



Microsoft Dynamics NAV Source Protection add-on

1. Purpose

Microsoft Dynamics NAV Source Protection add-on (hereinafter “**add-on**”) is intended for protection of a source code of **Microsoft Dynamics NAV** objects against viewing, copying and modification.

Usual object of **Microsoft Dynamics NAV** contains the source and compiled code at the same time. At removal of the source code the compiled code does not change, which allows the object to keep full working capacity.

2. Features of the technical solution

Add-on is officially registered in **Microsoft Business Solutions**, for this granule the separate range of numbers – from 21,066,874 to 21,066,883 is selected.

Add-on can be used in any database of **Microsoft Dynamics NAV** (both in native and in SQL format) and work with any objects of **Microsoft Dynamics NAV** (taking into point seven), including with the localized.








Supported versions of **Microsoft Dynamics NAV** - from 3.60 to 2009 inclusively.

The technology of removal a source code from objects is that that the protected object completely saves functionality, and also is fast cached by the client of **Microsoft Dynamics NAV**.

Note: To provide the correct work of **add-on**, trigger **Documentation** in protected object should not begin with an empty line.

3. Types of protected objects

Add-on carries out protection of a source code of the following types of the objects:

	Table
	Form
	Report
	Dataport
	XMLport
	Codeunit
	Page

Only source code of all triggers and functions is removed from the objects. Variables, text constants, function headers remain unchanged.

4. Functionality description

The note: for the version of **Microsoft Dynamics NAV 2009** it is necessary to use the classic client since all operation occurs to objects in it. Therefore the further description of functionality is fair for all supported versions of **Microsoft Dynamics NAV**.

Add-on is very simple in usage, it contains only pair of user's forms and some buttons and fields in them. You can start to use it at once. It is necessary to pay attention only to convenient possibility of a choice of the several objects, which code you wish to protect. About it is written hardly more low.



In the form of **Source Protection Setup** it is necessary to specify the objects which source code must be protected.

It is possible either to add objects one by one, using the fields **Type** and **ID** or to use the function **Add Objects** (**Functions-> Add Objects**):

Type	ID	Name	Date	Time	Version List
Table	81	Gen. Journal Line	12.01.06	12:00:00	NAVW14.00.02,NAVRU4.00.02
Codeunit	80	Sales-Post	12.01.06	12:00:00	NAVW14.00.02,NAVRU4.00

At choosing of the function **Add Objects** the form **Source Protection AllObj** opens where the necessary objects must be marked by means of markers and **OK** pressed:

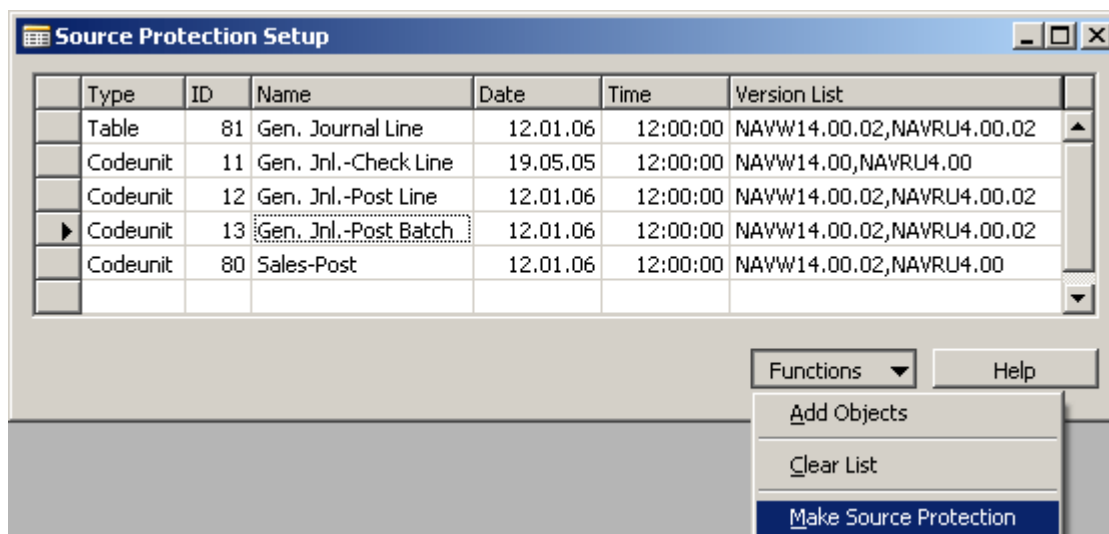
Type	ID	Name	M.	Version List	Date	Time	BLOB...	C..
Codeunit	8	AccSchedManagement		NAVW14.00.02	12.01.06	12:00:00	75980	✓
Codeunit	11	Gen. Jnl.-Check Line		NAVW14.00,NAVRU4.00	19.05.05	12:00:00	35016	✓
Codeunit	12	Gen. Jnl.-Post Line		NAVW14.00.02,NAVRU4.00.02	12.01.06	12:00:00	570152	✓
Codeunit	13	Gen. Jnl.-Post Batch		NAVW14.00.02,NAVRU4.00.02	12.01.06	12:00:00	106956	✓
Codeunit	14	Gen. Jnl.-Show Entries		NAVW14.00	15.09.04	12:00:00	6456	✓
Codeunit	15	Gen. Jnl.-Show Card		NAVW14.00	15.09.04	12:00:00	3036	✓

