© Sławek Guzek, sguzek@onet.pl

HOW TO USE IMPORTER

1. HOW IMPORTER WORKS

Importer is a simple tool which allows import data directly to ANY Dynamics NAV table from Excel file.

It uses first non-empty row from selected Excel file as column headers, and assumes that this row contains field captions (as seen on Card/List forms). Using this information Importer tries to build a field map which contains a match between column headers and field captions in selected destination table. Mapping is then presented to the user, and after confirmation is used to import all data in consecutive rows to the destination table.

!!! WARNING !!! Importer is a very powerful tool.You may use it to import data to ANY of NAV table. Importer WILL NOT warn you even if your data kills NAV. It will just read the data straight into the table of your choice. THINK TWICE please before you import anything

2. USING IMPORTER WITHOUT ACCESS TO C/AL CODE

Importer can be used by calling codeunit 99991.After calling Codeunit a dialog window will be Opened asking for selecting source Excel file, and workbook from that file.

Then importer reads a few firs rows from Excel, trying to find parameter definitions. If there is no parameters defined inside selected Excel Importer will ask to select destination table.

After selecting (or reading from parameters from file) destination table Importer will build mapping between column headers and display this map to the user. User can change mapping,can select which individual fields will be imported, on which fields NAV validation routines should be fired during the import. User can also select whether to fire or skip global table insertion validation routines and whether the record should be inserted immediately after filling all required primary key values, or whether it should be inserted upon completion.

Many of parameters can be pre-selected inside Excel file

2.1 WHAT PARAMETERS CAN BE SET INSIDE EXCEL FILE ?

Importer parameters can be set by specifying special parameter keywords with values in one or several cells in a single row, or by using text formatting in column headers. The first found non-empty row is considered as a ‘config’ row if it contains in any cell at least one recognizable parameter.

If there is no recognizable parameters found the first nonempty row is then considered as the header row.

If there is a ‘config’ row inside Excel file then next non-empty row is considered as header row, ora row which is set by a parameter

2.2 PARAMETERS DEFINED BY THE TEXT FORMAT OF HEADER ROW

Two possible options on each individual filed can be set by formatting cells in header row:

BOLD – formatting column header with BOLD tells importer to fire VALIDATE trigger on this field every time the filed will be read from Excel.

ITALIC – formatting column header with ITALIC tells importer to SKIP the entire column during import.

2.3 PARAMETERS DEFINED AS TEXT INSIDE CELLS

Parameter generally is a keyword, optionally followed by double colon and value.

Parameters are always read from first non-empty row containing anyone of all recognizable parameter keywords. When Importer finds first non-empty row it searches all cells from that row and tries to match the format with all recognizable parameter keywords. If Importer finds any one it considers current row as 'config' row.

Parameters are read only from single 'config' row, but may be read from many cells. A few parameters may be also stored in one cell as one long string, separated by semicolons.

Parameters are read from left cell to right, and the last read takes precedence before the first. There no need to put parameters in consecutive cells. There can be another data put in config row - it will be ignored.

Accepted parameters formats are:
1. ParameterID:ParameterValue
2. ParameterID:TargetField=ParameterValue

Syntax 'ParameterID:ParameterValue' is used when parameter defines general behavior for the Importer or for all the fields

Second syntax is used when parameter is applicable at individual field level. That case 'TargetField' can be defined as field caption, or just field number. If target caption contains equation (=) character then you have to use field number as 'TargetField'

Importer recognizes the following list of parameters:

table:<number>
- tells Importer the destination table number

Example:
table:27 – importer will not display destination table selection dialog but just uses table 27 (Item) as destination

importall
- has no additional parameter values, tells Importer to import all columns to which matching field will be found, and ignore settings defined by ITALIC formatting in table header

skiptriggers
- has no additional parameter values. Tells Importer to NOT to fire destination table OnInsert and OnModify triggers

tableheader:<number>
- tells the importer the row number from where Importer starts to search for row interpreted as table header. All rows above given number are skipped, and can contain any data, for example some comments, etc.

Example:
tableheader:6 - importer skip all rows between current row and row number 5 inclusive, and will start to search for header from row 6

const:<field_name>=<constant_value>
const:<field_number>=<constant_value>
- field 'field_name' in every record will be filled with constant_value

Example:

const:description 2=My preset - if the destination table is set to one having field'Description 2' (example Item - 27), field named 'Description 2' in every imported line will be filled in with text 'My preset'

const:5=My preset - in this case parameter table:number parameter setup BEFORE definition (in any cell to on the left) of const parameter. If for example destination table will be set to 27, then in every imported record field number 5 (which is 'Description 2' in case of table 27) will be filled in with text 'My preset'

incr:<field_name>=<start_value>
incr:<field_name>=<start_value>

- field 'field_name' in every record will be filled with value, starting with 'start_value' in first record, and incremented by one in every next record inserted. Very useful for reading into journal tables

Example:

counter:No.=IT0001 - when importing to table 27 field No. in first imported row will have IT0001 value, in second IT0001, and so on

counter:1=IT0001 - in this case parameter table:number parameter setup BEFORE definition (in any cell to on the left) of incr parameter. Then 1 is interpreted as field number of table set by parameter table:number (which is 'No. in case of table 27)

3. USING IMPORTER FROM INSIDE C/AL OBJECT (CALLING IMPORTER CODE)

There are two possible methods of calling importer function from C/AL code.

