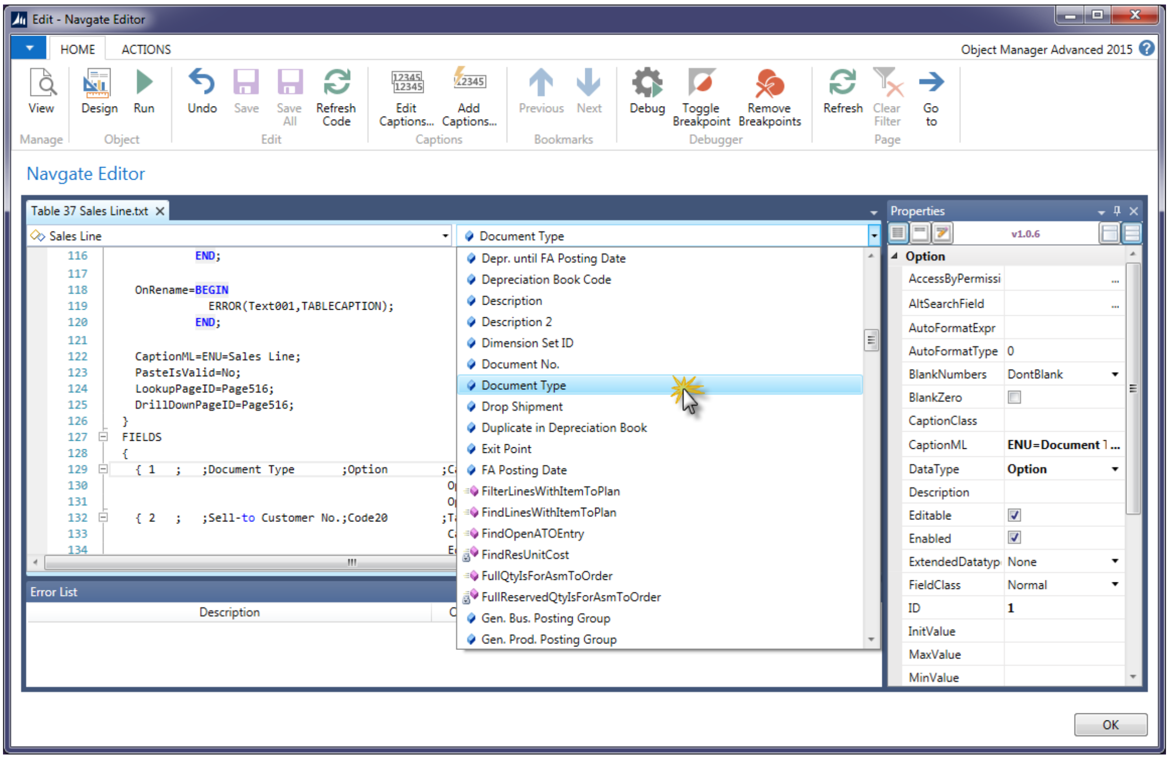


```
    DrillDownPageID=Page516;
  }
  FIELDS
  {
    { 1   ;   ;Document Type       ;Option        ;CaptionML=ENU=Document Type;
                                                   OptionCaptionML=ENU=Quote,Order,Invoice,Credit Memo,Blanket Orc
                                                   OptionString=Quote,Order,Invoice,Credit Memo,Blanket Order,Retu
    { 2   ;   ;Sell-to Customer No.;Code20        ;TableRelation=Customer;
                                                   CaptionML=ENU=Sell-to Customer No.;
                                                   Editable=No }
```

Table	
CaptionML	ENU=Sales Line
DataCaptionField	
DataPerCompany	☑
Date	9/9/2014
Description	
DrillDownPageID	Sales Lines
ID	37
LinkedInTransacti	☑
LinkedObject	☐
LookupPageID	Sales Lines
Modified	☐
Name	Sales Line
ObjectType	Table
OnDelete	
OnInsert	
OnModify	
OnRename	
PasteIsValid	☐

Figure 6-6: Navigable Symbol Selector

The right dropdown contains all element that are present in the object.

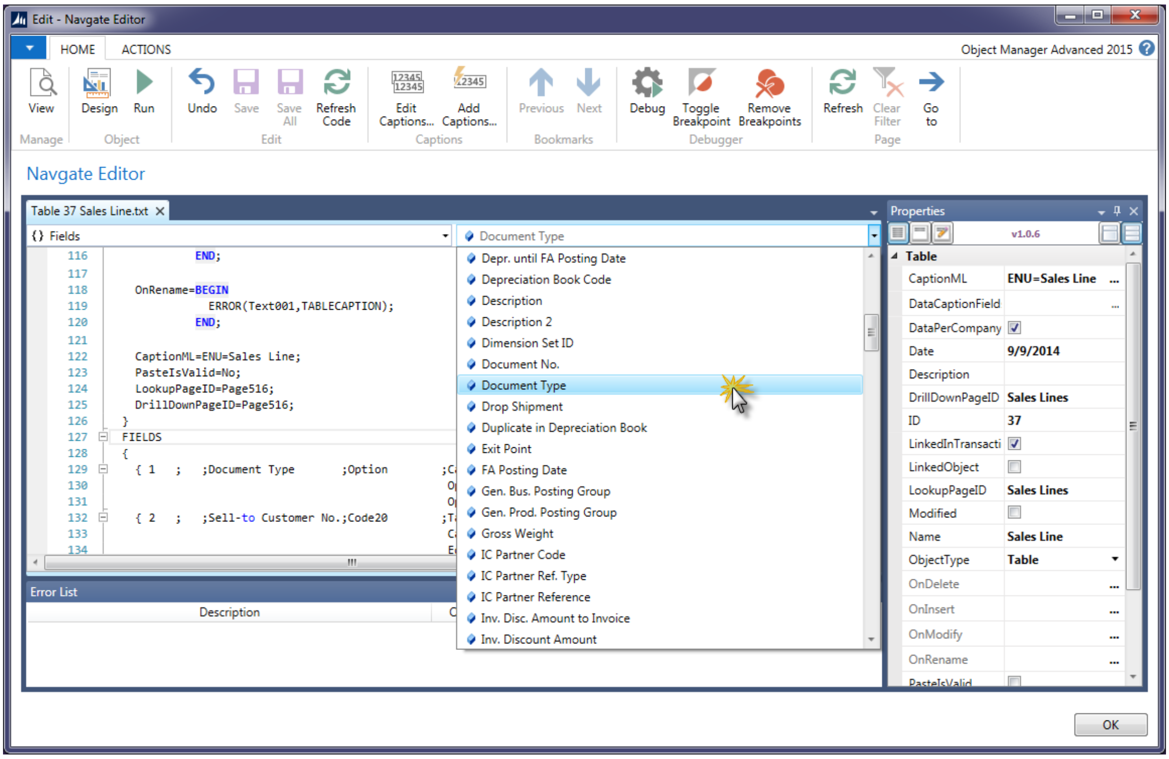


```
              END;

    OnRename=BEGIN
                ERROR(Text001,TABLECAPTION);
              END;

    CaptionML=ENU=Sales Line;
    PasteIsValid=No;
    LookupPageID=Page516;
    DrillDownPageID=Page516;
  }
  FIELDS
  {
    { 1   ;   ;Document Type       ;Option

    { 2   ;   ;Sell-to Customer No.;Code20
```

Option	
AccessByPermissi	...
AltSearchField	...
AutoFormatExpr	
AutoFormatType	0
BlankNumbers	DontBlank
BlankZero	☐
CaptionClass	
CaptionML	ENU=Document T...
DataType	Option
Description	
Editable	☑
Enabled	☑
ExtendedDatatyp	None
FieldClass	Normal
ID	1
InitValue	
MaxValue	
MinValue	

Figure 6-7: Functions and Fields in the Navigable Symbol Selector

After selecting a section on the left dropdown only the child elements are available in the right dropdown. In the following screen shot you can see that the procedures are not shown.



```
              END;

    OnRename=BEGIN
                ERROR(Text001,TABLECAPTION);
              END;

    CaptionML=ENU=Sales Line;
    PasteIsValid=No;
    LookupPageID=Page516;
    DrillDownPageID=Page516;
  }
  FIELDS
  {
    { 1   ;   ;Document Type       ;Option

    { 2   ;   ;Sell-to Customer No.;Code20
```

Table	
CaptionML	ENU=Sales Line ...
DataCaptionField	...
DataPerCompany	☑
Date	9/9/2014
Description	
DrillDownPageID	Sales Lines
ID	37
LinkedInTransacti	☑
LinkedObject	☐
LookupPageID	Sales Lines
Modified	☐
Name	Sales Line
ObjectType	Table
OnDelete	...
OnInsert	...
OnModify	...
OnRename	...
PasteIsValid	☐

Figure 6-8: Only Fields in the Navigable Symbol Selector

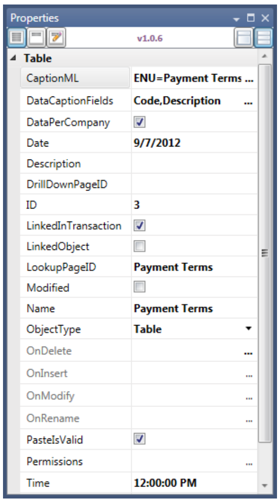
6.5 NAVIGATION LOG

When navigating with one of the above methods every location your cursor has been, is saved to a navigation log. You can navigate through these locations with the back and forward buttons on the side of your mouse. It is also possible to use **Ctrl+Comma** to go back (**Ctrl+, or Ctrl+<**) and **Ctrl+Period** to go forward (**Ctrl+. Ctrl+<).**

This is frequently used when you went to the definition of a keyword and you want to go back to the original location where you were.

7 PROPERTIES WINDOW

In a **C/AL** object (Tables, Fields, etc) everything has properties. **Navgate** provides a tool window named **Properties Window** which helps you set the properties in a simple way. The picture below shows the **Properties Window** for **Table 3: Payment Terms**. The **Properties Window** can be invoked by pressing **F4**.



CaptionML	ENU=Payment Terms ...
DataCaptionFields	Code,Description ...
DataPerCompany	☑
Date	9/7/2012
Description	
DrillDownPageID	
ID	3
LinkedInTransaction	☑
LinkedObject	☐
LookupPageID	Payment Terms
Modified	☐
Name	Payment Terms
ObjectType	Table
OnDelete	...
OnInsert	...
OnModify	...
OnRename	...
PasteIsValid	☑
Permissions	...
Time	12:00:00 PM

Figure 7-1: Properties Window

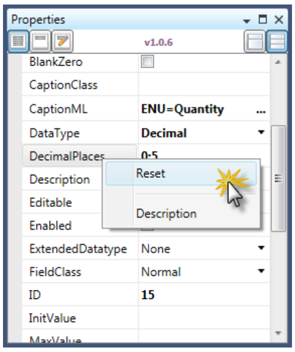
7.1 EDITING PROPERTIES

With the **Properties Window** it is easy to edit the properties of the elements you are working on. Every property type has its own input style. For example: a **Boolean** property has a checkbox and a numeric property only allows you to enter a valid numeric value.

When settings a property in the **Properties Window** your cursor jumps right to the edited property in the text editor allowing you to see what **Navgate** changes in your object.

For object names it is not needed to type the full name in the **Properties Window**. If you want to set **Payment Terms** in the **LookupPageID** property it is enough to type the first couple of characters like '**pay**' and **Navgate** will autocomplete the rest of the object name. If you know the number of the **LookupPageID** it is also possible to enter the number. **Navgate** will automatically replace it with the object name.

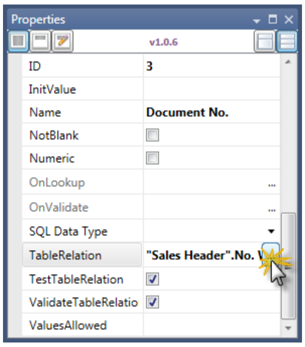
If you want to remove the property from your object you can right click with your mouse on the property name and click **Reset Property**.



BlankZero	☐
CaptionClass	
CaptionML	ENU=Quantity ...
DataType	Decimal
DecimalPlaces	0:5
Description	
Editable	
Enabled	
ExtendedDatatype	None
FieldClass	Normal
ID	15
InitValue	
MaxValue	

Figure 7-2: Reset Property

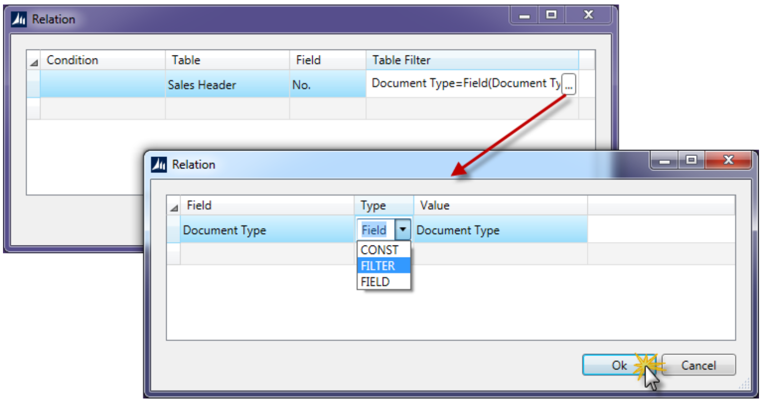
A lot of properties are difficult to edit in a single input field. To edit a table relation you must specify conditions and filters. To do this you can use the property dialogs. They can be invoked by clicking the three dots (**…**) in the **Properties Window** or placing your cursor on the property in the text editor and press **F2**.



ID	3
InitValue	
Name	Document No.
NotBlank	☐
Numeric	☐
OnLookup	...
OnValidate	...
SQL Data Type	
TableRelation	"Sales Header".No. W
TestTableRelation	☑
ValidateTableRelatio	☑
ValuesAllowed	

Figure 7-3: Editing Table Relation

A new window will open where the table relation can be set.



Condition	Table	Field	Table Filter
	Sales Header	No.	Document Type=Field(Document Ty

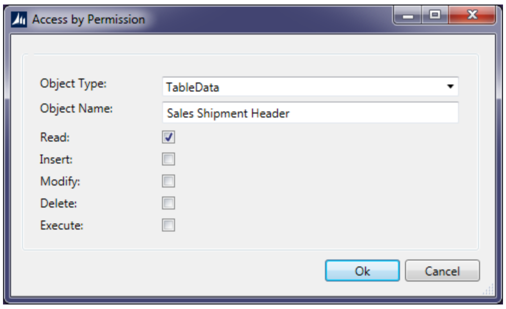
Field	Type	Value
Document Type	Field	Document Type

Figure 7-4: Dialog for Table Relation

Of course these windows have the same luxury as the **Properties Window**. It is not necessary to type full table and field names. The first characters or the number is enough. For field names a drop down is available to choose the field from a list.

The **Dialogs** below are available for the following property types:

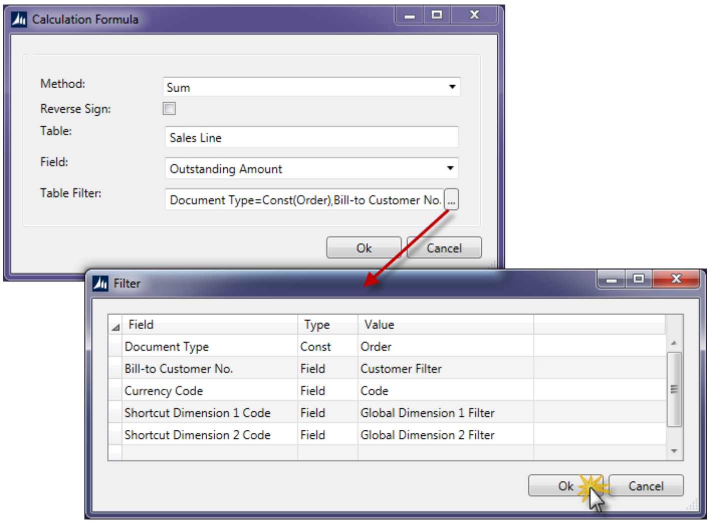
7.1.1 AccessByPermission



Object Type: TableData
Object Name: Sales Shipment Header
Read: ☑
Insert: ☐
Modify: ☐
Delete: ☐
Execute: ☐

Figure 7-5: Dialog for Access by Permission

7.1.2 CalcFormula

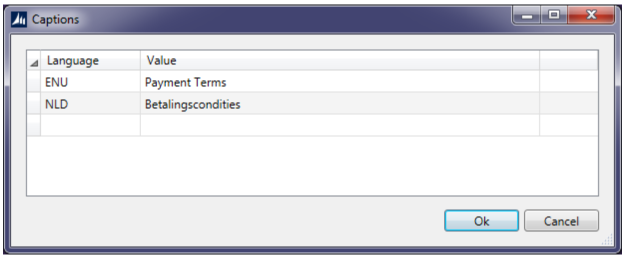


Method: Sum
Reverse Sign: ☐
Table: Sales Line
Field: Outstanding Amount
Table Filter: Document Type=Const(Order),Bill-to Customer No.