Marked objects will be added in the form **Source Protection Setup**:

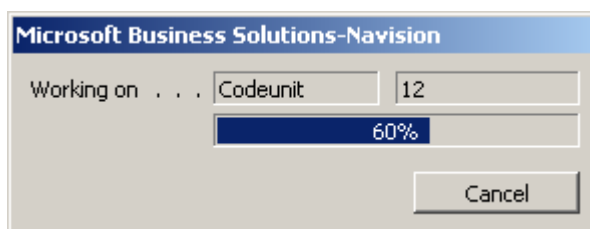
Type	ID	Name	Date	Time	Version List
Table	81	Gen. Journal Line	12.01.06	12:00:00	NAVW14.00.02,NAVRU4.00.02
Codeunit	11	Gen. Jnl.-Check Line	19.05.05	12:00:00	NAVW14.00,NAVRU4.00
Codeunit	12	Gen. Jnl.-Post Line	12.01.06	12:00:00	NAVW14.00.02,NAVRU4.00.02
Codeunit	13	Gen. Jnl.-Post Batch	12.01.06	12:00:00	NAVW14.00.02,NAVRU4.00.02
Codeunit	80	Sales-Post	12.01.06	12:00:00	NAVW14.00.02,NAVRU4.00



To start the process of removal of the source code from the chosen objects it is necessary to press **Make Source Protection (Functions -> Make Source Protection)**:



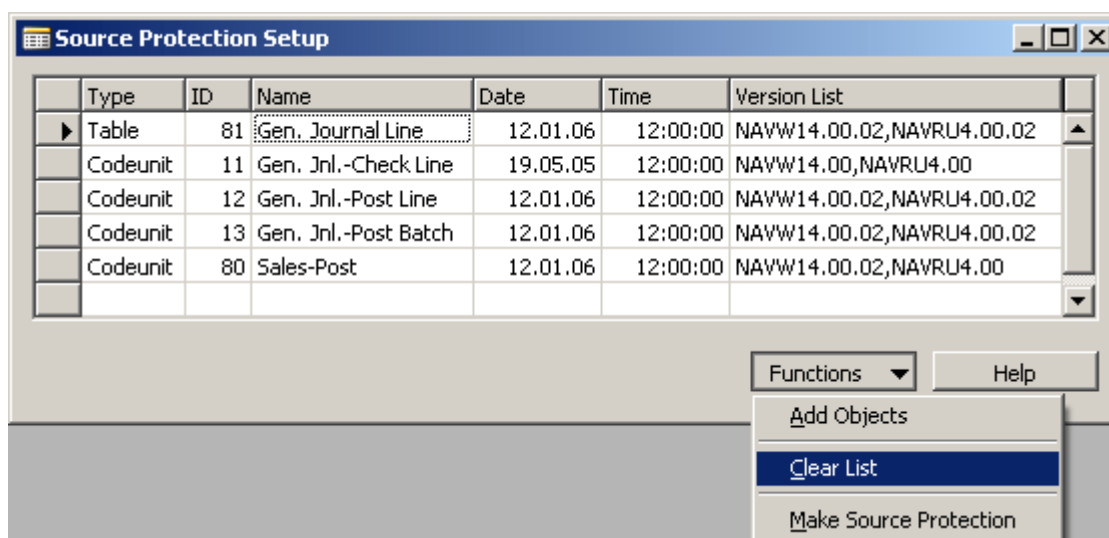
The window will open showing what object is being processed and the general progress of the process:



