Sommario

[CALCFIELDS 2](#_Toc45637493)

[Es.: CompanyInfo.CALCFIELDS(Picture); 2](#_Toc45637494)

[SETRANGE Function Record 3](#_Toc45637495)

[SETRANGE Function FieldRef 3](#_Toc45637496)

[0DT 3](#_Toc45637497)

[Data e Ora corrente 3](#_Toc45637498)

[Test sulla Data indefinita (01-01-1753 00:00:00.000) 3](#_Toc45637499)

[Data ultimo giorno dell’anno della data indicata 3](#_Toc45637500)

[Data primo giorno dell’anno della data indicata 3](#_Toc45637501)

[FIND 3](#_Toc45637502)

[Calcolo giorni tra due date 4](#_Toc45637503)

[FINDSET 4](#_Toc45637504)

[FINDFIRST 4](#_Toc45637505)

[FINDLAST 4](#_Toc45637506)

[ISEMPTY 4](#_Toc45637507)

[Struttura decisionale IF 4](#_Toc45637508)

[REPEAT … UNTIL 4](#_Toc45637509)

[MESSAGE 5](#_Toc45637510)

[Sottoscrizione evento 5](#_Toc45637511)

[Come aggiungere una funzione a una Codeunit in Dynamics NAV 6](#_Toc45637512)

[Per aggiungere una funzione 6](#_Toc45637513)

[Code Unit con funzione richiamabile dall’esterno 7](#_Toc45637514)

[Lettura del calendario NAV Colussi 8](#_Toc45637515)

[Chiamare una page da un’altra page passandogli dei parametri 8](#_Toc45637516)

[CurrentPage.UPDATE 9](#_Toc45637517)

[Rendere editable una subform quando è editable la form che la contiene 10](#_Toc45637518)

[Cose da verificare 10](#_Toc45637519)

## CALCFIELDS

### Es.: CompanyInfo.CALCFIELDS(Picture);

Consente di calcolare il valore di una colonna FlowFields.

Il FlowFields è una colonna virtuale, il suo valore non è salvato in tabella.

Ciò significa che è necessario utilizzare la funzione CALCFIELDS o la funzione SETAUTOCALCFIELDS(Record) per aggiornarli.

Ad esempio, se si recupera un record utilizzando la funzione FIND (Record) e la funzione NEXT (Record), i FlowField in tali record vengono impostati su zero (0). Quindi, quando si chiama CALCFIELDS, i loro valori vengono aggiornati.

Quando FlowField è un'espressione di origine diretta su una pagina o un report, il calcolo viene eseguito automaticamente.

È inoltre possibile utilizzare la funzione CALCFIELDS per recuperare oggetti binari di grandi dimensioni (BLOB). Per ulteriori informazioni, consultare Tipo di dati BLOB.

Se possibile, la funzione CALCFIELDS utilizza la tecnologia SumIndexField (SIFT). SIFT viene utilizzato solo se sono vere le seguenti condizioni:

La chiave Dynamics NAV contiene i campi utilizzati nei filtri definiti per FlowField.

I SumIndexFields sulla chiave Dynamics NAV contengono i campi forniti come parametri per il calcolo.

La proprietà MaintainSIFTIndex è impostata su Sì.

## SETRANGE Function Record

ReminderHeader.SETRANGE("Archiving Sending Date",0DT);

Imposta un filtro semplice, ad esempio un singolo intervallo o un singolo valore, su un campo.

Per togliere il filtro :

ReminderHeader.SETRANGE("Archiving Sending Date");

## SETRANGE Function FieldRef

CustomerRecref.OPEN(DATABASE::Customer);

MyFieldRef := CustomerRecref.FIELD(1);

MyFieldRef.SETRANGE('10000' , '40000');

Imposta un filtro semplice, ad esempio un singolo intervallo o un singolo valore, su un campo.

## Test data blank

IF (DateDate <> 0D) THEN

…

## 0DT

Corrisponde alla data 1753-01-01 00:00:00.000 che è la data minima rappresentabile in SQL Server.

## Data e Ora corrente

DataRange := CREATEDATETIME(0D,0T);

## Test sulla Data indefinita (01-01-1753 00:00:00.000)

IF DateDate = 01011753D THEN

DateDate := 0D; (0D è data indefinita o blank)

## Data ultimo giorno dell’anno della data indicata

CALCDATE(‘<CY>’, Date)

## Data primo giorno dell’anno della data indicata

CALCDATE(‘<-CY>’, Date)

## FIND

Trova un record in una tabella basato sui valori memorizzati nelle chiavi.