Field	Type	Value
Document Type	Const	Order
Bill-to Customer No.	Field	Customer Filter
Currency Code	Field	Code
Shortcut Dimension 1 Code	Field	Global Dimension 1 Filter
Shortcut Dimension 2 Code	Field	Global Dimension 2 Filter

Figure 7-6: Dialog for Calculation Formula

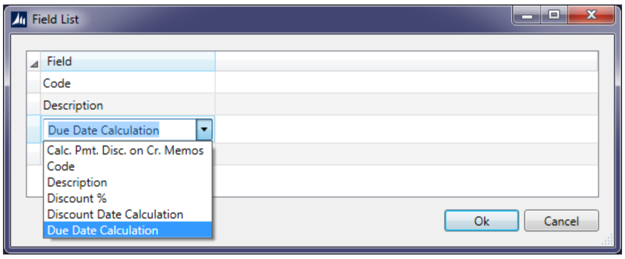
7.1.3 Caption



Language	Value
ENU	Payment Terms
NLD	Betalingscondities

Figure 7-7: Dialog for Caption

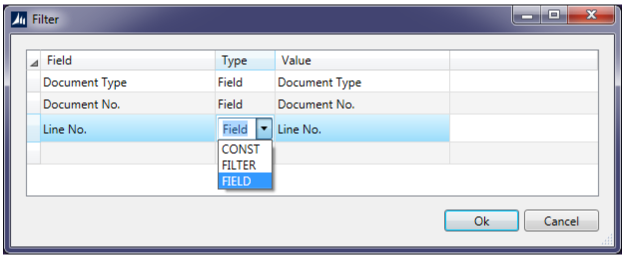
7.1.4 FieldList



Field
Code
Description
Due Date Calculation

Figure 7-8: Dialog for Field List

7.1.5 Link



Field	Type	Value
Document Type	Field	Document Type
Document No.	Field	Document No.
Line No.	Field	Line No.

Figure 7-9: Dialog for Link

7.1.6 Permissions

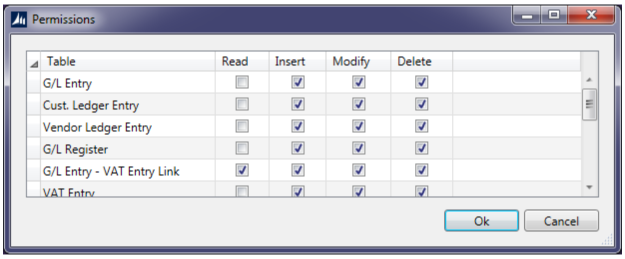
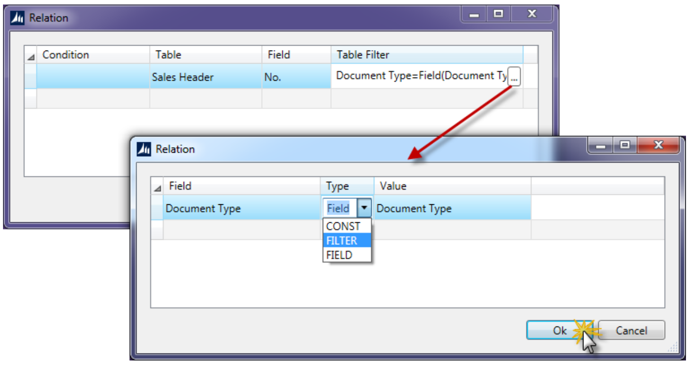


Table	Read	Insert	Modify	Delete
G/L Entry	☐	☑	☑	☑
Cust. Ledger Entry	☐	☑	☑	☑
Vendor Ledger Entry	☐	☑	☑	☑
G/L Register	☐	☑	☑	☑
G/L Entry - VAT Entry Link	☑	☑	☑	☑
VAT Entry	☐	☑	☑	☑

Figure 7-10: Dialog for Indirect Permissions

7.1.7 Relation

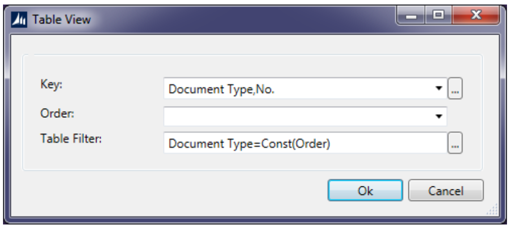


Condition	Table	Field	Table Filter
	Sales Header	No.	Document Type=Field(Document Ty

Field	Type	Value
Document Type	Field	Document Type

Figure 7-11: Dialog for Table Relation

7.1.8 View



Key: Document Type,No.

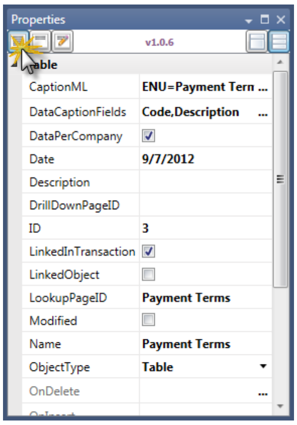
Order:

Table Filter: Document Type=Const(Order)

Figure 7-12: Dialog for View

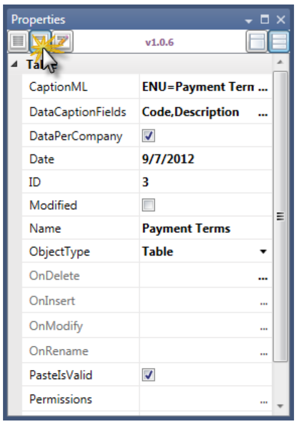
7.2 FILTERING PROPERTIES

The **Properties Window** shows by default all available properties. It is possible to filter the list in three ways as shown on the images below:



able	
CaptionML	ENU=Payment Tern ...
DataCaptionFields	Code,Description ...
DataPerCompany	☑
Date	9/7/2012
Description	
DrillDownPageID	
ID	3
LinkedInTransaction	☑
LinkedObject	☐
LookupPageID	Payment Terms
Modified	☐
Name	Payment Terms
ObjectType	Table
OnDelete	...

All Properties



Ta	
CaptionML	ENU=Payment Tern ...
DataCaptionFields	Code,Description ...
DataPerCompany	☑
Date	9/7/2012
ID	3
Modified	☐
Name	Payment Terms
ObjectType	Table
OnDelete	...
OnInsert	...
OnModify	...
OnRename	...
PasteIsValid	☑
Permissions	...

Most Used Properties

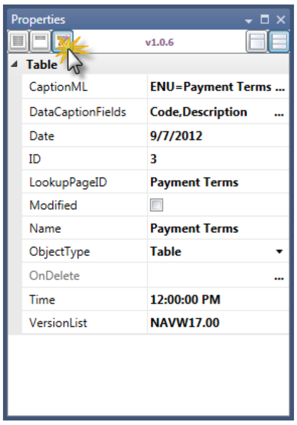
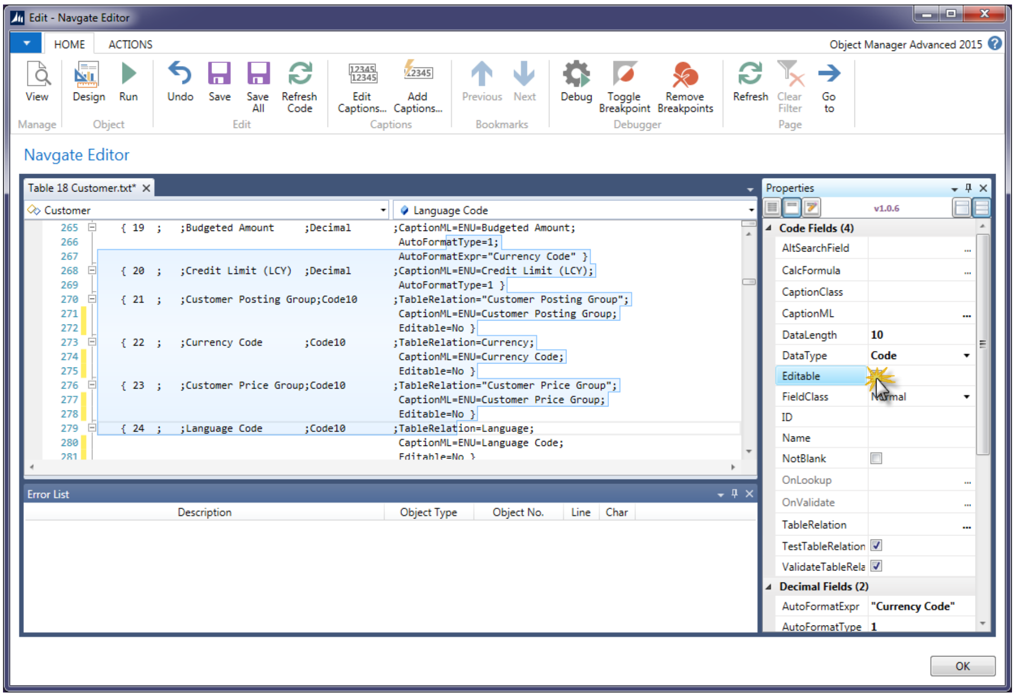


Table	
CaptionML	ENU=Payment Terms ...
DataCaptionFields	Code,Description ...
Date	9/7/2012
ID	3
LookupPageID	Payment Terms
Modified	☐
Name	Payment Terms
ObjectType	Table
OnDelete	...
Time	12:00:00 PM
VersionList	NAVW17.00

Specified Properties

7.3 GROUPING PROPERTIES

The **Properties Window** makes it possible to modify the properties of multiple elements at the same time. To do this select the elements in the text editor and modify the property. In the following example **6** fields are selected. In the **Property Window** they are split in two areas: **Code Fields (4)** and **Decimal Fields (2)**. When changing the **Editable Property** of the **Code Fields** only the code fields are updated.



```
{ 19  ;   ;Budgeted Amount     ;Decimal       ;CaptionML=ENU=Budgeted Amount;
                                               AutoFormatType=1;
                                               AutoFormatExpr="Currency Code" }
{ 20  ;   ;Credit Limit (LCY)  ;Decimal       ;CaptionML=ENU=Credit Limit (LCY);
                                               AutoFormatType=1 }
{ 21  ;   ;Customer Posting Group;Code10      ;TableRelation="Customer Posting Group";
                                               CaptionML=ENU=Customer Posting Group;
                                               Editable=No }
{ 22  ;   ;Currency Code       ;Code10        ;TableRelation=Currency;
                                               CaptionML=ENU=Currency Code;
                                               Editable=No }
{ 23  ;   ;Customer Price Group;Code10        ;TableRelation="Customer Price Group";
                                               CaptionML=ENU=Customer Price Group;
                                               Editable=No }
{ 24  ;   ;Language Code       ;Code10        ;TableRelation=Language;
                                               CaptionML=ENU=Language Code;
                                               Editable=No }
```

Code Fields (4)

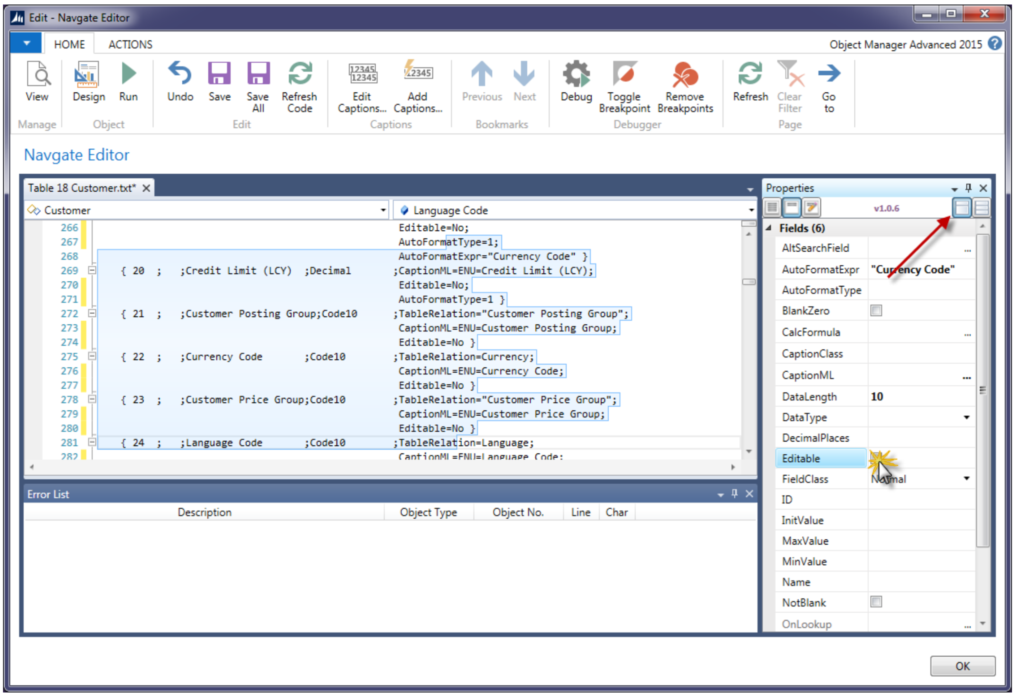
AltSearchField	...
CalcFormula	...
CaptionClass	
CaptionML	...
DataLength	10
DataType	Code
Editable	
FieldClass	Normal
ID	
Name	
NotBlank	☐
OnLookup	...
OnValidate	...
TableRelation	...
TestTableRelation	☑
ValidateTableRela	☑

Decimal Fields (2)

AutoFormatExpr	"Currency Code"
AutoFormatType	1

Figure 7-13: Edit Multiple Properties at Once

To update all fields at once select the other grouping button in the top of the **Properties Window**. In this case all **6** fields are combined in the window and can be edited together.



```
                                               Editable=No;
                                               AutoFormatType=1;
                                               AutoFormatExpr="Currency Code" }
{ 20  ;   ;Credit Limit (LCY)  ;Decimal       ;CaptionML=ENU=Credit Limit (LCY);
                                               Editable=No;
                                               AutoFormatType=1 }
{ 21  ;   ;Customer Posting Group;Code10      ;TableRelation="Customer Posting Group";
                                               CaptionML=ENU=Customer Posting Group;
                                               Editable=No }
{ 22  ;   ;Currency Code       ;Code10        ;TableRelation=Currency;
                                               CaptionML=ENU=Currency Code;
                                               Editable=No }
{ 23  ;   ;Customer Price Group;Code10        ;TableRelation="Customer Price Group";
                                               CaptionML=ENU=Customer Price Group;
                                               Editable=No }
{ 24  ;   ;Language Code       ;Code10        ;TableRelation=Language;
                                               CaptionML=ENU=Language Code;
```

Fields (6)

AltSearchField	...
AutoFormatExpr	"Currency Code"
AutoFormatType	
BlankZero	☐
CalcFormula	...
CaptionClass	
CaptionML	...
DataLength	10
DataType	
DecimalPlaces	
Editable	
FieldClass	Normal
ID	
InitValue	
MaxValue	
MinValue	
Name	
NotBlank	☐
OnLookup	...