3.1 BASIC FUNCTION

Basic function requires three parameters.Basis syntax requires minimum parameters and rely on user choices of file, and various options. The Syntax is:

ImportData(tableNo , columnFilter, importOptions) : Integer

tableNo – defines destination table number. If set to 0 Importer tries to read table no from Excel File, from config row.If destination table number cannot be determined from parameters stored inside Excel Importer will open a dialog box asking for selecting destination table

columnFilter – column filter determines filter (in Navision syntax) which columns (strictly – column numbers) from Excel file will be allowed to import. Usually use empty string here, which allows to read and interpret all columns from Excel file, unless you will want top restrict number of columns read. You may want to restrict number of columns read from file to speed up read time
Importer don't allow to import data to flowfields or flowfilters, regardless of list passed in columnFilter parameter.
Remember: When specifying filter remember to include all columns which maps to primary key fields in it !

importOptions – integer value, defining Importer behavior. Tells Importer what to do with data. Detailed description of all possible options is follows next section

3.2 EXTENDED FUNCTION

Extended requires 6 parameters, and allows programmer to have for more control over Importer behavior directly from C/AL Code

ImportDataFromFile(VAR fileName, VAR sheetName, tableNo, headerRow, columnFilter, importOptions)

fileName – text VARIABLE storing source Excel file name . If variable is empty or file does not exist Importer asks for new file, and returns selected filename in this variable.

sheetName - put a TEXT VARIABLE here storing Excel worksheet to import. If variable is empty, or worksheet doesn't exist in Excel file, Importer allows user for select one worksheet and returns user selection in this variable

tableNo - destination table number, meaning as in basic function syntax

headerRow - tells Importer where to find in Excel file row with field captions, names or numbers, or with parameters. The data to import is assumed to be in non empty rows below header.

columnFilter – meaning as in basic function syntax

importOptions - define Importer behavior, tells Importer what to do with data

3.3 IMPORTER RUNTIME OPIONS (ONES THAT CAN BE SET FROM C/AL CODE)

To set option programmatically (during run-time) an 32 bit integer value (standard NAV integer variable) have to be passed to both basic and advanced Importer function calls.

Options mapped to the bits of importOptions variable

d1 d0 – defines whether Importahs to import only new data, update only existing data, or do both.
00 - illegal, Importer will change to 3 - Insert or Update
01 (importOptions=1) – only insert new records
10 (importOptions=2) – only update existing data
11 (importOptions=3) – insert new or update data

when inserting new records insert will be done just after filling all necessary primary key fields, then Importer fills the rest of the record, and then launches MODIFY upon completion. This behaviour may be changed with DelayedInsert option

d2=1 (importOptions=4) - Delayed insert.
when inserting Importer waits with inserting the record upon filling all the fields from Excel

d3=1 (importOptions=8) - import all fields
imports all columns regardless of Excel ITALIC attribute in Excel header

d5 d4 defines what will be used for matching the fields in destination table with Excel headers
00 - match Excel column header to Field Caption
01 (importOptions=16) - match Excel column header to Field Name
1x (importOptions=32) - match Excel column header to Field No.

When bit d4 is set the column headers in the Excel file have to contains field NAMES, which usually are the same as captions in English US language, but beware there are exceptions.

When bit d5 is set the column headers in the Excel file have to contains field NUMBERS. This option (as well as previous) may be useful to prepare international template to import the data regardless of current language set in NAV Client

d7 d6 – determines use of VALIDATE trigger
00 - VALIDATE all filed by default. Ignore BOLD formatting in Excel header
01 (importOptions=64) - don't VALIDATE nothing. Ignore BOLD formatting in Excel header
1x (importOptions=128) - VALIDATE according to the formatting header in Excel (Bold=VALIDATE)

d8=1 (importOptions=256) - don't fire OnInsert/OnModify table triggers

d10=1 (importOptions=1024) - don't allow to change options
user will not be allowed to change options (update, insert, skip OnInsert/OnModify) on UserMapping dialog

d11=1 (importOptions=2048) - don't allow to change mapping
user will not be allowed to add or change destination field mappings

d12=1 (importOptions=4096) - don't allow to change validation
user will not be allowed to choose whether to VALIDATE destination field or not

d31=1 (importOptions=2147483648) - silent mode
no questions, no dialogs, just import based on parameters and excel settings

HOW TO COMBINE VARIOUS OPTIONS ?

Just add their decimal values. For example: if you want Importer to update only fields in destination table you need ImportOption=2, and if you've put field numbers as header row in Excel and wants Importer to try to match it using field numbers you will need to pass ImportOption=32. To have both option set just pass the sum of options' decimal values, like that: importOptions=34 (2+32)