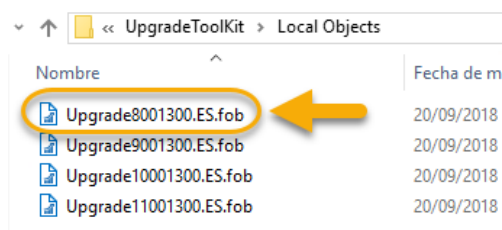


Ejercicio A.13: Upgrade de datos a Business Central on-premise y los desarrollos traspasados a extensiones

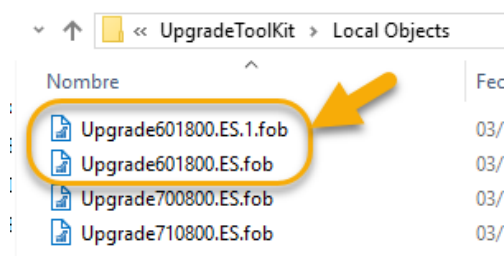
En este ejercicio vamos a ver los pasos a seguir para hacer un Upgrade de datos a BC on-premise donde los desarrollos se han convertido a extensiones.

El UpgradeToolkit que proporciona Microsoft nos permite ejecutar los procesos acumulados de más de una versión. Por ejemplo, en el DVD de Business Central (13) encontramos los siguientes fobs.

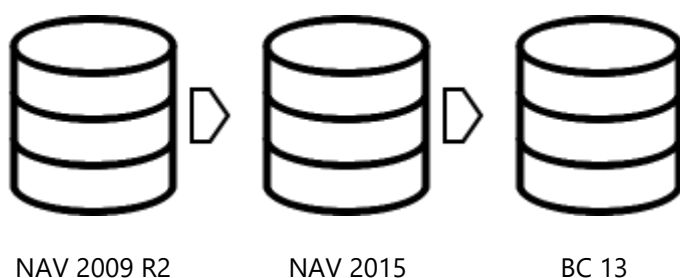


De esta forma podemos hacer un Upgrade a la versión 13 desde la versión 8.

De la misma forma en el DVD de NAV2015 (a partir del CU1), encontramos los objetos necesarios para pasar de NAV 2009 R2 a NAV2015.



Así, para pasar de NAV 2009 R2 a BC 13, deberíamos hacer el upgrade en 3 pasos, y ejecutar los procesos de Upgrade de la Cronus en las siguientes versiones:



Nombre	Versión
NAV 5	5
NAV 2009	6
NAV 2013	7
NAV 2015	8
NAV 2016	9
NAV 2017	10
NAV 2018	11
BC primavera año 2018	12
BC otoño año 2018	13

Para los campos y tablas 50.000 que queramos traspasar a extensiones deberemos:

- En la versión de origen, traspasarlos a tablas temporales
- En la versión de destino, traspasarlos a las extensiones de tabla (a través de una extensión dependiente)

El proceso está explicado en detalle en el **Curso Upgrade de datos a NAV2018** que está disponible en la plataforma ClipDynamics.com

Indicaciones para el ejercicio

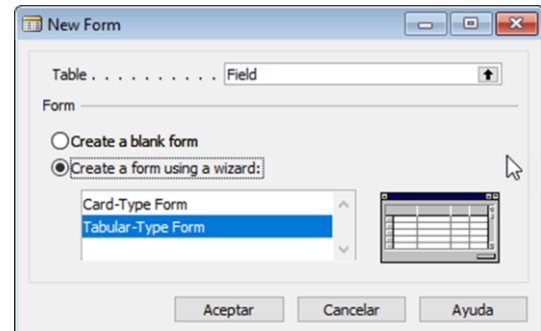
En la versión de origen, crear un proceso en C/SIDE para traspasar los campos 50.000 a tablas temporales.

En la versión de destino, crear un proceso en VSCODE para traspasar los datos a las extensiones de tabla.