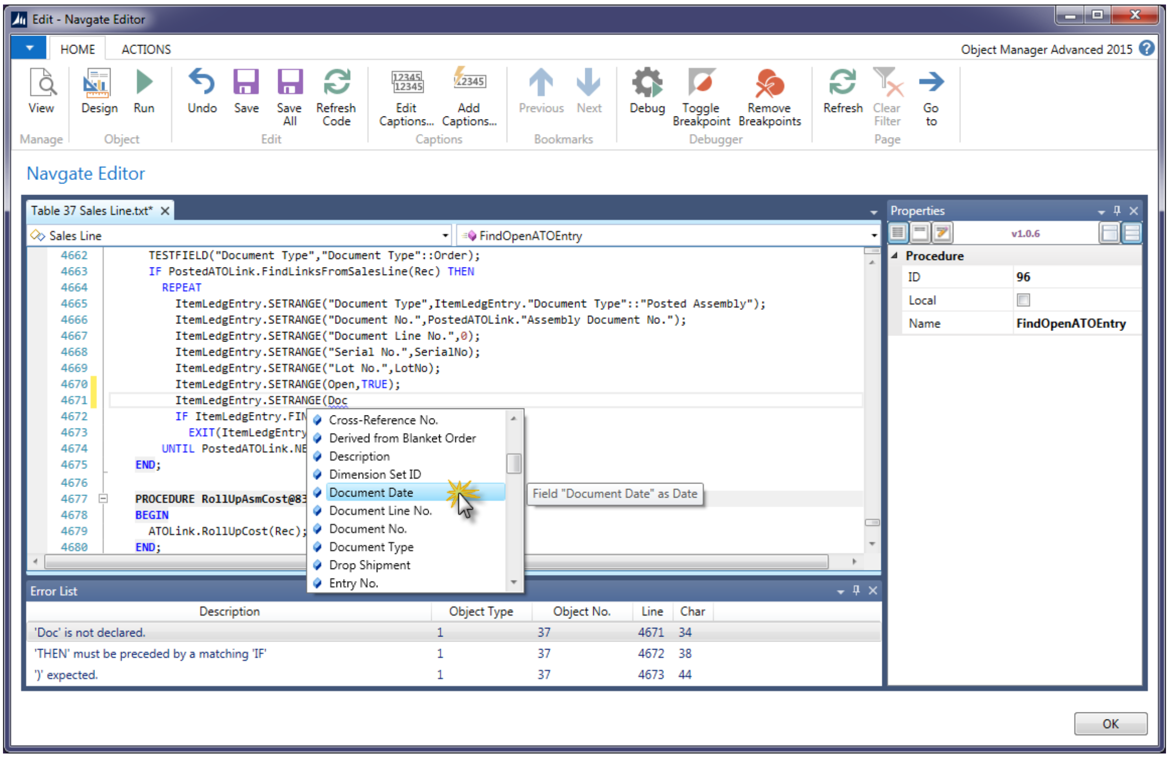
Figure 7-14: Combining Different Field Types

8 INTELLISENSE

IntelliSense is the name applied to a collection of useful functions available in **Navgate**. Its purpose is to speed up the writing of code and help you with writing code with less typos.

8.1 AUTO COMPLETION

Auto Completion is probably the best timesaving invention in the history of coding. After you have typed enough characters for **Navgate** to guess what you are trying to write you can press **Ctrl+Space** and **Navgate** will complete the word that you were typing. If multiple options are possible a list of all possible alternatives (known as the completion list) is presented.



```
    TESTFIELD("Document Type","Document Type"::Order);
    IF PostedATOLink.FindLinksFromSalesLine(Rec) THEN
      REPEAT
        ItemLedgEntry.SETRANGE("Document Type",ItemLedgEntry."Document Type"::"Posted Assembly");
        ItemLedgEntry.SETRANGE("Document No.",PostedATOLink."Assembly Document No.");
        ItemLedgEntry.SETRANGE("Document Line No.",0);
        ItemLedgEntry.SETRANGE("Serial No.",SerialNo);
        ItemLedgEntry.SETRANGE("Lot No.",LotNo);
        ItemLedgEntry.SETRANGE(Open,TRUE);
        ItemLedgEntry.SETRANGE(Doc
        IF ItemLedgEntry.FIN
          EXIT(ItemLedgEntry
      UNTIL PostedATOLink.NE
  END;

  PROCEDURE RollUpAsmCost@83
  BEGIN
    ATOLink.RollUpCost(Rec);
  END;
```

Description	Object Type	Object No.	Line	Char
'Doc' is not declared.	1	37	4671	34
'THEN' must be preceded by a matching 'IF'	1	37	4672	38
')' expected.	1	37	4673	44

ID	96
Local	☐
Name	FindOpenATOEntry

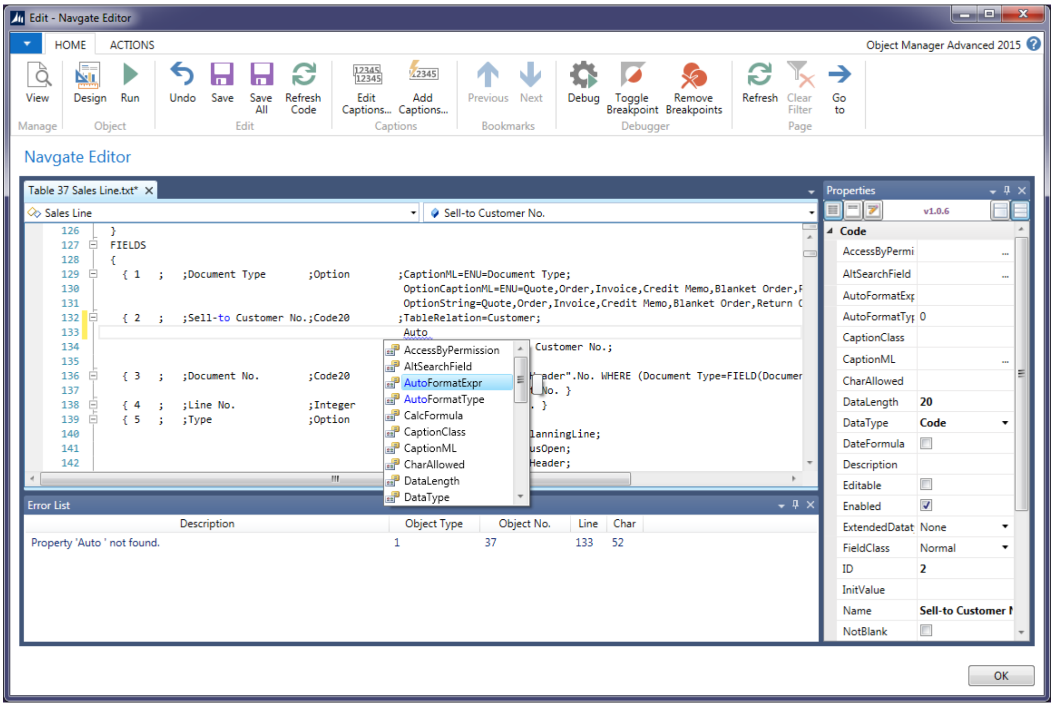
Figure 8-1: Auto Completion

You can then click the word that applies to your case or press the **TAB** key to complete the word. **Navgate** knows the context of the word you are typing. In the example in the above screen shot you can see that only field names of the **Item Ledger Entry** table are available in the list.

The completion list is automatically opened when typing a period.

If the completion list is opened and you type a period the selected item in the list is injected in the editor and a new completion list is opened with all valid members.

Auto Completion is also enabled in properties and property values.



```
126   }
127   FIELDS
128   {
129     { 1   ;   ;Document Type       ;Option        ;CaptionML=ENU=Document Type;
130                                                   OptionCaptionML=ENU=Quote,Order,Invoice,Credit Memo,Blanket Order,R
131                                                   OptionString=Quote,Order,Invoice,Credit Memo,Blanket Order,Return C
132     { 2   ;   ;Sell-to Customer No.;Code20        ;TableRelation=Customer;
133                                                   Auto
134                                                                         Customer No.;
135
136     { 3   ;   ;Document No.        ;Code20                              ader".No. WHERE (Document Type=FIELD(Documen
137                                                                         No. }
138     { 4   ;   ;Line No.            ;Integer                           . }
139     { 5   ;   ;Type                ;Option
140                                                                        lanningLine;
141                                                                        usOpen;
142                                                                        Header;
```

Description	Object Type	Object No.	Line	Char
Property 'Auto ' not found.	1	37	133	52

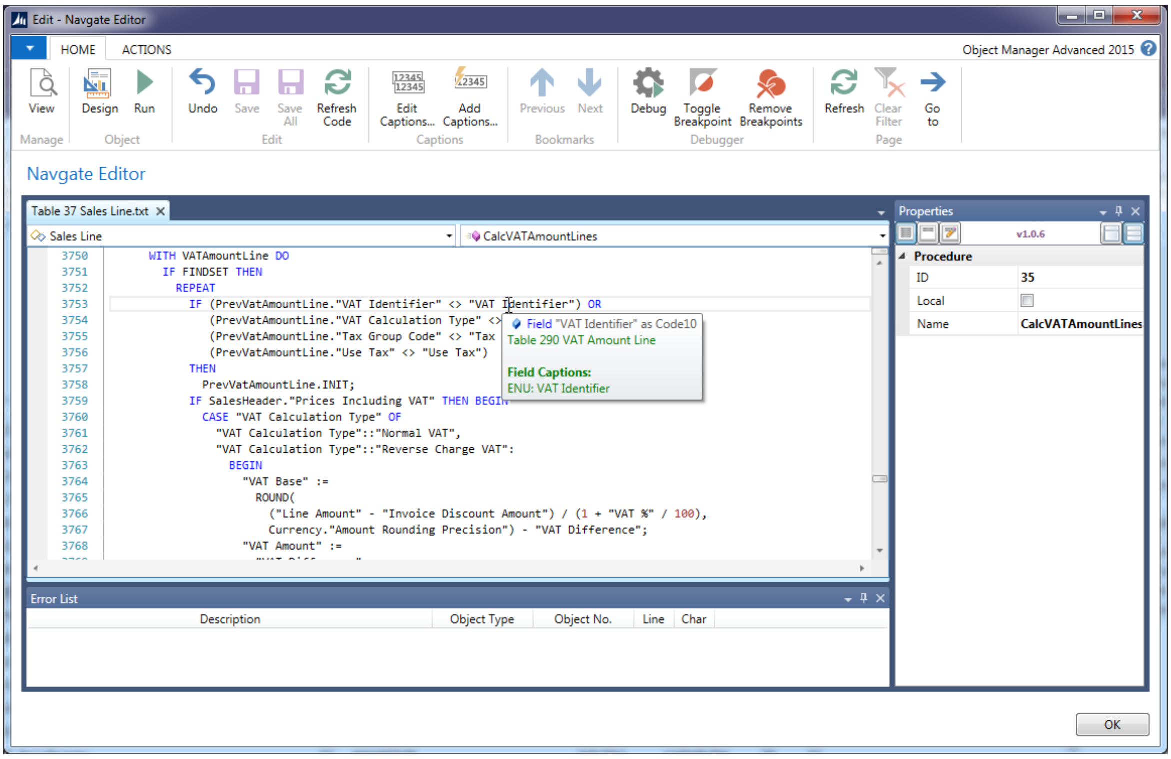
Code	
AccessByPermi	...
AltSearchField	...
AutoFormatExp	
AutoFormatTyp	0
CaptionClass	
CaptionML	...
CharAllowed	
DataLength	**20**
DataType	**Code**
DateFormula	☐
Description	
Editable	☐
Enabled	☑
ExtendedDatat	None
FieldClass	Normal
ID	**2**
InitValue	
Name	**Sell-to Customer N**
NotBlank	☐

Figure 8-2: Auto Completion on a Property Name

TIP: Holding down the Ctrl key while the completion list is opened makes it partially transparent. This is useful if, during selecting an item from the list, you need to see any of the lines of code that are hidden behind the list

8.2 QUICK INFO

Quick Info displays the complete code declaration and help information for any keyword in the object. You invoke it by pointing your mouse to the keyword.



```
    WITH VATAmountLine DO
      IF FINDSET THEN
        REPEAT
          IF (PrevVatAmountLine."VAT Identifier" <> "VAT Identifier") OR
             (PrevVatAmountLine."VAT Calculation Type" <>
             (PrevVatAmountLine."Tax Group Code" <> "Tax
             (PrevVatAmountLine."Use Tax" <> "Use Tax")
          THEN
            PrevVatAmountLine.INIT;
          IF SalesHeader."Prices Including VAT" THEN BEGIN
            CASE "VAT Calculation Type" OF
              "VAT Calculation Type"::"Normal VAT",
              "VAT Calculation Type"::"Reverse Charge VAT":
                BEGIN
                  "VAT Base" :=
                    ROUND(
                      ("Line Amount" - "Invoice Discount Amount") / (1 + "VAT %" / 100),
                      Currency."Amount Rounding Precision") - "VAT Difference";
                  "VAT Amount" :=
```

Field "VAT Identifier" as Code10
Table 290 VAT Amount Line

Field Captions:
ENU: VAT Identifier

Procedure
ID 35
Local ☐
Name CalcVATAmountLines

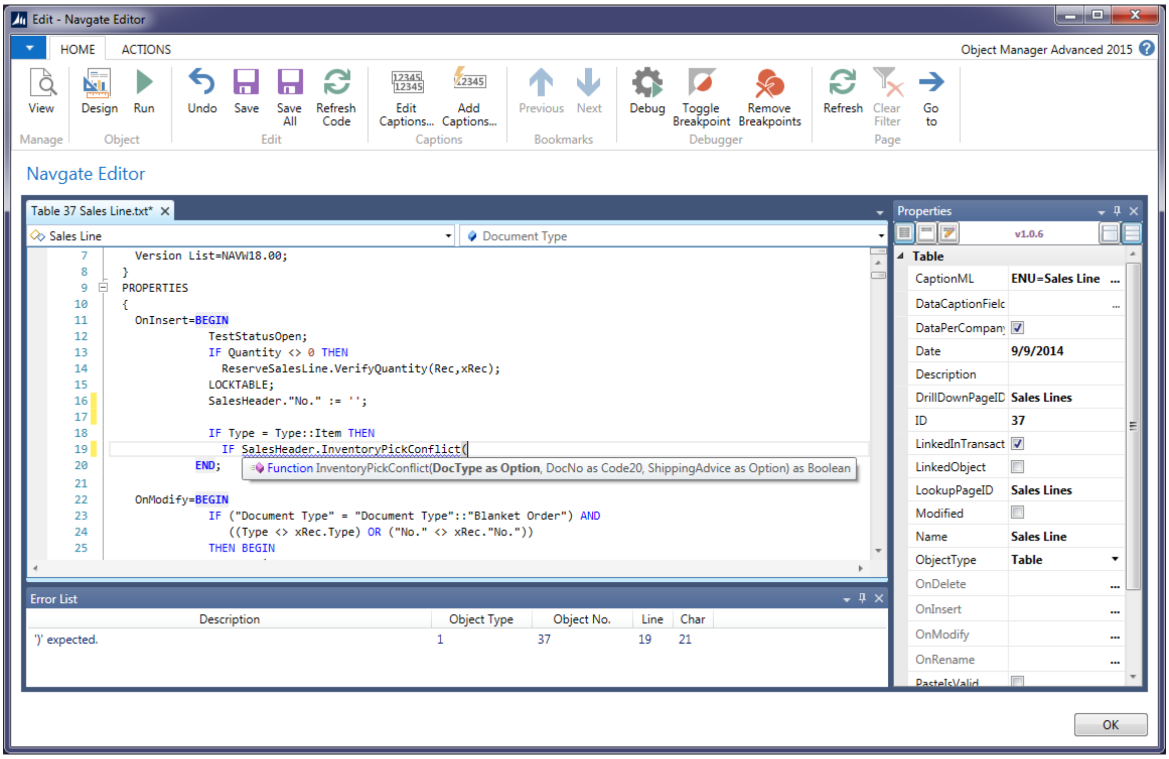
Figure 8-3: Quick Info

In the example in the above screen shot you can see that the **VAT Identifier** field value is **Code10** and belongs to **Table 290: Vat Amount Line**. Beside that you can see the caption of the field.

8.3 PARAMETER INFO

Parameter Info is designed to provide interactive guidance for the parameters needed for any given function call. This feature proves especially useful for making function calls that have a long list of parameters.