È possibile combinare i caratteri '=', '<' e '>'. Puoi usare i seguenti caratteri:

* = per cercare un record uguale ai valori chiave (impostazione predefinita)
* > per cercare un record più grande dei valori chiave
* <per cercare un record inferiore ai valori chiave
* + per cercare l'ultimo record nella tabella (+ può essere utilizzato solo da solo)
* - per cercare il primo record nella tabella (- può essere utilizzato solo da solo)

Se questo parametro contiene '=', '>' o '<', è necessario assegnare un valore a tutti i campi della chiave corrente e primaria prima di chiamare FIND.

ItemRec."No." := '1100';

IF ItemRec.FIND('**=**') THEN

MESSAGE(TEXT000, ItemRec."No.", ItemRec.Description, ItemRec."Unit Price")

ELSE

MESSAGE(TEXT001);

## Calcolo giorni tra due date

Definisco una variabile (days) di tipo integer che conterrà il risultato del calcolo :

days := "Sell-In Ending Date" - "Sell-In Starting Date";

## FINDSET

Trova un set di record in una tabella in base alla chiave e al filtro correnti. FINDSET può recuperare i record solo in ordine crescente.

## FINDFIRST

Recupera solo un record, il primo all'interno del filtro, ordinato in modo crescente.

## FINDLAST

Recupera solo un record, l'ultimo all'interno del filtro, ordinato in modo crescente.

## ISEMPTY

Se vuoi sapere se esistono record all'interno di un determinato filtro, usa ISEMPTY.

## Struttura decisionale IF



Buffer.CALCFIELDS("Amount Including VAT");

value\_TOT\_FATTURA := CONVERTSTR(FORMAT(Buffer."Amount Including VAT", 0, '<Integer><Decimal,3>'), ',', '.');

IF Buffer."Currency Code" = '' THEN

value\_COD\_VALUTA := 'EUR'

ELSE

value\_COD\_VALUTA := Buffer."Currency Code";

## REPEAT … UNTIL

REPEAT

…

UNTIL GLAcc.NEXT = 0;

IF MDIIOU\_ORDNOTESalesNote.FINDSET THEN REPEAT

MDIIOU\_ORDNOTESalesNote.VALIDATE(SYNC\_DT\_INSERT, DateTimeInsert);

MDIIOU\_ORDNOTESalesNote.VALIDATE(CODPROCESS, CODPROCESSPar);

MDIIOU\_ORDNOTESalesNote.VALIDATE(DTEPROCESS, DTEPROCESSPar);

MDIIOU\_ORDNOTESalesNote.VALIDATE(IDSM1, IDSM1Par);

MDIIOU\_ORDNOTESalesNote.MODIFY(TRUE);

UNTIL MDIIOU\_ORDNOTESalesNote.NEXT = 0;

## MESSAGE

MESSAGE('Tesidet - Carico : ' + '%1' + ' Riga: ' + '%2' + ' Importo : ' + '%3', Tesidet.DENCAR, Tesidet.DENRIG, Tesidet.DENETD);

Visualizza un alert con messaggio.