Indicaciones paso a paso

Sigue los siguientes pasos en el entorno de desarrollo C/SIDE:

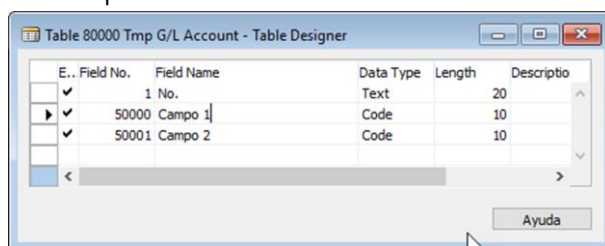
1. Crea un formulario sobre la tabla de sistema field.
Añade todos los campos disponibles.



2. Pon un filtro sobre el campo No. para visualizar todos los campos en el rango 50000..99999.

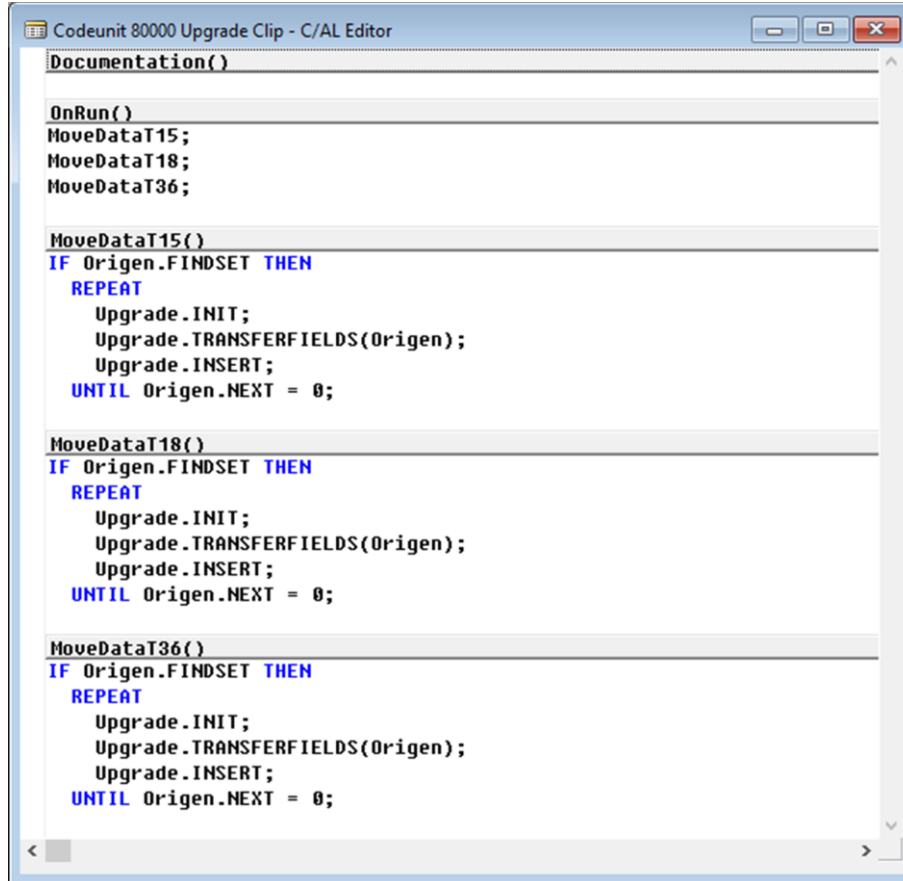
TableNo	No.	TableName	FieldName	T... Len	C... Enabled	Type Name	Field
15	50000	G/L Account	Campo 1	C..	10 N..	Code10	Can
15	50001	G/L Account	Campo 2	C..	10 N..	Code10	Can
18	50000	Customer	Campo 1	C..	10 N..	Code10	Can
18	50001	Customer	Campo 2	C..	10 N..	Code10	Can
36	50000	Sales Header	Campo 1	C..	10 N..	Code10	Can
36	50001	Sales Header	Campo 2	C..	10 N..	Code10	Can

3. Para cada tabla del estándar que contenga campos 50000, crea una tabla temporal que contenga
 - 3.1. Los campos de clave primaria de la tabla origen
 - 3.2. Los campos 50000