Parameter Info is invoked whenever you type an opening parenthesis after a function name or pressing **Ctrl+Shift+Space**.



```
  Version List=NAVW18.00;
}
PROPERTIES
{
  OnInsert=BEGIN
             TestStatusOpen;
             IF Quantity <> 0 THEN
               ReserveSalesLine.VerifyQuantity(Rec,xRec);
             LOCKTABLE;
             SalesHeader."No." := '';

             IF Type = Type::Item THEN
               IF SalesHeader.InventoryPickConflict(
           END;

  OnModify=BEGIN
             IF ("Document Type" = "Document Type"::"Blanket Order") AND
                ((Type <> xRec.Type) OR ("No." <> xRec."No."))
             THEN BEGIN
```

Function InventoryPickConflict(**DocType as Option**, DocNo as Code20, ShippingAdvice as Option) as Boolean

Description	Object Type	Object No.	Line	Char
')' expected.	1	37	19	21

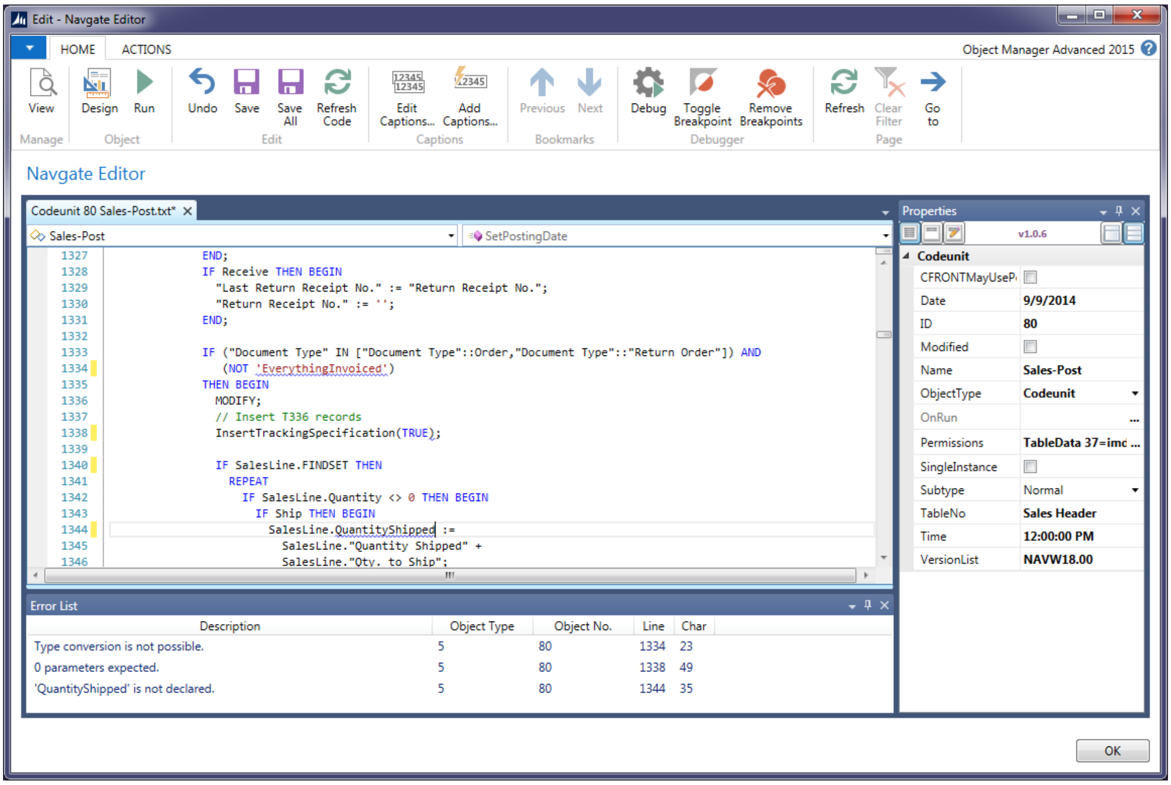
Table	
CaptionML	**ENU=Sales Line** ...
DataCaptionFielc	...
DataPerCompany	☑
Date	**9/9/2014**
Description	
DrillDownPageID	**Sales Lines**
ID	**37**
LinkedInTransact	☑
LinkedObject	☐
LookupPageID	**Sales Lines**
Modified	☐
Name	**Sales Line**
ObjectType	**Table**
OnDelete	...
OnInsert	...
OnModify	...
OnRename	...
PasteIsValid	☐

Figure 8-4: Parameter Info

As you type, the parameter info pop-up continues to guide you through the parameters by bolding the current parameter.

9 ERROR LIST

The **Error List** in the bottom of **Navgate** lists all errors that are present in all open objects. You can double click on an error to jump right to the location of the error. All errors are recognizable by a blue squiggle line.



```
END;
IF Receive THEN BEGIN
  "Last Return Receipt No." := "Return Receipt No.";
  "Return Receipt No." := '';
END;

IF ("Document Type" IN ["Document Type"::Order,"Document Type"::"Return Order"]) AND
   (NOT 'EverythingInvoiced')
THEN BEGIN
  MODIFY;
  // Insert T336 records
  InsertTrackingSpecification(TRUE);

  IF SalesLine.FINDSET THEN
    REPEAT
      IF SalesLine.Quantity <> 0 THEN BEGIN
        IF Ship THEN BEGIN
          SalesLine.QuantityShipped :=
            SalesLine."Quantity Shipped" +
            SalesLine."Qty. to Ship";
```

Error List

Description	Object Type	Object No.	Line	Char
Type conversion is not possible.	5	80	1334	23
0 parameters expected.	5	80	1338	49
'QuantityShipped' is not declared.	5	80	1344	35

Codeunit

CFRONTMayUseP	☐
Date	9/9/2014
ID	80
Modified	☐
Name	Sales-Post
ObjectType	Codeunit
OnRun	...
Permissions	TableData 37=imd ...
SingleInstance	☐
Subtype	Normal
TableNo	Sales Header
Time	12:00:00 PM
VersionList	NAVW18.00

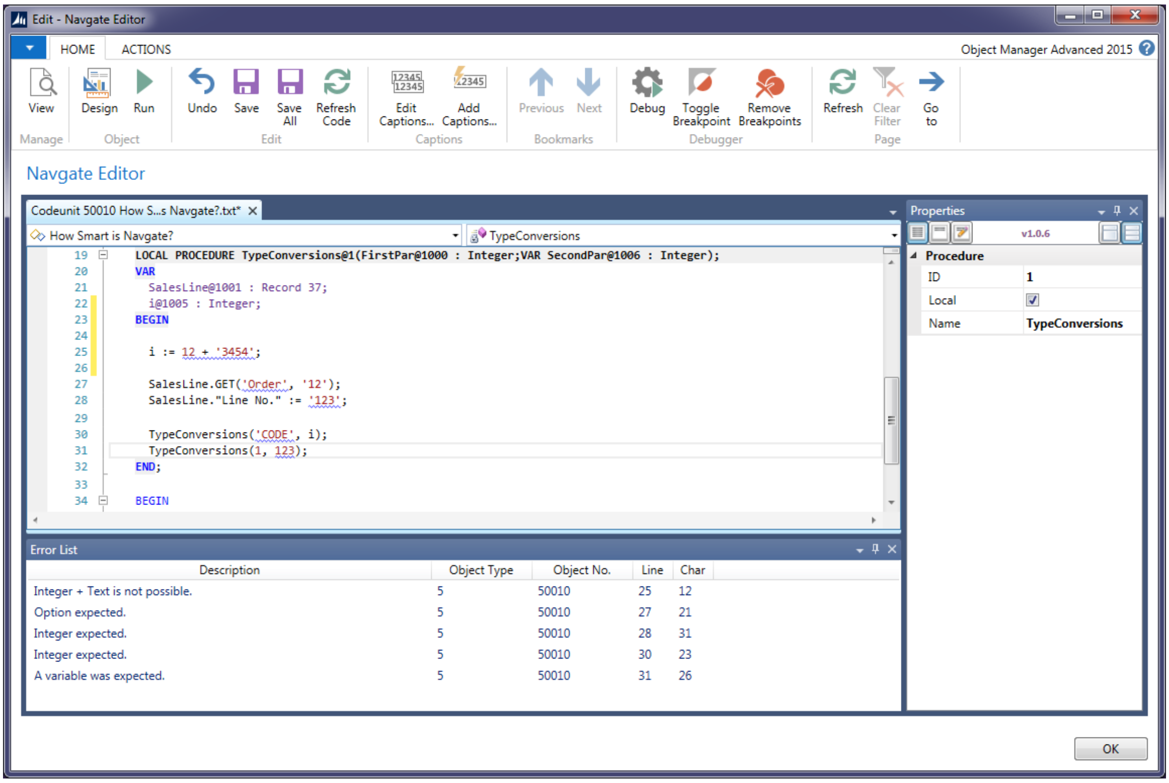
Figure 9-1: Errors Shown in the Error List

9.1 UNKNOWN DECLARATIONS

If a word is not recognized by **Navgate**, it is placed in the **Error List**. In the above screen shot you can see that **QuantityShipped** is not a member of **SalesLine**.

9.2 TYPE CONVERSION ERRORS

All type conversion errors are traced. It is for example not possible to add an integer with a text variable. Or if you use an integer as a parameter where a code field is expected. See the following screen shot for some examples.



```
19    LOCAL PROCEDURE TypeConversions@1(FirstPar@1000 : Integer;VAR SecondPar@1006 : Integer);
20    VAR
21      SalesLine@1001 : Record 37;
22      i@1005 : Integer;
23    BEGIN
24
25      i := 12 + '3454';
26
27      SalesLine.GET('Order', '12');
28      SalesLine."Line No." := '123';
29
30      TypeConversions('CODE', i);
31      TypeConversions(1, 123);
32    END;
33
34    BEGIN
```

Description	Object Type	Object No.	Line	Char
Integer + Text is not possible.	5	50010	25	12
Option expected.	5	50010	27	21
Integer expected.	5	50010	28	31
Integer expected.	5	50010	30	23
A variable was expected.	5	50010	31	26

Procedure

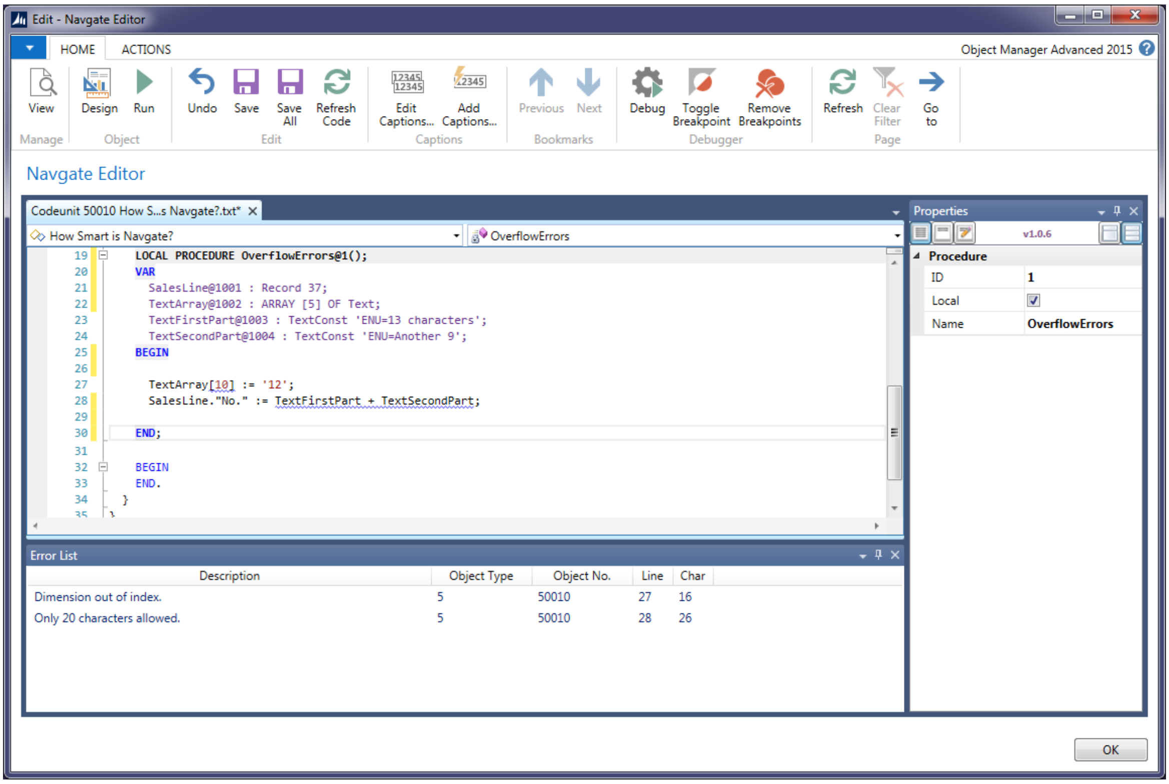
ID	1
Local	☑
Name	**TypeConversions**

Figure 9-2: Type Conversion Errors

9.3 OVERFLOW ERRORS

Overflow errors are traced. If you try to assign a **TextConst** or text string with length **21** to a field with length **20** you get an error in the **Error List**. **Navgate** even knows when strings are concatenated as you can see in the following screen shot.

It is also not possible to use an array outside its borders.



```
19      LOCAL PROCEDURE OverflowErrors@1();
20      VAR
21        SalesLine@1001 : Record 37;
22        TextArray@1002 : ARRAY [5] OF Text;
23        TextFirstPart@1003 : TextConst 'ENU=13 characters';
24        TextSecondPart@1004 : TextConst 'ENU=Another 9';
25      BEGIN
26
27        TextArray[10] := '12';
28        SalesLine."No." := TextFirstPart + TextSecondPart;
29
30      END;
31
32      BEGIN
33      END.
34    }
35  }
```

Procedure

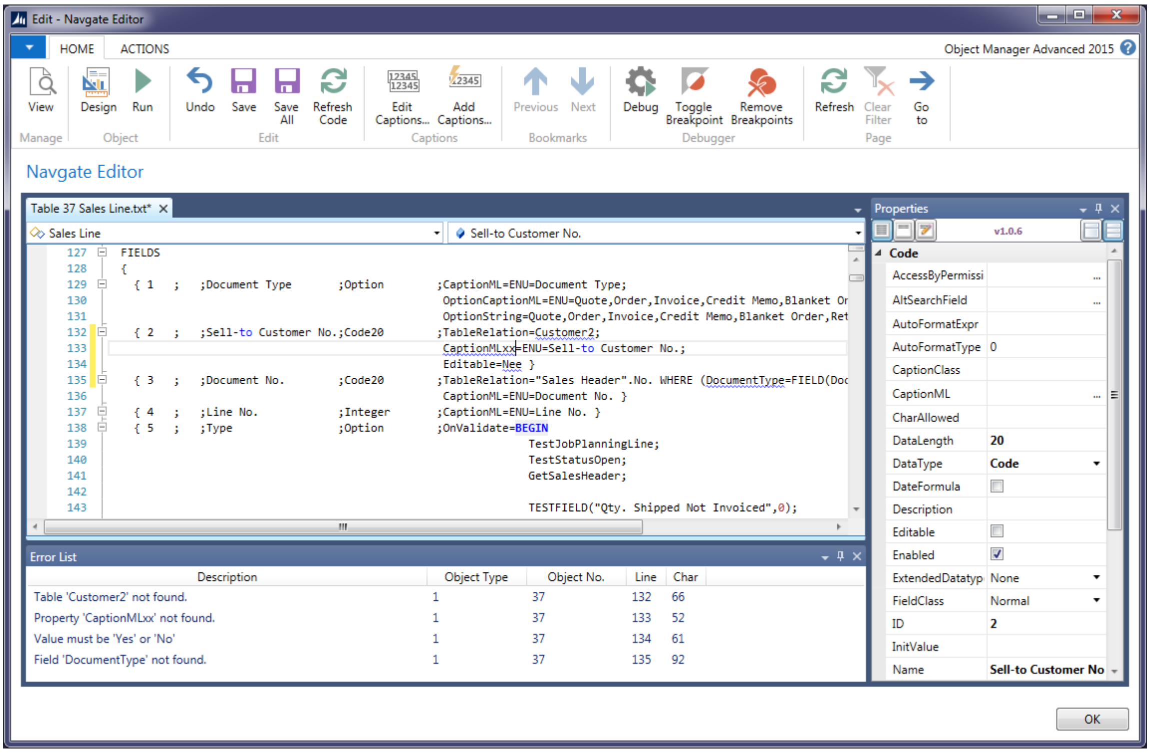
ID	1
Local	☑
Name	OverflowErrors

Description	Object Type	Object No.	Line	Char
Dimension out of index.	5	50010	27	16
Only 20 characters allowed.	5	50010	28	26