## Sottoscrizione evento

|  |  |
| --- | --- |
| **Campo** | **Descrizione** |
| ID codeunit sottoscrittore | Specifica l'ID della codeunit contenente la funzione di sottoscrittore di eventi. |
| Funzione di sottoscrittore | Specifica la funzione di sottoscrittore di eventi nella codeunit del sottoscrittore che sottoscrive l'evento. |
| Tipo di evento | Specifica il tipo di evento, che può essere **Business**, **Integrazione** o **Trigger**. Per ulteriori informazioni sui tipi, vedere [Event Types](http://help2016.eos-solutions.it/help/it-it/conEventTypes.htm). |
| Tipo di oggetto autore | Tipo di oggetto che contiene la funzione di autore di eventi che pubblica l'evento. |
| ID oggetto autore | Specifica l'ID dell'oggetto che contiene la funzione di autore di eventi che pubblica l'evento. |
| Funzione pubblicata | Specifica il nome della funzione di autore di eventi nell'oggetto autore a cui si sottoscrive la funzione di sottoscrittore di eventi. |
| Attivo | Specifica se la sottoscrizione dell'evento è attiva o inattiva. Quando la casella di controllo è deselezionata, la sottoscrizione dell'evento è inattiva. Ciò si verifica se l'oggetto o la funzione di autore di eventi non viene trovata. |
| Numero di chiamate | Specifica quante volte la funzione di sottoscrittore di eventi è stata chiamata. La funzione di sottoscrittore di eventi viene chiamata quando l'evento pubblicato viene generato nell'applicazione.  Questo campo specifica il totale delle chiamate dall'ultimo avvio dell'istanza Microsoft Dynamics NAV Server o dopo la correzione dell'ultimo errore di sottoscrizione dell'evento. Il campo viene reimpostato su zero quando l'istanza di Microsoft Dynamics NAV Server è riavviata o quando si verifica un errore di sottoscrizione dell'evento. |
| Informazioni sull'errore | Specifica il messaggio di errore quando si verifica un errore e la sottoscrizione dell'evento non riesce.  Per ulteriori informazioni sui messaggi di errore, vedere [Event Errors](http://help2016.eos-solutions.it/help/it-it/conEventErrors.htm). |

## Come aggiungere una funzione a una Codeunit in Dynamics NAV

Per aggiungere una funzione a una codeunit, utilizzare la seguente procedura.

### Per aggiungere una funzione

1. Seleziona l'editor C / AL, scegli **Visualizza** , quindi scegli **Globali C / AL** .

**Nota**

Per ulteriori informazioni su come aprire l'editor C / AL, vedere [**Procedura: modificare un'unità di codice esistente**](https://docs.microsoft.com/en-us/dynamics-nav/how-to--modify-an-existing-codeunit) .

1. Nella finestra **Globali C / AL** , selezionare la scheda **Funzioni** .
2. Immettere un nome per ciascuna funzione che si desidera aggiungere.
3. Scegliere **Locali** per definire i parametri, il valore restituito, le variabili locali e le costanti di testo nella finestra **Locali C / AL** .
4. Nella finestra **Locals C / AL** , selezionare la scheda **Parametri** e specificare il metodo di chiamata, il nome e il tipo di dati di ciascun parametro. Puoi anche specificare un sottotipo e una lunghezza, ma questo è facoltativo.

Per specificare il metodo di chiamata di un parametro, selezionare o deselezionare la casella di controllo **Var** . Se si seleziona la casella di controllo **Var** , il parametro viene passato per riferimento anziché per valore. Il valore di una variabile può essere modificato da una funzione solo quando viene passato alla funzione per riferimento. Quando il parametro non è specificato come Var, alla funzione viene passata solo una copia della variabile. Se la funzione modifica quel valore, la modifica influisce solo sulla copia e non sulla variabile stessa.

Se il tipo selezionato corrisponde a un oggetto applicazione, è necessario aggiungere anche un sottotipo, ovvero il nome di un oggetto specifico nel database. Se si selezionano i tipi di dati Testo o Codice, è necessario definire una lunghezza. La lunghezza predefinita è 10 caratteri per il codice e 30 caratteri per il testo.

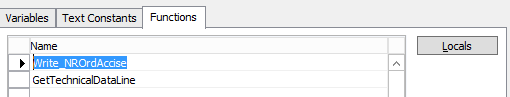
1. Nella finestra **Locals C / AL** , seleziona la scheda **Valore di ritorno** e specifica il valore di ritorno per la tua nuova funzione. Immettere un nome per il valore restituito e selezionare un tipo di dati dall'elenco a discesa. Se il tipo di dati è Testo o Codice, puoi anche selezionare una lunghezza.
2. Nella finestra **Locals C / AL** , selezionare la scheda **Variabili** e definire le variabili locali.

Per ogni variabile locale, è necessario inserire un nome e selezionare un tipo di dati. Se il tipo di dati selezionato corrisponde a un oggetto applicazione, è necessario aggiungere anche un sottotipo, ovvero il nome di un oggetto specifico nel database. Se si selezionano i tipi di dati Testo o Codice, è necessario definire una lunghezza per la variabile. La lunghezza predefinita è 10 caratteri per il codice e 30 caratteri per il testo.