Type	ID	Name	Modificado	Lista versiones	Fecha	Hora
Table	15	G/L Account	✓	NAVW 16.00, LIQEM5.00, CACE...	11/12/18	12:40:0
Table	18	Customer	✓	NAVW 16.00.10, LIQEM5.00, N...	11/12/18	12:40:1
Table	36	Sales Header	✓	NAVW 16.00.10, LIQEM6.00.01...	11/12/18	12:40:2
Table	80000	Tmp G/L Account	✓	UPG Clip	11/12/18	12:56:3
Table	80001	Tmp Customer	✓	UPG Clip	11/12/18	12:58:0
Table	80002	Tmp Sales Header	✓	UPG Clip	11/12/18	12:59:3

4. Crear una codeunit de upgrade, para traspasar los datos a las tablas temporales.



```
Codeunit 80000 Upgrade Clip - C/AL Editor

Documentation()

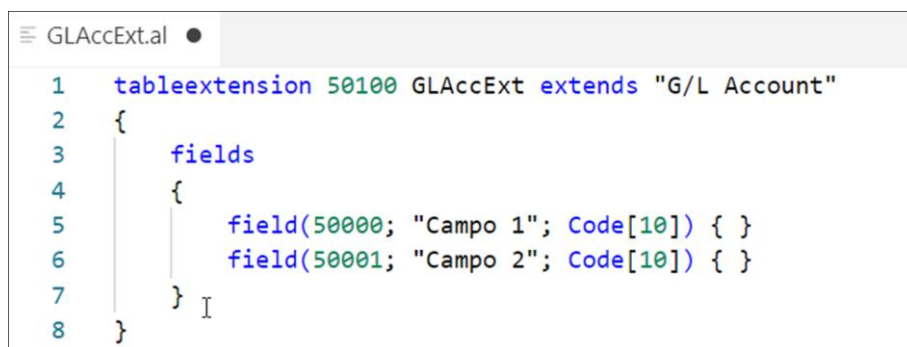
OnRun()
MoveDataT15;
MoveDataT18;
MoveDataT36;

MoveDataT15()
IF Origen.FINDSET THEN
    REPEAT
        Upgrade.INIT;
        Upgrade.TRANSFERFIELDS(Origen);
        Upgrade.INSERT;
    UNTIL Origen.NEXT = 0;

MoveDataT18()
IF Origen.FINDSET THEN
    REPEAT
        Upgrade.INIT;
        Upgrade.TRANSFERFIELDS(Origen);
        Upgrade.INSERT;
    UNTIL Origen.NEXT = 0;

MoveDataT36()
IF Origen.FINDSET THEN
    REPEAT
        Upgrade.INIT;
        Upgrade.TRANSFERFIELDS(Origen);
        Upgrade.INSERT;
    UNTIL Origen.NEXT = 0;
```

Como parte del Upgrade de código, se habrá creado, para cada tabla que contenga campos 50000, una table extensión parecida a esta:



```
GLAccExt.al

1  tableextension 50100 GLAccExt extends "G/L Account"
2  {
3      fields
4      {
5          field(50000; "Campo 1"; Code[10]) { }
6          field(50001; "Campo 2"; Code[10]) { }
7      }
8  }
```

Se debe crear una codeunit para traspasar los datos que hay en las tablas temporales, a los campos de la extensión:

```
codeunit 50101 UpgradeMgt
{
    trigger OnRun()
    begin
    end;

    local procedure MoveDataT15()
    var
        Origen: Record "G/L Account";
        Upgrade: Record "Tmp G/L Account";
    begin
        if Upgrade.FindSet() then
            repeat
                Origen.get(Upgrade."No.");
                Origen."Campo 1" := Upgrade."Campo 1";
                Origen."Campo 2" := Upgrade."Campo 2";
                Origen.Modify();
            until Upgrade.next = 0;
        end;
    end;
}
```

A tener en cuenta:

- Es recomendable crear el proceso de Upgrade como una extensión separada de la extensión que contiene el código. De esta forma, se podrá desinstalar la extensión de Upgrade una vez finalizado el proceso.
La extensión de Upgrade deberá tener una dependencia con la extensión que contiene el código.
- Las tablas temporales donde están los datos 50000 se encuentran en C/SIDE.
Se deben generar los símbolos para que las tablas sean visibles desde vscode.