Figure 9-3: Overflow Errors

9.4 PROPERTY ERRORS

Not only code segments are parsed and checked for errors but also all elements and properties. If a property is not known or not in the right context it gets flagged as an error. See the following screen shot for some examples.



```
127   FIELDS
128   {
129     { 1   ;   ;Document Type       ;Option        ;CaptionML=ENU=Document Type;
130                                                    OptionCaptionML=ENU=Quote,Order,Invoice,Credit Memo,Blanket Or
131                                                    OptionString=Quote,Order,Invoice,Credit Memo,Blanket Order,Ret
132     { 2   ;   ;Sell-to Customer No.;Code20        ;TableRelation=Customer2;
133                                                    CaptionMLxx=ENU=Sell-to Customer No.;
134                                                    Editable=Nee }
135     { 3   ;   ;Document No.        ;Code20        ;TableRelation="Sales Header".No. WHERE (DocumentType=FIELD(Doc
136                                                    CaptionML=ENU=Document No. }
137     { 4   ;   ;Line No.            ;Integer       ;CaptionML=ENU=Line No. }
138     { 5   ;   ;Type                ;Option        ;OnValidate=BEGIN
139                                                                 TestJobPlanningLine;
140                                                                 TestStatusOpen;
141                                                                 GetSalesHeader;
142 
143                                                                 TESTFIELD("Qty. Shipped Not Invoiced",0);
```

Error List

Description	Object Type	Object No.	Line	Char
Table 'Customer2' not found.	1	37	132	66
Property 'CaptionMLxx' not found.	1	37	133	52
Value must be 'Yes' or 'No'	1	37	134	61
Field 'DocumentType' not found.	1	37	135	92

Code

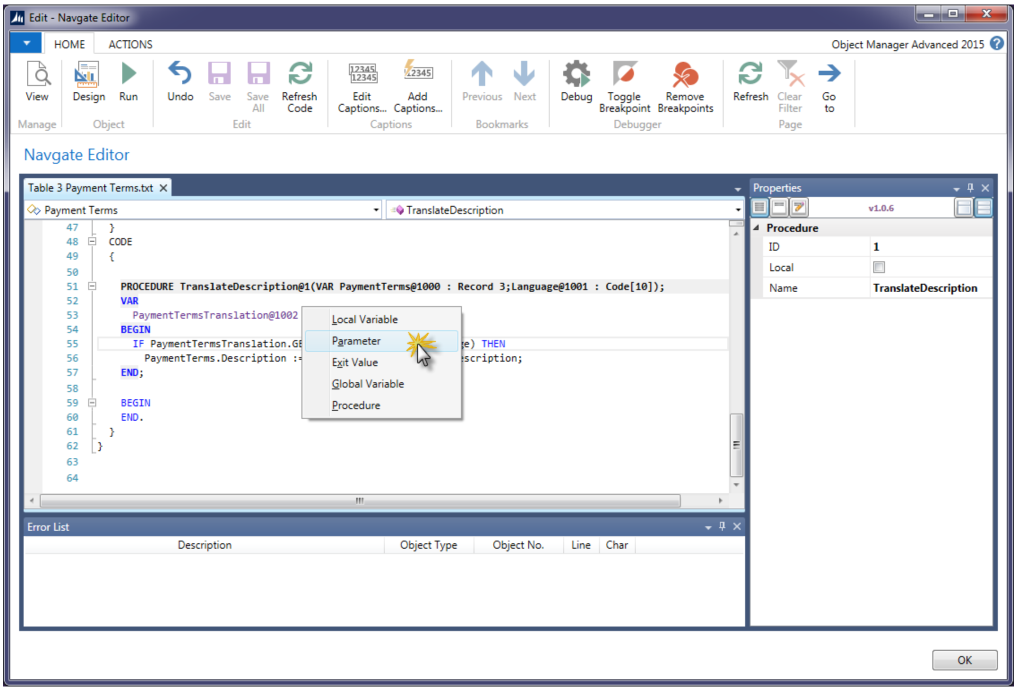
AccessByPermissi	...
AltSearchField	...
AutoFormatExpr	
AutoFormatType	0
CaptionClass	
CaptionML	...
CharAllowed	
DataLength	**20**
DataType	**Code**
DateFormula	☐
Description	
Editable	☐
Enabled	☑
ExtendedDatatyp	None
FieldClass	Normal
ID	**2**
InitValue	
Name	**Sell-to Customer No**

Figure 9-4: Property Errors

10 ADD ELEMENTS

The grammar of **C/AL** objects is at some places very difficult to write. Adding a variable for example must be done in a very precise way, otherwise the **NAV Compiler** will result with an error. You can use the **Add Element** functionality to help you create code that **NAV** can understand.

Open the context menu with available elements by pressing **Ctrl+N**.



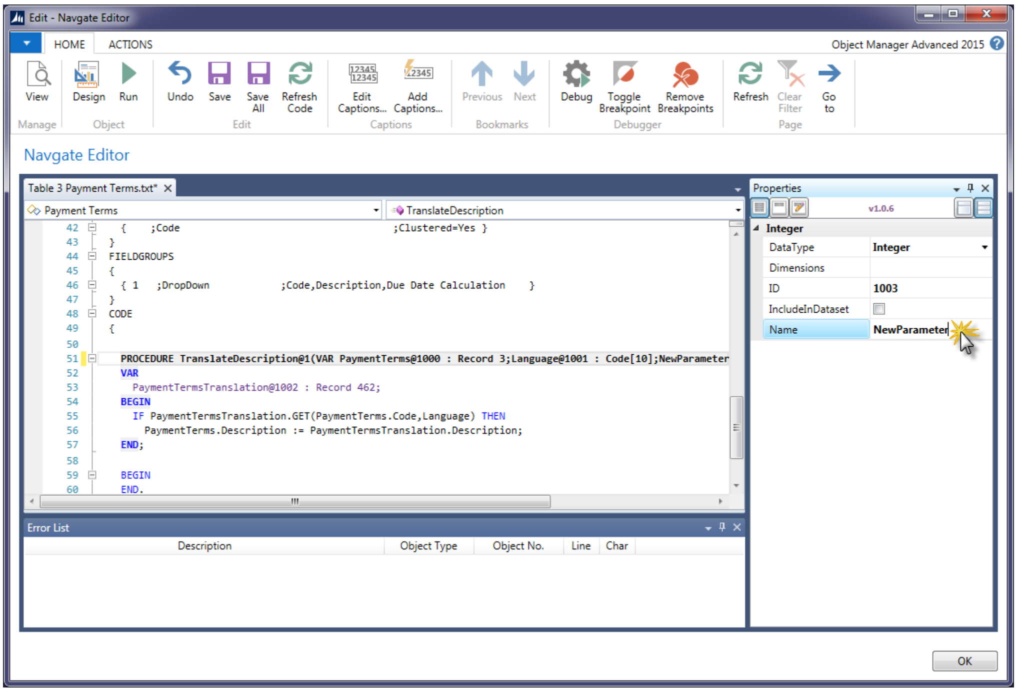
```
  }
  CODE
  {

    PROCEDURE TranslateDescription@1(VAR PaymentTerms@1000 : Record 3;Language@1001 : Code[10]);
    VAR
      PaymentTermsTranslation@1002
    BEGIN
      IF PaymentTermsTranslation.GE                 e) THEN
        PaymentTerms.Description :=                 scription;
    END;

    BEGIN
    END.
  }
}
```

Procedure
ID 1
Local ☐
Name TranslateDescription

Figure 10-1: Add Elements

Depending on the location of your cursor **Navgate** populates this context menu with different elements.

After selecting the element, the cursor jumps right to the added position and the **Properties Window** is updated with the newly added element where you can specify all the necessary properties.

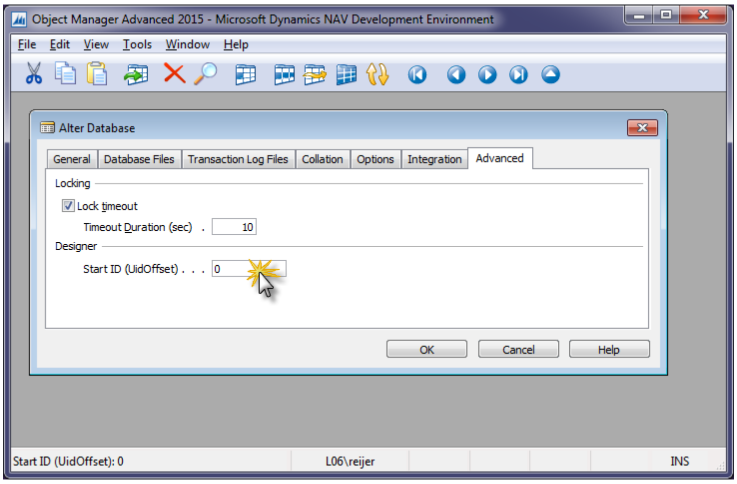


```
  {    ;Code                                        ;Clustered=Yes }
}
FIELDGROUPS
{
  { 1   ;DropDown            ;Code,Description,Due Date Calculation    }
}
CODE
{

  PROCEDURE TranslateDescription@1(VAR PaymentTerms@1000 : Record 3;Language@1001 : Code[10];NewParameter
  VAR
    PaymentTermsTranslation@1002 : Record 462;
  BEGIN
    IF PaymentTermsTranslation.GET(PaymentTerms.Code,Language) THEN
      PaymentTerms.Description := PaymentTermsTranslation.Description;
  END;

  BEGIN
  END.
```

Integer
DataType: Integer
Dimensions:
ID: 1003
IncludeInDataset: ☐
Name: NewParameter

Figure 10-2: Parameter Added to the Procedure

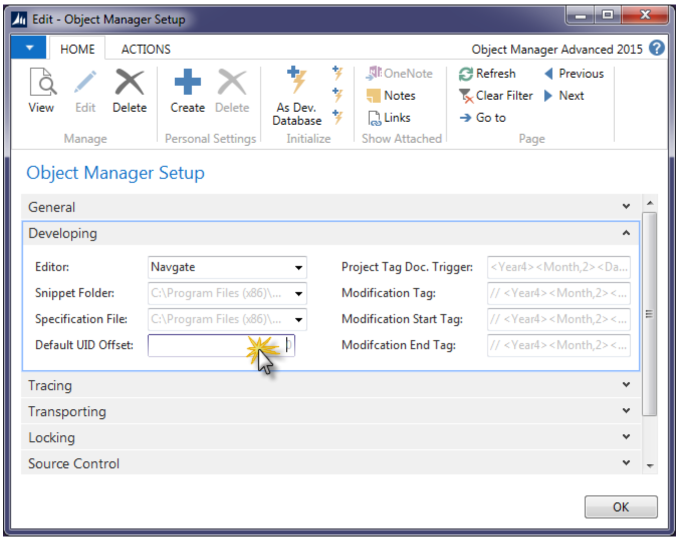
Navgate gives your new element a **Name** and **ID**. This new **ID** is created with the **UidOffset** which is specified in your database, the **Object Manager Setup** or the **Project Type** of the project you are currently working on.



☑ Lock timeout
Timeout Duration (sec) . 10
Start ID (UidOffset) . . . 0

Start ID (UidOffset): 0 L06\reijer INS

Figure 10-3: UidOffset in the Database



Object Manager Setup

General

Developing

Editor: Navgate

Snippet Folder: C:\Program Files (x86)\...

Specification File: C:\Program Files (x86)\...

Default UID Offset:

Project Tag Doc. Trigger: <Year4><Month,2><Da...

Modification Tag: // <Year4><Month,2><...

Modification Start Tag: // <Year4><Month,2><...

Modifcation End Tag: // <Year4><Month,2><...

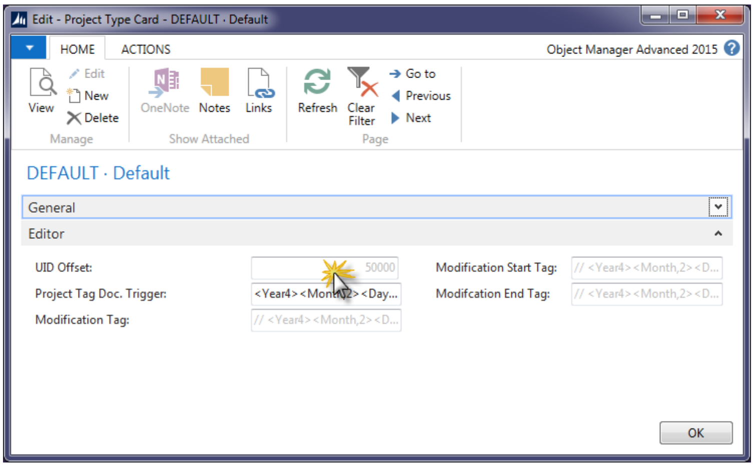
Tracing

Transporting

Locking

Source Control

Figure 10-4: UID Offset in the Object Manager Setup



DEFAULT · Default

General

Editor

UID Offset: 50000

Project Tag Doc. Trigger: <Year4><Month,2><Day...

Modification Tag: // <Year4><Month,2><D...

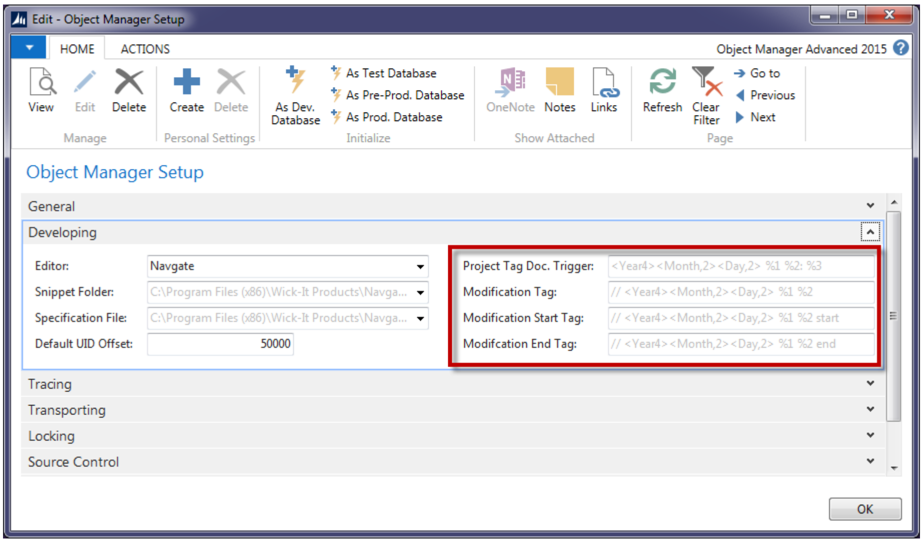
Modification Start Tag: // <Year4><Month,2><D...

Modifcation End Tag: // <Year4><Month,2><D...

Figure 10-5: UID Offset on the Project Type

11 COMMENTING

Every modification that you make on **NAV** objects must be tagged so other developers can see what is changed. **Navgate** has several commenting templates that you can use to speed up such tagging of modifications. These templates can be setup in the **Object Manager Setup** or on the **Project Type** that you are currently working on.