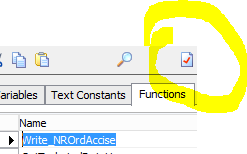
1. Nella finestra **Locals C / AL** , selezionare la scheda **Costanti di testo** e definire le costanti di testo per la funzione.

## Code Unit con funzione richiamabile dall’esterno

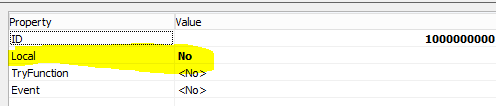
Vedi “Come aggiungere una funzione a una Codeunit in Dynamics NAV“, aggiungendo l’impostazione che la rende non locale :



Andare nelle proprietà :



e impostando :



Per richiamarla, ad esempio da una Page, definire una variabile :

Name DataType Subtype Length

ColussiRicevimenti Codeunit Colussi Ricevimenti NEXT

E richiamarla, passandogli tutti i parametri previsti :

**ColussiRicevimenti.Write\_NROrdAccise(WhseReceiptHeadL, WhseReceiptLine, DocShippingInfo);**

## Lettura del calendario NAV Colussi

Definire le seguenti variabili :

Name DataType Subtype Length

BaseCalChange Record Base Calendar Change

ColussiSetup Record Colussi Setup

CalendarCode Code 20

TargetDate Date

Nell’esempio seguente "Requested Delivery Date" contiene la data da verificare.

ColussiSetup.GET();

CalendarCode := ColussiSetup."Expected Deliv. Date Calendar";

TargetDate := "Requested Delivery Date";

BaseCalChange.RESET;

BaseCalChange.SETRANGE("Base Calendar Code",CalendarCode);

IF BaseCalChange.FINDSET THEN

REPEAT

CASE BaseCalChange."Recurring System" OF

BaseCalChange."Recurring System"::" ":

IF TargetDate = BaseCalChange.Date THEN BEGIN

ERROR(Text0001);

END;

BaseCalChange."Recurring System"::"Weekly Recurring":

IF DATE2DWY(TargetDate,1) = BaseCalChange.Day THEN BEGIN

ERROR(Text0001);

END;

BaseCalChange."Recurring System"::"Annual Recurring":

IF (DATE2DMY(TargetDate,2) = DATE2DMY(BaseCalChange.Date,2)) AND

(DATE2DMY(TargetDate,1) = DATE2DMY(BaseCalChange.Date,1))

THEN BEGIN

ERROR(Text0001);

END;

END;

UNTIL BaseCalChange.NEXT = 0;

## Chiamare una page da un’altra page passandogli dei parametri

L’esempio in questione coinvolge la page 46 che può chiamare la 80101.

Nella page 46 c’è un bottone che :

SalesOrdLineExcerptWizard.LOOKUPMODE(TRUE);

SalesOrdLineExcerptWizard.SetValue(Rec.Quantity, Rec."Unit of Measure Code", Rec."Excerpt Reason Code",SalesHeader."Sent to MDII", Rec."Sub Excerpt Reason Code");

IF SalesOrdLineExcerptWizard.RUNMODAL = ACTION::LookupOK THEN BEGIN

SalesOrdLineExcerptWizard.GetValue(RemainingQuantity, ExcerptReasonCode, SubExcerptReasonCode);

La SalesOrdLineExcerptWizard è la page 80101 nella quale sono state definite delle variabili globali che rappresentano i nostri parametri.

Name DataType Subtype Length

ExcerptQuantity Decimal

ExcerptReasonCode Code 3

UnitofMeasureCode Code 10

SendToMDII Boolean

SubExcerptReasonCode Option

Per scrivere i parametri si deve utilizzare la SetValue mentre per leggere i valori di ritorno si utilizza la GetValue:

**SetValue(QuantityPar : Decimal;UnitofMeasureCodePar : Code[10];CurrExcerptReasonCodePar : Code[3];SendToMDIIPar : Boolean;CurrSubExcerptReasonCodePar : Option)**

**UnitofMeasureCode := UnitofMeasureCodePar;**

**OriginalQuantity := QuantityPar;**

**ExcerptReasonCode := CurrExcerptReasonCodePar;**

**SendToMDII := SendToMDIIPar;**

**SubExcerptReasonCode := CurrSubExcerptReasonCodePar;**

**GetValue**(VAR RemainingQuantityPar : Decimal;VAR ExcerptReasonCodePar : Code[3];VAR SubExcerptReasonCodePar : 'Sostituito con altro prodotto,Riordino totale,Riordino parziale,Non riordinato,In fase di definizione')