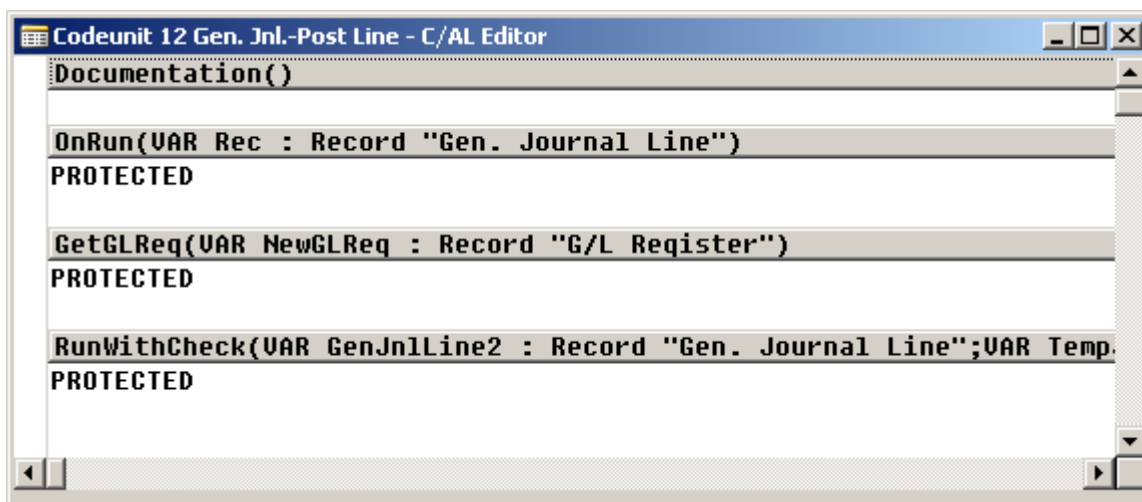
On completion of processing of the objects the above window will be closed.

Removal of the source code from the chosen objects is completed.

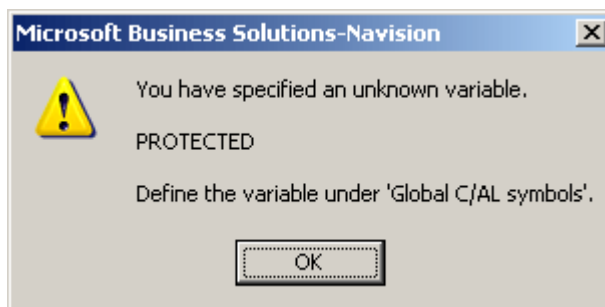
For clearing of the form **Source Protection Setup** it is possible to use the function **Clear List (Functions-> Clear List)**:



After protection the source code of object will look as follows:



The word **PROTECTED** is placed instead of a source code for protection of the object against random compilation. At the attempt to compile the protected object the following message will appear:



The note: in case if the source code of the function or trigger is not sufficient for placing of the word **PROTECTED** - it will not be added.

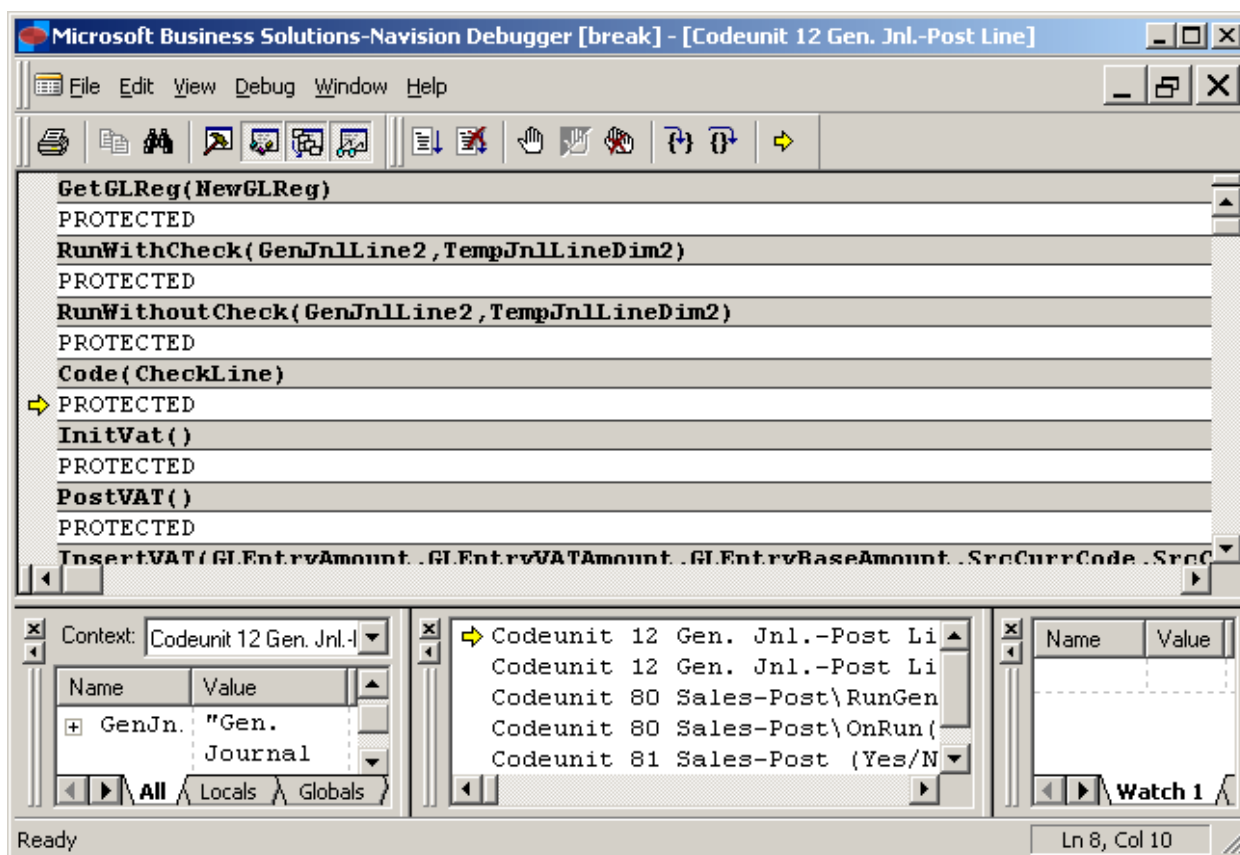
5. An opportunity of the source code restoration