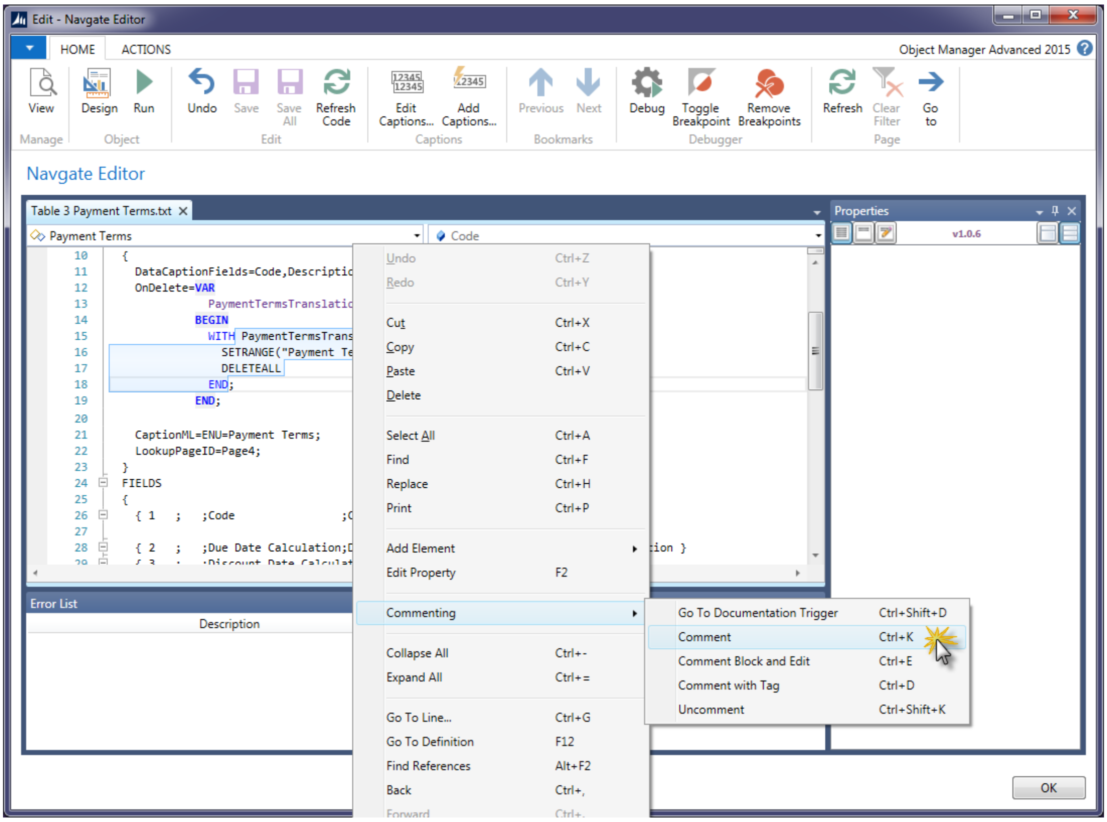
Editor: Navgate
Snippet Folder: C:\Program Files (x86)\Wick-It Products\Navga...
Specification File: C:\Program Files (x86)\Wick-It Products\Navga...
Default UID Offset: 50000
Project Tag Doc. Trigger: <Year4><Month,2><Day,2> %1 %2: %3
Modification Tag: // <Year4><Month,2><Day,2> %1 %2
Modification Start Tag: // <Year4><Month,2><Day,2> %1 %2 start
Modifcation End Tag: // <Year4><Month,2><Day,2> %1 %2 end

Figure 11-1: Comment Tags in the Object Manager Setup

In the templates you can use date expressions and the following placeholders:

- **%1** for the project number
- **%2** for the active user's initials
- **%3** for the description of the modification. This setting is not applicable for **Navgate**. This is used if you assign a modification to a project in the **Object Manager**.

All commenting functions are listed in the context menu.



```
{
  DataCaptionFields=Code,Descriptic
  OnDelete=VAR
             PaymentTermsTranslatic
           BEGIN
             WITH PaymentTermsTrans
               SETRANGE("Payment Te
               DELETEALL
             END;
           END;

  CaptionML=ENU=Payment Terms;
  LookupPageID=Page4;
}
FIELDS
{
  { 1   ;   ;Code                 ;C

  { 2   ;   ;Due Date Calculation;D
```

Figure 11-2: Comment Functions in Context Menu

11.1 COMMENT

The **Comment** option is the simplest one. It just adds // at the beginning of the code lines like this:

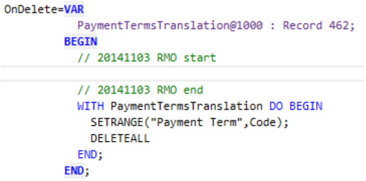
```
OnDelete=VAR
           PaymentTermsTranslation@1000 : Record 462;
         BEGIN
           // WITH PaymentTermsTranslation DO BEGIN
           //   SETRANGE("Payment Term",Code);
           //   DELETEALL
           // END;
         END;
```

11.2 COMMENT BLOCK AND EDIT

If you want to keep the original code to let other developers know what you have modified you can use the **Comment Block and Edit** option. It will comment the original code and makes a copy where you can do your modifications.

```
OnDelete=VAR
           PaymentTermsTranslation@1000 : Record 462;
         BEGIN
           // 20141103 RMO start
           // WITH PaymentTermsTranslation DO BEGIN
           //   SETRANGE("Payment Term",Code);
           //   DELETEALL
           // END;
           WITH PaymentTermsTranslation DO BEGIN
             SETRANGE("Payment Term",Code);
             DELETEALL
           END;
           // 20141103 RMO end
         END;
```

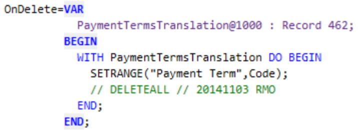
If you use this option on an empty line it will surround your code with an opening and closing tag.



```
OnDelete=VAR
           PaymentTermsTranslation@1000 : Record 462;
         BEGIN
           // 20141103 RMO start

           // 20141103 RMO end
           WITH PaymentTermsTranslation DO BEGIN
             SETRANGE("Payment Term",Code);
             DELETEALL
           END;
         END;
```

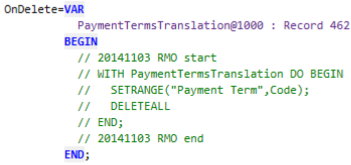
11.3 COMMENT WITH TAG

To disable code and leave a tag you can use the **Comment with Tag** option. Applying this on a single line results in the following:



```
OnDelete=VAR
           PaymentTermsTranslation@1000 : Record 462;
         BEGIN
           WITH PaymentTermsTranslation DO BEGIN
             SETRANGE("Payment Term",Code);
             // DELETEALL // 20141103 RMO
           END;
         END;
```

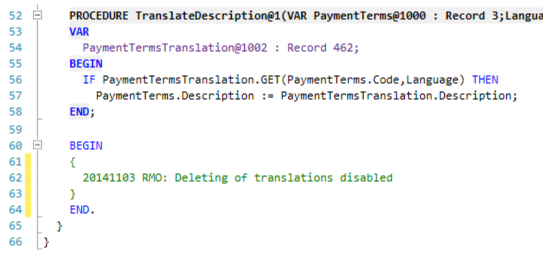
Applying on a selection of code gives you the following result:



```
OnDelete=VAR
           PaymentTermsTranslation@1000 : Record 462;
         BEGIN
           // 20141103 RMO start
           // WITH PaymentTermsTranslation DO BEGIN
           //   SETRANGE("Payment Term",Code);
           //   DELETEALL
           // END;
           // 20141103 RMO end
         END;
```

11.4 ADD COMMENT TO DOCUMENT TRIGGER

It is also good practice to leave a comment in the documentation trigger describing what you have changed in the object. You can jump to the documentation trigger with the shortcut **Ctrl+Shift+D**. Adding a comment can be done with shortcut **Ctrl+D**.



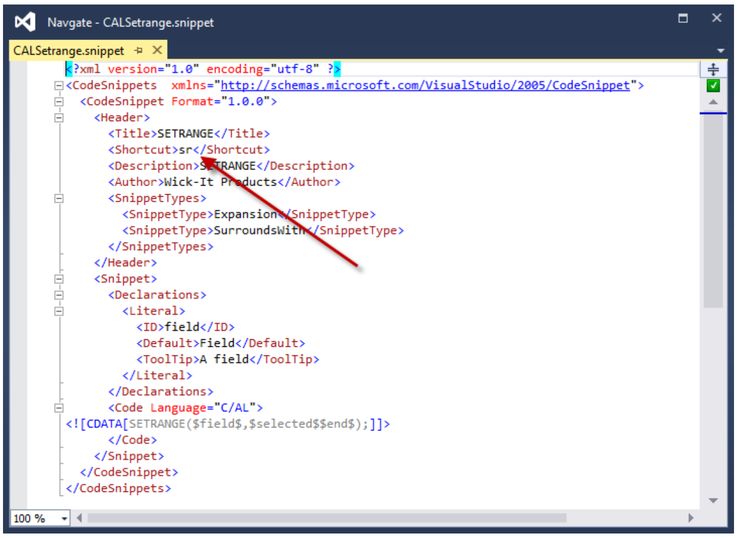
```
    PROCEDURE TranslateDescription@1(VAR PaymentTerms@1000 : Record 3;Langua
    VAR
      PaymentTermsTranslation@1002 : Record 462;
    BEGIN
      IF PaymentTermsTranslation.GET(PaymentTerms.Code,Language) THEN
        PaymentTerms.Description := PaymentTermsTranslation.Description;
    END;

    BEGIN
    {
      20141103 RMO: Deleting of translations disabled
    }
    END.
  }
}
```

12 CODE SNIPPETS

Code snippets are predefined lines of code that can be inserted in the text editor. They are used to automate repetitive typing. You can create your own code snippets or use the snippets that are installed with **Navgate** by default in the following folder:

C:\Program Files (x86)\Wick-It Products\Navgate\CALSnippets

A code snippet can be inserted by typing its shortcut followed by the **TAB** key. Each snippet has its own shortcut defined in the xml of the snippet.



```
<?xml version="1.0" encoding="utf-8" ?>
<CodeSnippets  xmlns="http://schemas.microsoft.com/VisualStudio/2005/CodeSnippet">
  <CodeSnippet Format="1.0.0">
    <Header>
      <Title>SETRANGE</Title>
      <Shortcut>sr</Shortcut>
      <Description>SETRANGE</Description>
      <Author>Wick-It Products</Author>
      <SnippetTypes>
        <SnippetType>Expansion</SnippetType>
        <SnippetType>SurroundsWith</SnippetType>
      </SnippetTypes>
    </Header>
    <Snippet>
      <Declarations>
        <Literal>
          <ID>field</ID>
          <Default>Field</Default>
          <ToolTip>A field</ToolTip>
        </Literal>
      </Declarations>
      <Code Language="C/AL">
<![CDATA[SETRANGE($field$,$selected$$end$);]]>
      </Code>
    </Snippet>
  </CodeSnippet>
</CodeSnippets>
```

Figure 12-1: Setrange Code Snippet

More information about creating code snippets can be found on MSDN:
http://msdn.microsoft.com/en-us/library/ms165394.aspx

13 WHERE USED

The **Where Used** tool of **Object Manager** is used to see where objects, fields, triggers, etc. are used. For more information about how to use the **Where Used** functionality see the '**User Guide of Object Manager Advanced**'.

It is possible to show the where used entries in **Navgate** in two ways:

- Open **Navgate** from the **Where Used** tool.
- Place your cursor on a keyword in **Navgate** and invoking the where used function by pressing **Ctrl+Alt+F2**.

13.1 OPEN NAVGATE FROM WHERE USED

Example of use: As a developer you would like to know where the field **Description** of **Table Payment Terms** is used through the database, to do this, open the **Where Used** tool of **Object Manager** and open all where used entries.

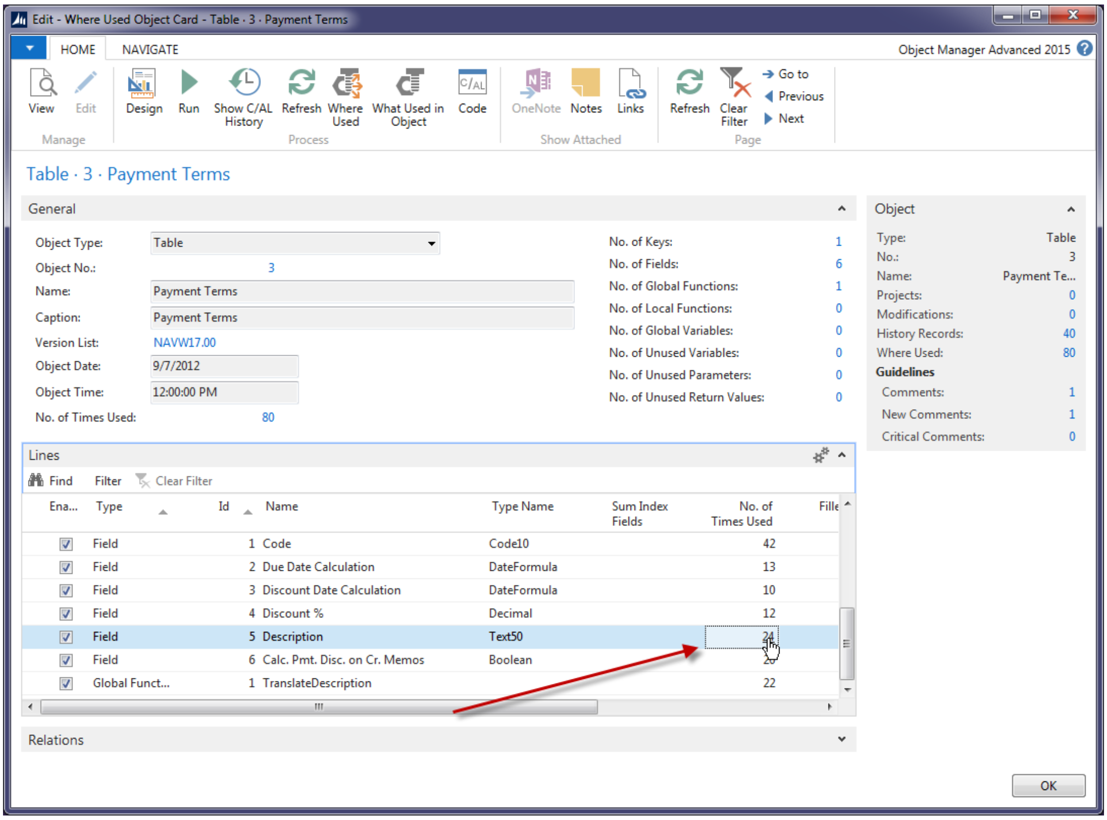


Table · 3 · Payment Terms

General

Object Type: Table
Object No.: 3
Name: Payment Terms
Caption: Payment Terms
Version List: NAVW17.00
Object Date: 9/7/2012
Object Time: 12:00:00 PM
No. of Times Used: 80

No. of Keys: 1
No. of Fields: 6
No. of Global Functions: 1
No. of Local Functions: 0
No. of Global Variables: 0
No. of Unused Variables: 0
No. of Unused Parameters: 0
No. of Unused Return Values: 0

Object

Type: Table
No.: 3
Name: Payment Te...
Projects: 0
Modifications: 0
History Records: 40
Where Used: 80

Guidelines

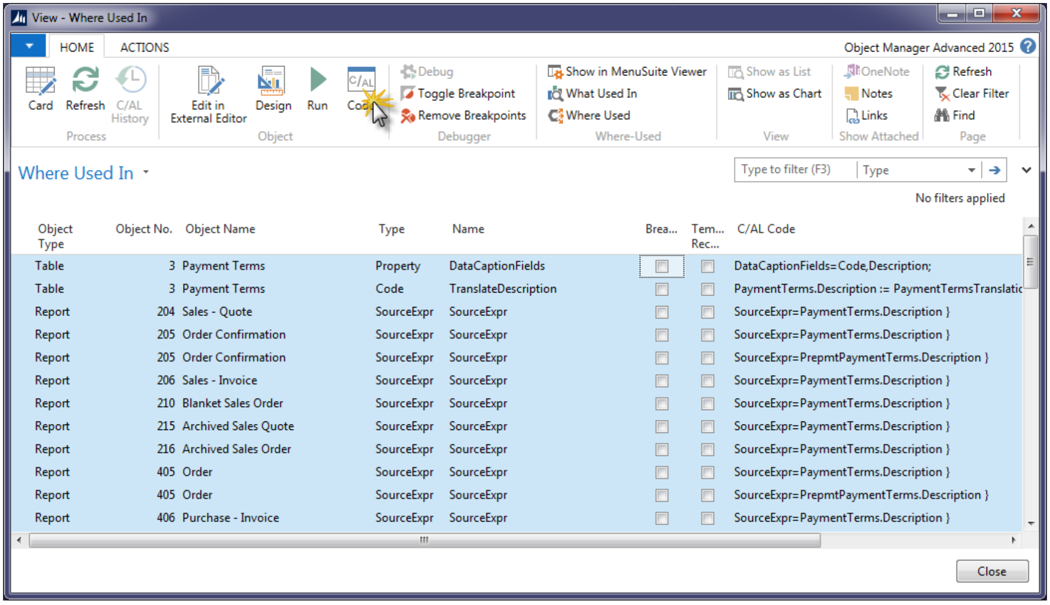
Comments: 1
New Comments: 1
Critical Comments: 0

Lines

Ena...	Type	Id	Name	Type Name	Sum Index Fields	No. of Times Used	Fille
☑	Field	1	Code	Code10		42	
☑	Field	2	Due Date Calculation	DateFormula		13	
☑	Field	3	Discount Date Calculation	DateFormula		10	
☑	Field	4	Discount %	Decimal		12	
☑	Field	5	Description	Text50		24	
☑	Field	6	Calc. Pmt. Disc. on Cr. Memos	Boolean		20	
☑	Global Funct...	1	TranslateDescription			22	

Figure 13-1: Where Used Card of Payment Terms

Select all entries and click the **Code** button to open **Navgate**.

