

roman's(TI – Translations for Navision 3.60+

Translate any field from any table in Navision 3.60+ with maximal 3 statements. Due to new features like RecordRef, RecordID and GUID it's very easy to install, implement and use. This is THE tool if you want to display data in several languages.

roman's(Translations is a very little (4 Object) add-on to translate descriptions (or any other field) in Navision. For example, Navision is multilanguage but although the complete invoice can be translated, values like "Payment Terms" will always be in your local language. Until now! ;)

.....
Please not this version has been low level tested only. In fact, I've check if the objects compiled ;) Please report bugs to emiel.romein@gmail.com
.....

Objects :

Soort	ID	Naam	Comment
Table	3	Payment Terms	Sample, not needed
Table	59030	Translation Connection	
Table	59031	Translation	
Form	4	Payment Terms	Sample, not needed
Form	59031	Translations	
Codeunit	59030	Translation Mgt.	

Implementation :

roman's(Translations can be integrated into (linked to) any record (table) in Navision. You will only need to code three statements for every field you want to translate.

To create and show translations, code the following in a "Translations" button (or assist button) on a form (or field). In this example form 4, "Payments Terms"

```
TranslationMgt.ShowTranslation(  
    DATABASE::"Payment Terms",  
    GETPOSITION,  
    FIELDNO(Description));
```

To delete translations, code the following in the OnDelete trigger of the table. In this example table 3, "Payments Terms"

```
TranslationMgt.DeleteTranslations(  
    DATABASE::"Payment Terms",  
    GETPOSITION,  
    FIELDNO(Description));
```

To rename translations, code the following in the OnRename trigger of the table. In this example table 3, "Payments Terms"

```
TranslationMgt.RenameTranslations(  
    DATABASE::"Payment Terms",  
    xRec.GETPOSITION,  
    GETPOSITION,  
    FIELDNO(Description));
```

This three statements apply to every table in Navision. Here, for example, the statement to create translations for the "Posting Description" in the "Sales Header"

```
TranslationMgt.ShowTranslation(  
    DATABASE::"Sales Header",  
    GETPOSITION,  
    FIELDNO("Posting Description"));
```

To get (show or print) a specific translation use the following statement. The return value of this statement is the translation or the original value in case no translations exist.

```
TranslationMgt.GetTranslation(  
    DATABASE::"Payment Terms",  
    GETPOSITION,  
    FIELDNO(Description),  
    "Language Code");
```

You could call this statement from the source expression (SourceExpr) of a TextBox on a report or a form.