The process of protecting the objects by **add-on** is one-way operation because the source code is physically removed from the objects during protection. This one-way operation provides a high level of protection.

Thus, no one, but you will be able to restore the source code of the objects protected by you.

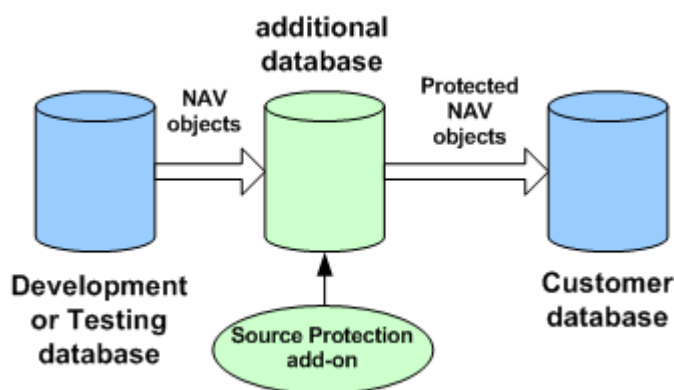
Important: You always have to save "master" copy of the protected objects where source code should be present. Without "master" version you cannot restore the source code of the objects.

When debugging a session and code is not available debugger works good, but source code is inaccessible:



6. Working with protected objects

Let's consider the use of **add-on** by way of the following example:



Add-on should be deployed in the additional database where protection of the objects will take place.

Necessary objects should be updated in additional database before the protection. After protection the objects are transferred to the customer database only in fob files format.

Note: you can export and import a fob file, containing both protected and unprotected objects.

Note: there is no sense to export protected object to the text format, since the source code will not be present in the text file, and the compiled code is not uploaded during exporting to a text format.

If you need to restore a source code in the objects of the customer database, the necessary objects are simply transferred to it from the testing database ("master" copy) where the source code is present.



Thus, all development should be carried out on usual objects, and the copies of the objects should be protected after completion of the development.

7. Requirements to the Microsoft Dynamics NAV license

For the correct work of the **add-on** in your **Microsoft Dynamics NAV** license, in addition to acquired **Microsoft Dynamics NAV Source Protection** granule, there must be the granules providing the reading and modification of **Microsoft Dynamics NAV** objects of the types which you want to protect. It can be one or several of the following granules:

Granule ID	Granule Name
7130	Table Designer
7120, 7150	Form and Page Designer
7110	Report and Dataport Designer
7140	XML Port Designer
7200	Application Builder
7300	Solution Developer

In fact, **add-on** able to secure any **Microsoft Dynamics NAV** object and have only one requirement - it will only secure objects that are readable and modifiable by your developer license.

8. Add-on Licensing

The granule **Microsoft Dynamics NAV Source Protection** is registered at all operational centers of Microsoft Business Solutions.

Add-on is bought once for your developer license. After you have purchased it you can protect any objects (as per p. 7) in any of your databases using your developer license. Thus, once having bought **add-on**, you can use it for all your clients and databases.

When your developer license is being updated, **add-on** granule will be added in it automatically.