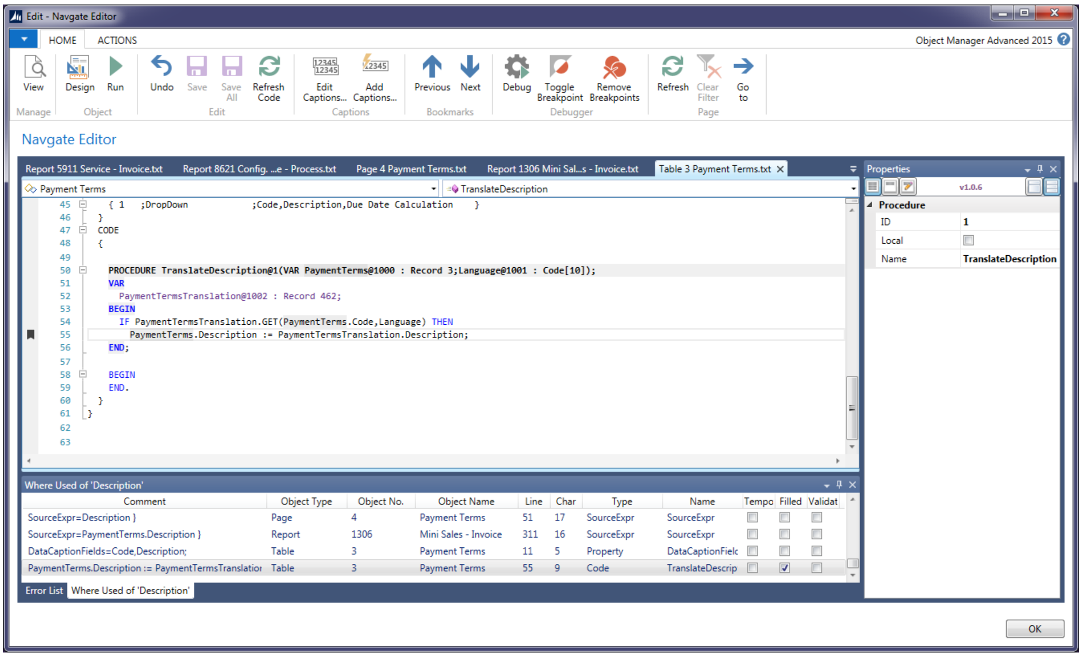


No filters applied

Object Type	Object No.	Object Name	Type	Name	Brea...	Tem... Rec...	C/AL Code
Table	3	Payment Terms	Property	DataCaptionFields	☐	☐	DataCaptionFields=Code,Description;
Table	3	Payment Terms	Code	TranslateDescription	☐	☐	PaymentTerms.Description := PaymentTermsTranslatic
Report	204	Sales - Quote	SourceExpr	SourceExpr	☐	☐	SourceExpr=PaymentTerms.Description }
Report	205	Order Confirmation	SourceExpr	SourceExpr	☐	☐	SourceExpr=PaymentTerms.Description }
Report	205	Order Confirmation	SourceExpr	SourceExpr	☐	☐	SourceExpr=PrepmtPaymentTerms.Description }
Report	206	Sales - Invoice	SourceExpr	SourceExpr	☐	☐	SourceExpr=PaymentTerms.Description }
Report	210	Blanket Sales Order	SourceExpr	SourceExpr	☐	☐	SourceExpr=PaymentTerms.Description }
Report	215	Archived Sales Quote	SourceExpr	SourceExpr	☐	☐	SourceExpr=PaymentTerms.Description }
Report	216	Archived Sales Order	SourceExpr	SourceExpr	☐	☐	SourceExpr=PaymentTerms.Description }
Report	405	Order	SourceExpr	SourceExpr	☐	☐	SourceExpr=PaymentTerms.Description }
Report	405	Order	SourceExpr	SourceExpr	☐	☐	SourceExpr=PrepmtPaymentTerms.Description }
Report	406	Purchase - Invoice	SourceExpr	SourceExpr	☐	☐	SourceExpr=PaymentTerms.Description }

Figure 13-2: Where Used Entries of Field Description

All the objects where the field **Description** is used are opened in **Navgate** and bookmarks are placed in the selection margin. Scrolling through all bookmarks is done with shortcuts **F3** and **Shift+F3** or the bookmark buttons on the ribbon.



```
      { 1   ;DropDown           ;Code,Description,Due Date Calculation    }
    }
    CODE
    {

      PROCEDURE TranslateDescription@1(VAR PaymentTerms@1000 : Record 3;Language@1001 : Code[10]);
      VAR
        PaymentTermsTranslation@1002 : Record 462;
      BEGIN
        IF PaymentTermsTranslation.GET(PaymentTerms.Code,Language) THEN
          PaymentTerms.Description := PaymentTermsTranslation.Description;
      END;

      BEGIN
      END.
    }
  }
```

ID: 1
Local: ☐
Name: TranslateDescription

Where Used of 'Description'

Comment	Object Type	Object No.	Object Name	Line	Char	Type	Name	Tempo	Filled	Validat
SourceExpr=Description }	Page	4	Payment Terms	51	17	SourceExpr	SourceExpr	☐	☐	☐
SourceExpr=PaymentTerms.Description }	Report	1306	Mini Sales - Invoice	311	16	SourceExpr	SourceExpr	☐	☐	☐
DataCaptionFields=Code,Description;	Table	3	Payment Terms	11	5	Property	DataCaptionFielc	☐	☐	☐
PaymentTerms.Description := PaymentTermsTranslatior	Table	3	Payment Terms	55	9	Code	TranslateDescrip	☐	☑	☐

Figure 13-3: Where Used Entries in Navgate

Information about the field use is placed in a new tool window. It contains information like if the field was part of a temporary table and if it was filled and/or validated. If you double click on an entry in the tool window the cursor will move to the right location.

13.2 USE WHERE USED BASE IN NAVGATE

Another way to show the where used entries is to invoke it from **Navgate**. You just have to place your cursor on a keyword and press **Ctrl+Alt+F2** or click the button on the ribbon.

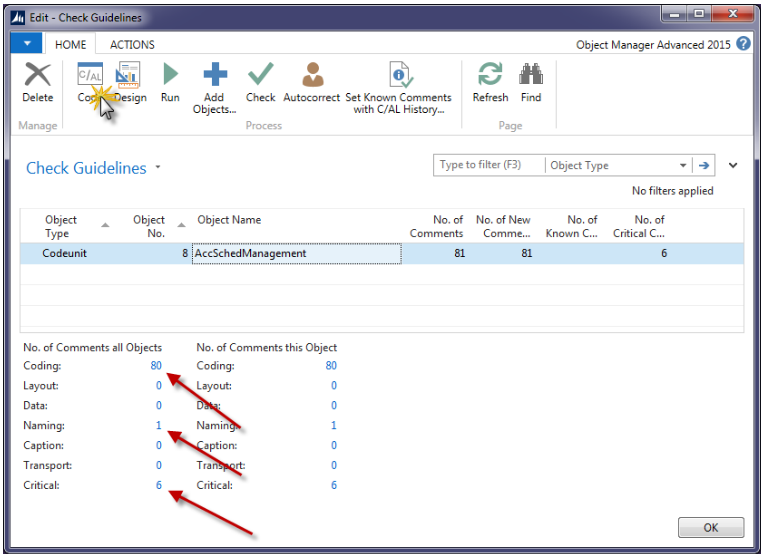
14 CHECK GUIDELINES

Writing code in a way that it’s easy to read by other developers is very important. The **Check Guidelines** of **Object Manager** can help you with doing that. It tracks down many flaws and mistakes and can even correct a lot of them. You can read more about the **Check Guidelines** tool in the ‘**User Guide of Object Manager Advanced**’.

The **Check Guidelines** can be invoked in two ways: from the **Check Guidelines** tool of **Object Manager** or from **Navgate**.

14.1 OPEN NAVGATE FROM CHECK GUIDELINES TOOL

Example of use: As a developer you want to improve the quality of **Codeunit 8: AccSchedManagement** we open the **Check Guidelines** tool, add the object and press **F9** to analyze the object.

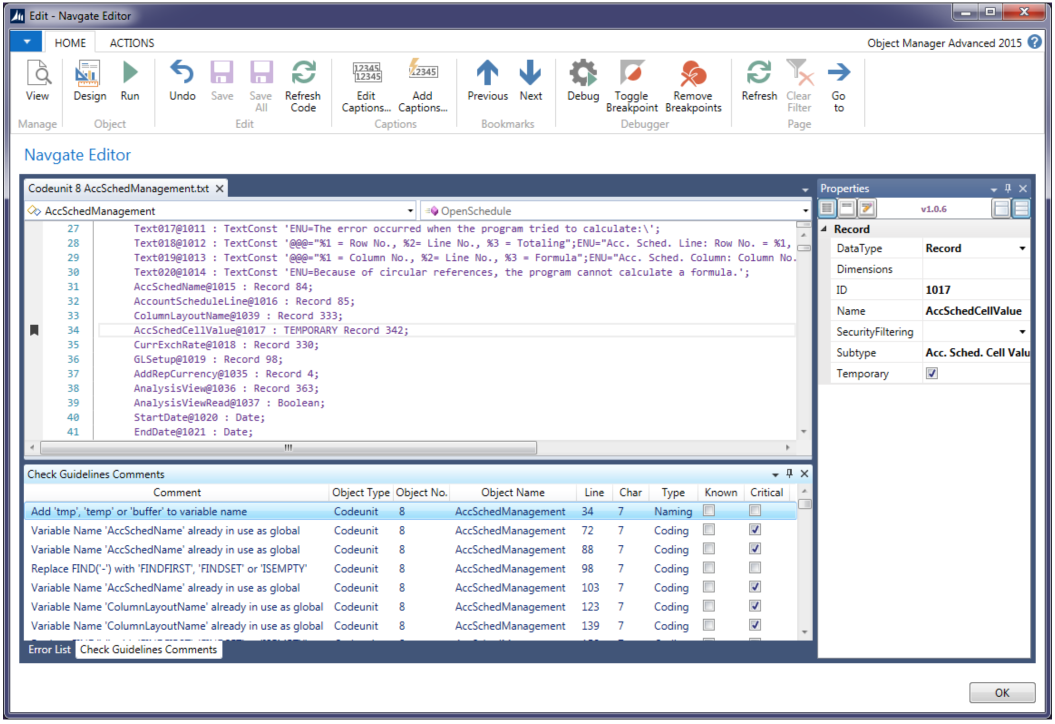


Object Type	Object No.	Object Name	No. of Comments	No. of New Comme...	No. of Known C...	No. of Critical C...
Codeunit	8	AccSchedManagement	81	81		6

No. of Comments all Objects		No. of Comments this Object	
Coding:	80	Coding:	80
Layout:	0	Layout:	0
Data:	0	Data:	0
Naming:	1	Naming:	1
Caption:	0	Caption:	0
Transport:	0	Transport:	0
Critical:	6	Critical:	6

Figure 14-1: Check Guidelines Comments of Codeunit 8: AccSchedManagement

There are 80 possible code improvements, 1 naming problem and 6 critical problems. To see the comments in **Navgate** press the **Code** button.



```
Text017@1011 : TextConst 'ENU=The error occurred when the program tried to calculate:\';
Text018@1012 : TextConst '@@@="%1 = Row No., %2= Line No., %3 = Totaling";ENU="Acc. Sched. Line: Row No. = %1,
Text019@1013 : TextConst '@@@="%1 = Column No., %2= Line No., %3 = Formula";ENU="Acc. Sched. Column: Column No.
Text020@1014 : TextConst 'ENU=Because of circular references, the program cannot calculate a formula.';
AccSchedName@1015 : Record 84;
AccountScheduleLine@1016 : Record 85;
ColumnLayoutName@1039 : Record 333;
AccSchedCellValue@1017 : TEMPORARY Record 342;
CurrExchRate@1018 : Record 330;
GLSetup@1019 : Record 98;
AddRepCurrency@1035 : Record 4;
AnalysisView@1036 : Record 363;
AnalysisViewRead@1037 : Boolean;
StartDate@1020 : Date;
EndDate@1021 : Date;
```

Record
DataType: Record
Dimensions:
ID: 1017
Name: AccSchedCellValue
SecurityFiltering:
Subtype: Acc. Sched. Cell Valu
Temporary: ☑

Check Guidelines Comments

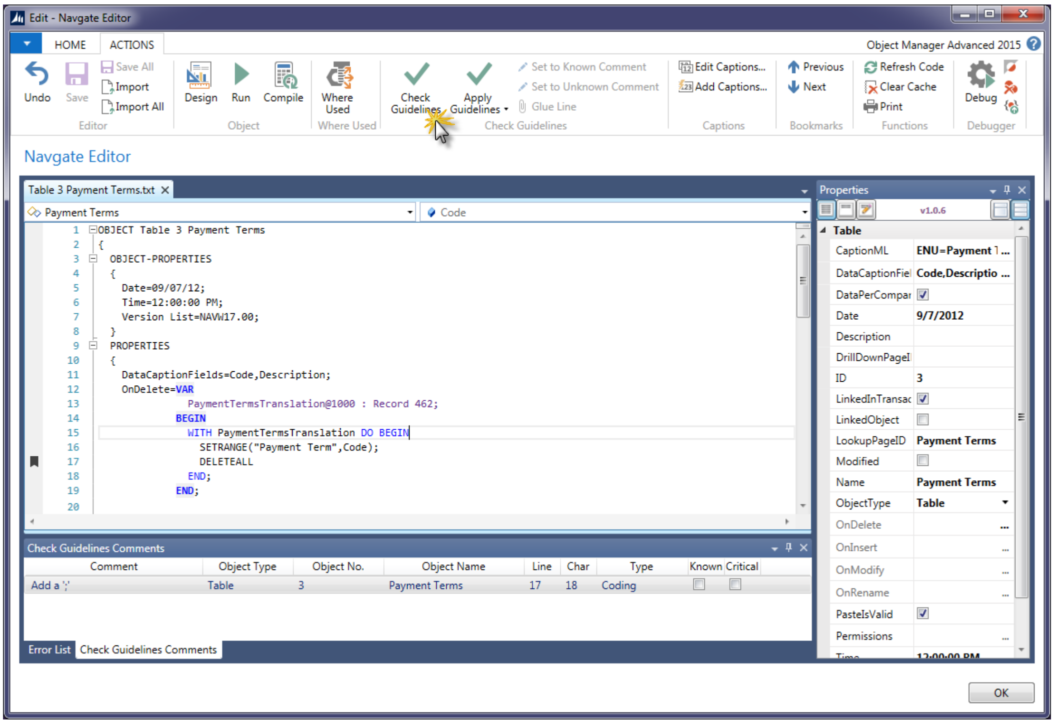
Comment	Object Type	Object No.	Object Name	Line	Char	Type	Known	Critical
Add 'tmp', 'temp' or 'buffer' to variable name	Codeunit	8	AccSchedManagement	34	7	Naming	☐	☐
Variable Name 'AccSchedName' already in use as global	Codeunit	8	AccSchedManagement	72	7	Coding	☐	☑
Variable Name 'AccSchedName' already in use as global	Codeunit	8	AccSchedManagement	88	7	Coding	☐	☑
Replace FIND('-') with 'FINDFIRST', 'FINDSET' or 'ISEMPTY'	Codeunit	8	AccSchedManagement	98	7	Coding	☐	☐
Variable Name 'AccSchedName' already in use as global	Codeunit	8	AccSchedManagement	103	7	Coding	☐	☑
Variable Name 'ColumnLayoutName' already in use as global	Codeunit	8	AccSchedManagement	123	7	Coding	☐	☑
Variable Name 'ColumnLayoutName' already in use as global	Codeunit	8	AccSchedManagement	139	7	Coding	☐	☑

Figure 14-2: Check Guidelines Comments in Navgate

A tool window is opened in the bottom of **Navgate** where you can see all comments. If you double click on an entry you jump right to the code line in the text editor.