RemainingQuantityPar := RemainingQuantity;

ExcerptReasonCodePar := ExcerptReasonCode;

SubExcerptReasonCodePar := SubExcerptReasonCode;

## CurrentPage.UPDATE

Salva il record corrente e quindi aggiorna i controlli nella pagina. Se si imposta il parametro SaveRecord su false, questa funzione non salverà il record prima dell'aggiornamento della pagina.

CurrPage.UPDATE[(SaveRecord)]

parametri

SaveRecord

Tipo: booleano

Impostare questo parametro su true se si desidera salvare il record corrente. Impostare questo parametro su false se si desidera aggiornare senza salvare il record corrente. Il valore predefinito è true.

## Rendere editable una subform quando è editable la form che la contiene

**OnAfterGetCurrRecord**()

DynamicEditable := CurrPage.EDITABLE;

dove DynamicEditable è una variabile globale booleana

## Formattazione data

FORMAT (DateP, 8 , '<year4><month,2><day,2>')

## GET

Ottiene un record in base ai valori momorizzati nei campi chiave primaria.

CustomerRec.GET('1120');

// This statement causes a run-time error if customer 1120 cannot be

// found. To avoid this, use the following construct:

IF CustomerRec.GET('1120') THEN

MESSAGE(Text000)

ELSE

MESSAGE(Text001);

## SETASCENDING (Record)

Imposta il criterio di ordinamento per i record restituiti. Utilizzare questa funzione dopo aver impostato i tasti per ordinare dopo, utilizzando SETCURRENTKEY. L'ordinamento predefinito è crescente. È possibile utilizzare SETASCENDING per modificare l'ordinamento in decrescente per un campo specifico, mentre gli altri campi nella chiave specificata sono ordinati in ordine crescente.

Ad esempio, è possibile leggere i dati da un servizio Web OData in cui i dati sono ordinati in ordine crescente nel campo Nome ma in ordine decrescente nel campo Città.

[Ascending :=] Record.SETASCENDING(Field,[ SetAscending])

#### parametri

Record  
Type: Record

Il record nella tabella per cui si desidera impostare l'ordinamento.

Campo  
Tipo: Campo

Il campo per cui si desidera impostare l'ordinamento.

SetAscending  
Type: Boolean

**L'ordinamento. Specificare FALSE se i dati devono essere ordinati in ordine decrescente; altrimenti TRUE.**

## Esempio

L'esempio di codice seguente mostra come utilizzare SETCURRENTKEY e SETASCENDING per ordinare i dati in due direzioni diverse in base a due campi.

copia

Customer.SETCURRENTKEY(City,"Name");

Customer.SETASCENDING(City,FALSE);

## Testo del messaggio in lingua

Consideriamo lo scenario in cui in una page vogliamo generare un messaggio di errore in lingua.

Per farlo dobbiamo creare, tra le variabili globali, una “Text Constants”, ad esempio:

Errore001

Nelle sue proprietà indicare :

ConstValueML : ENU=<testo in inglese>;ITA=<testo in italiano>

L’utilizzo invece è lo stesso ossia :

ERROR(Errore001);

## SETFILTER (combinare condizioni AND e OR)

SalesLine.RESET;

SalesLine.SETRANGE("Document Type", SalesLine."Document Type"::Order);

// CLS 30.09.2020 START

SalesLine.FILTERGROUP(-1); // CONSENTE LA DEFINIZIONE DI UNA CONDIZIONE OR

SalesLine.SETFILTER("Excerpt Reason Code", SetupUser."Exc. Reas. Log. Aut.");

SalesLine.SETFILTER("Reason Code Moved", SetupUser."Reason Code Moved");

Questo esempio è visibile nella page 80078

## Filtro per data odierna

REC.FILTERGROUP(2);

Rec.SETFILTER(Nome Campo, '>=%1', TODAY);

## Richiamare un scheda di sincronizzazione in modo programmatico

ExtSyncIntegrTableMap.GET('NAV', 'IC\_ITEM');

ExtSyncIntegrTableMap.SynchronizeNow(FALSE);