14.2 ANALYZE CODE IN NAVGATE

It is also possible to check the object you are working on by clicking the **Check Guidelines** button on the ribbon.



```
OBJECT Table 3 Payment Terms
{
  OBJECT-PROPERTIES
  {
    Date=09/07/12;
    Time=12:00:00 PM;
    Version List=NAVW17.00;
  }
  PROPERTIES
  {
    DataCaptionFields=Code,Description;
    OnDelete=VAR
               PaymentTermsTranslation@1000 : Record 462;
             BEGIN
               WITH PaymentTermsTranslation DO BEGIN
                 SETRANGE("Payment Term",Code);
                 DELETEALL
               END;
             END;

```

Check Guidelines Comments

Comment	Object Type	Object No.	Object Name	Line	Char	Type	Known	Critical
Add a ';'	Table	3	Payment Terms	17	18	Coding	☐	☐

Table

CaptionML	ENU=Payment T ...
DataCaptionFiel	Code,Descriptio ...
DataPerCompar	☑
Date	9/7/2012
Description	
DrillDownPageI	
ID	3
LinkedInTransac	☑
LinkedObject	☐
LookupPageID	Payment Terms
Modified	☐
Name	Payment Terms
ObjectType	Table
OnDelete	...
OnInsert	...
OnModify	...
OnRename	...
PasteIsValid	☑
Permissions	...
Time	12:00:00 PM

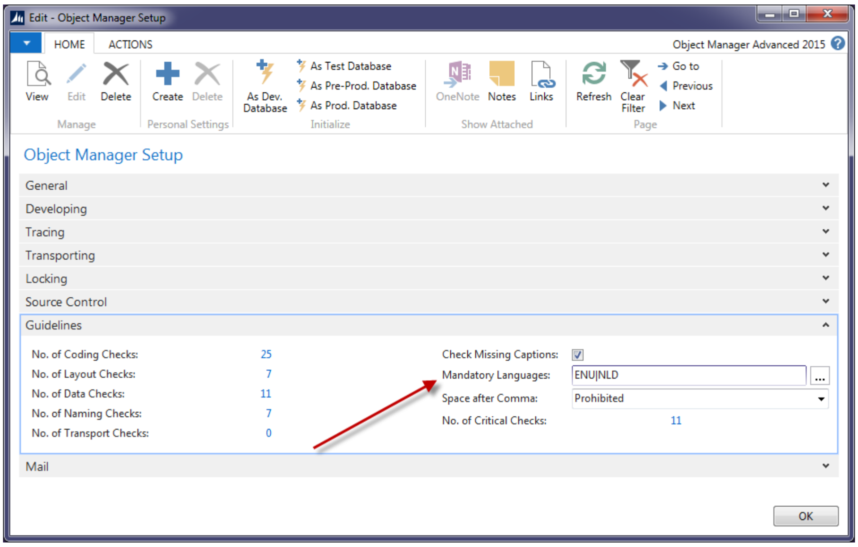
Figure 14-3: Analyze Code in Navgate

14.3 AUTO APPLY GUIDELINES

To let **Object Manager** auto correct the object press the **Apply Guidelines** button. All comments that can be corrected will be applied. In the above example the semicolon will be added to the **DELETEALL** statement.

15 EDIT CAPTIONS

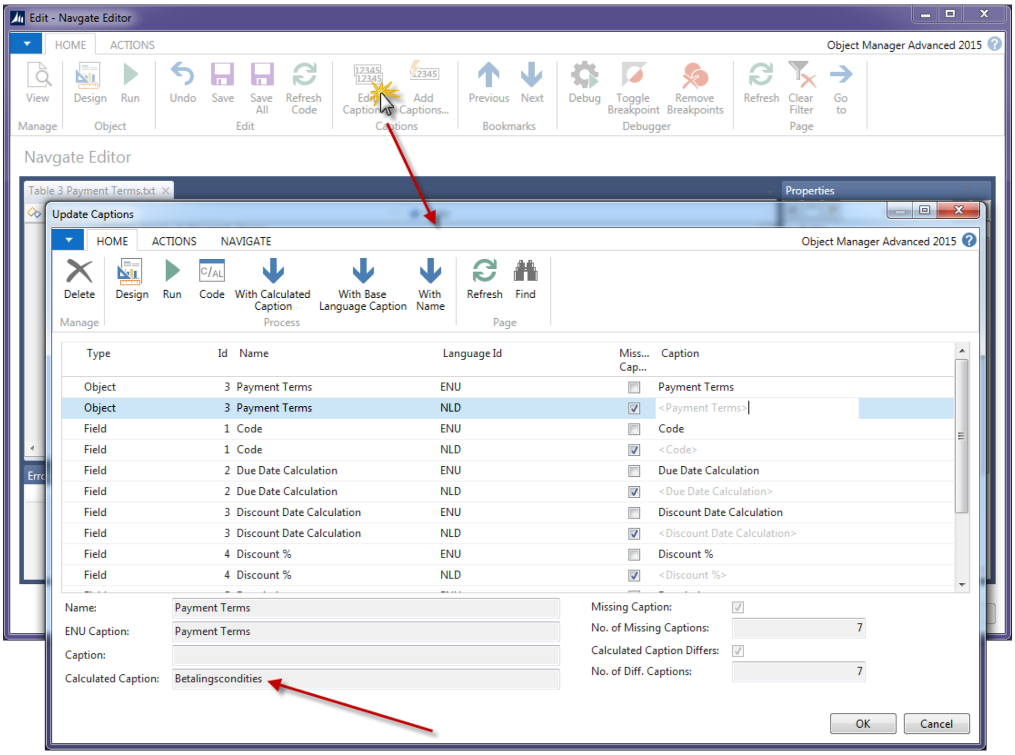
The translation tool of **Object Manager** can be used if you want to add or edit your captions. The languages that can be edited are set in the **Object Manager Setup**.



No. of Coding Checks: 25
No. of Layout Checks: 7
No. of Data Checks: 11
No. of Naming Checks: 7
No. of Transport Checks: 0
Check Missing Captions: ☑
Mandatory Languages: ENU|NLD
Space after Comma: Prohibited
No. of Critical Checks: 11

Figure 15-1: Mandatory Languages in Object Manager Setup

Press the **Edit Captions** button on the ribbon. If the **Translation Tool** has rules about a translation you see the result in the **Calculated Caption** and it can be inserted with the **With Calculated Caption** on the ribbon.



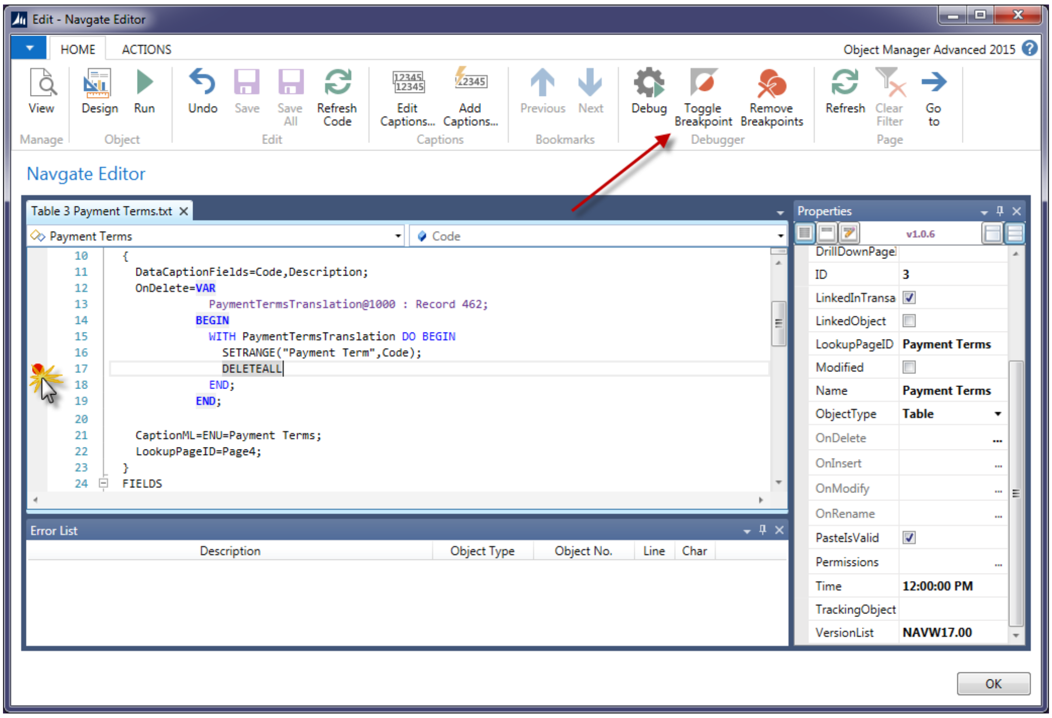
Type	Id	Name	Language Id	Miss... Cap...	Caption
Object	3	Payment Terms	ENU	☐	Payment Terms
Object	3	Payment Terms	NLD	☑	<Payment Terms>
Field	1	Code	ENU	☐	Code
Field	1	Code	NLD	☑	<Code>
Field	2	Due Date Calculation	ENU	☐	Due Date Calculation
Field	2	Due Date Calculation	NLD	☑	<Due Date Calculation>
Field	3	Discount Date Calculation	ENU	☐	Discount Date Calculation
Field	3	Discount Date Calculation	NLD	☑	<Discount Date Calculation>
Field	4	Discount %	ENU	☐	Discount %
Field	4	Discount %	NLD	☑	<Discount %>

Name: Payment Terms
ENU Caption: Payment Terms
Caption:
Calculated Caption: Betalingscondities
Missing Caption: ☑
No. of Missing Captions: 7
Calculated Caption Differs: ☑
No. of Diff. Captions: 7

Figure 15-2: Edit Captions

16 DEBUGGING

All the debugging functions can be used in **Navgate**. Click in the selection margin of the text editor to add or remove a breakpoint. The same happens if you press **F9** or click the **Toggle Breakpoint** button on the ribbon.

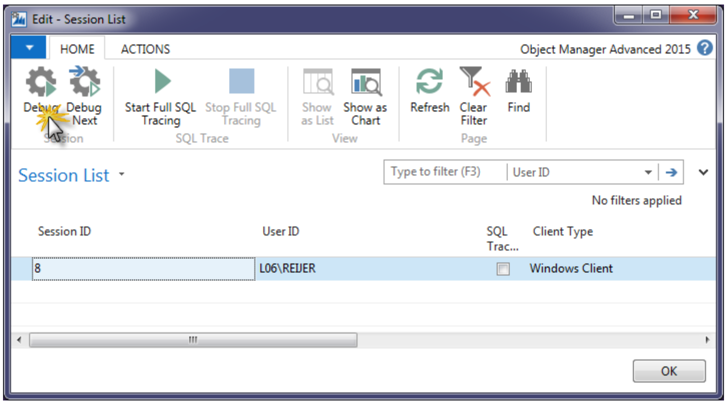


```
{
  DataCaptionFields=Code,Description;
  OnDelete=VAR
             PaymentTermsTranslation@1000 : Record 462;
           BEGIN
             WITH PaymentTermsTranslation DO BEGIN
               SETRANGE("Payment Term",Code);
               DELETEALL
             END;
           END;

  CaptionML=ENU=Payment Terms;
  LookupPageID=Page4;
}
FIELDS
```

DrillDownPageI	
ID	3
LinkedInTransa	☑
LinkedObject	☐
LookupPageID	Payment Terms
Modified	☐
Name	Payment Terms
ObjectType	Table
OnDelete	...
OnInsert	...
OnModify	...
OnRename	...
PasteIsValid	☑
Permissions	...
Time	12:00:00 PM
TrackingObject	
VersionList	NAVW17.00

Figure 16-1: Settings a Breakpoint

The **Debug** button opens the debugger **Session List**.



Session ID	User ID	SQL Trac...	Client Type
8	L06\REIJER	☐	Windows Client

Figure 16-2: Session List

17 SHORTCUTS

Object	
Ctrl + S	Save
Ctrl + Shift + S	Save All
Ctrl + M	Import
Ctrl + Shift + M	Import All
Ctrl + F2	Design Object
Ctrl + R	Run Object
Debugger	
F9	Toggle Breakpoint
Ctrl + Shift + F9	Remove all Breakpoints
Commenting	
Ctrl + K	Comment
Ctrl + E	Comment Block and Edit
Ctrl + D	Comment with Tag
Ctrl + Shift + K	Uncomment
Clipboard/Undo	
Ctrl + C	Copy to Clipboard
Ctrl + Ins	Copy to Clipboard
Ctrl + X	Cut to Clipboard
Shift + Del	Cut to Clipboard
Ctrl + L	Cut Line to Clipboard
Ctrl + V	Paste from Clipboard
Shift + Ins	Paste from Clipboard
Ctrl + Y	Redo
Ctrl + Shift + Z	Redo
Ctrl + Z	Undo
Deletion	
Backspace	Backspace
Shift + Backspace	Backspace
Ctrl + Backspace	Backspace to Previous Word
Del	Delete
Ctrl + Shift + L	Delete Line
Ctrl + Del	Delete to Next Word
Insertion	
Enter	Insert Line Break
Shift + Enter	Insert Line Break

Ctrl + Enter	Open Line Above
Ctrl + Shift + Enter	Open Line Below
IntelliPrompt	
Ctrl + Space	Auto Complete
Ctrl + Shift + Space	Parameter Info
Miscellaneous	
Ctrl + F	Find
Ctrl + H	Replace
Ctrl + P	Print
Ctrl + N	Add Element
F4	Properties
F2	Edit Property
Ctrl + -	Collapse All
Ctrl + =	Expand All
Alt + F2	Find References
Ctrl + Shift + F2	Where Used
Ctrl + Shift + F5	Reset Windows Positions
F1	Help
Ctrl + Q	Edit Captions
Ctrl + Shift + Q	Add Captions
Ctrl + F3	Insert Snippet
Tab	Insert Tab Stop or Indent
Ctrl + U	Make Lowercase
Ctrl + Shift + U	Make Uppercase
Shift + Tab	Remove Tab Stop or Outdent
Insert	Toggle Overwrite Mode
Ctrl + T	Transpose Characters
Ctrl + Shift + T	Transpose Words
Shift + Alt + T	Transpose Lines
Movement	
Down	Move Down
Up	Move Up
Left	Move Left
Right	Move Right
Ctrl + Left	Move to Previous Word
Ctrl + Right	Move to Next Word
Home	Move to Line Start
End	Move to Line End
Ctrl + Home	Move to Document Start
Ctrl + End	Move to Document End
Page Up	Move Page Up

Page Down	Move Page Down
Ctrl + Page Up	Move to Visible Top
Ctrl + Page Down	Move to Visible Bottom
Ctrl +]	Move to Matching Bracket
Ctrl + ,	Back in Navigation Log
Ctrl + .	Forward in Navigation Log
Ctrl + Shift + D	Go to Documentation Trigger
Ctrl + G	Go To Line
F12	Go To Definition
Shift F3	Previous Bookmark
F3	Next Bookmark
Scroll	
Ctrl + Down	Scroll Down
Ctrl + Up	Scroll Up
Selection	
Shift + Down	Select Down
Shift + Up	Select Up
Shift + Left	Select Left
Shift + Right	Select Right
Ctrl + Shift + Left	Select to Previous Word
Ctrl + Shift + Right	Select Block to Next Word
Shift + Home	Select to Line Start
Shift + End	Select to Line End
Ctrl + Shift + Home	Select to Document Start
Ctrl + Shift + End	Select to Document End
Shift + Page Up	Select Page Up
Shift + Page Down	Select Page Down
Ctrl + Shift + Page Up	Select to Visible Top
Ctrl + Shift + Page Down	Select to Visible Bottom
Ctrl + A	Select All
Ctrl + Shift + W	Select Word
Ctrl + Shift +]	Select to Matching Bracket
Shift + Alt + Down	Select Block Down
Shift + Alt + Up	Select Block Up
Shift + Alt + Left	Select Block Left
Shift + Alt + Right	Select Block Right
Ctrl + Shift + Alt + Left	Select Block to Previous Word
Ctrl + Shift + Alt + Right	Select Block to Next Word