



Generatore Modelli

Funzioni avanzate

Gennaio 2020

Sommario:

Informazioni sul documento	2
Versione	2
Scopo	2
Riferimenti ad altri documenti	2
Definizioni – acronimi – glossario	2
1 Generatore modelli: istruzioni avanzate	3
2 Gestione delle variabili numeriche	3
2.1 Variabili numeriche o totalizzatori	3
2.2 Cambiare formato alle variabili numeriche	4
2.3 Le righe "trasparenti"	4
2.4 Riassegnare o azzerare una variabile numerica	5
2.5 Particolarità nella gestione delle variabili con campi di tipo data	6
3 Gestione delle variabili di tipo stringa	7
3.1 Variabili di tipo stringa o alfanumeriche	7
3.2 Cambiare formato alle variabili alfanumeriche	7
3.3 Riassegnare o azzerare una variabile alfanumerica	8
3.4 Concatenare più stringhe	9
4 Funzioni di conversione delle variabili	10
4.1 Conversione di stringhe in numeri	10
4.2 Conversione di numeri in stringhe	11
5 L'istruzione condizionale	12
5.1 Operatori logici di confronto	15
5.2 Relazioni tra condizioni	15
5.3 Particolarità dell'istruzione di condizione alternativa	16
5.4 Particolarità di condizioni su mnemonici tabellati	17
5.5 Particolarità di condizioni su mnemonici di tipo Si/No	17
5.6 Particolarità di condizioni su mnemonici di tipo data	18
6 Troncare e arrotondare le cifre con l'uso di formati e condizioni	19
7 Istruzione di ripetizione (ciclo, loop)	20
7.1 Particolarità dei cicli senza entità	21
8 Parametri di input	22
8.1 Inserimento e configurazione dei parametri	22
8.2 Utilizzo dei parametri all'interno del modello	23

Informazioni sul documento

Versione

Versione	Data	Modifiche apportate
1.0	01/10/2008	Non applicabile in quanto questa è la prima versione del documento
2.0	16/05/2019	Aggiornamento complessivo del documento per restyling Aggiornamento comandi con nuova sintassi per [IF] e [FOR]
3.0	07/01/2020	Aggiunto caso particolare al cap. "3.4 Concatenare più stringhe"

Scopo

Il presente documento descrive la funzionalità "Generatore modelli" che permette di comporre un documento a partire da un modello predefinito nel quale venga indicata un'apposita sintassi di programmazione.

Riferimenti ad altri documenti

Riferimenti
1. Generatore modelli: Funzioni base

Definizioni – acronimi – glossario

Termine – acronimo	Significato

1 Generatore modelli: istruzioni avanzate

Questo capitolo descrive i concetti e le istruzioni avanzate che consentono di sfruttare maggiormente il Generatore modelli.

2 Gestione delle variabili numeriche

2.1 Variabili numeriche o totalizzatori

Può essere necessario talvolta, soprattutto per operazioni più complesse, "registrare" il risultato parziale in una variabile, in modo da poterlo anche utilizzare in altre parti del testo. Per questo è possibile utilizzare i totalizzatori che sono appunto variabili numeriche alle quali posso assegnare un valore, il risultato di un'espressione o utilizzarle in un'espressione stessa.

TOTALIZZATORI

Sintassi: #TOTnn#

Esempi: #TOT1#

Note: I totalizzatori sono individuati con il mnemonico TOT al quale è seguito un numero da 1 a 997.

Esistono delle particolari istruzioni per la gestione dei totalizzatori qui sotto elencate:

<code>\++nn\</code>	Inizializza il totalizzatore #TOTnn#
<code>\+nn\</code>	Somma un valore al totalizzatore #TOTnn#
<code>\-nn\</code>	Sottrae un valore al totalizzatore #TOTnn#

Esempio:

```
@#CODIMP#  
  
@Assegno il valore 5 al totalizzatore TOT1  
\5\|++1\  
Il valore di TOT1 è #TOT1#  
  
@Sommo 10 al totalizzatore TOT1  
\10\|+1\  
Il nuovo valore di TOT1 sommando 10 è #TOT1#
```

Il risultato del file composto sarà:

```
5.00  
Il valore di TOT1 è 5.00000  
  
10.00  
Il nuovo valore di TOT1 sommando 10 è 15.00000
```

2.2 Cambiare formato alle variabili numeriche

Le variabili numeriche (o totalizzatori) vengono riportate con un formato numerico senza separatori di migliaia e con 5 decimali. Per cambiarne il formato è possibile utilizzare le Funzioni di cambio formato già viste per i mnemonici.

Esempio:

```
@#CODIMP#  
  
@Assegno il valore 5 al totalizzatore TOT1  
\5\\++1\  
Il valore di TOT1 è #TOT1[N5]#  
  
@Sommo 10 al totalizzatore TOT1  
\10\\+1\  
Il nuovo valore di TOT1 sommando 10 è #TOT1{MONEY}#
```

Il risultato del file composto sarà:

```
5.00  
Il valore di TOT1 è 5  
  
10.00  
Il nuovo valore di TOT1 sommando 10 è 15.00
```

2.3 Le righe "trasparenti"

Quando si esegue un'operazione o si assegna un valore ad una variabile numerica può essere necessario evitare di riportare il risultato nel documento composto.

Il comando di commento @ a inizio riga permette anche di eseguire istruzioni che però non si vogliono vengano riportate nel documento composto.

Esempio:

```
@#CODIMP#  
  
@Assegno il valore 5 al totalizzatore TOT1  
@\5\\++1\  
Il valore di TOT1 è #TOT1[N5]#  
  
@Sommo 10 al totalizzatore TOT1  
@\10\\+1\  
Il nuovo valore di TOT1 sommando 10 è #TOT1{MONEY}#
```

Il risultato del file composto sarà:

```
Il valore di TOT1 è 5  
  
Il nuovo valore di TOT1 sommando 10 è 15.00
```

2.4 Riassegnare o azzerare una variabile numerica

L'esempio seguente mostra come riassegnare (ad esempio per riutilizzare per altri scopi) un valore ad una variabile numerica.

Esempio:

```
@#CODIMP#  
  
Il valore predefinito di TOT1 è #TOT1#  
  
@Assegno il valore 5 al totalizzatore TOT1 \5\\++1\  
Il valore di TOT1 ora è #TOT1#  
  
@Sommo 10 al totalizzatore TOT1 \10\\+1\  
Il nuovo valore di TOT1 sommando 10 è #TOT1#  
  
@Tolgo 3 al totalizzatore TOT1 \3\\-1\  
Il nuovo valore di TOT1 togliendo 3 è #TOT1#  
  
@Salvo in TOT2 il valore attuale di TOT1 \#TOT1#\\++2\  
@Assegno al totalizzatore TOT1 un nuovo valore \50\\++1\  
Il nuovo valore riassegnato a TOT1 è #TOT1#  
Il precedente valore era #TOT2#  
  
@Azzerò TOT1 \0\\++1\  
Ora TOT1 vale #TOT1#
```

Il risultato del file composto sarà:

```
Il valore predefinito di TOT1 è 0.00000  
  
Il valore di TOT1 ora è 5.00000  
  
Il nuovo valore di TOT1 sommando 10 è 15.00000  
  
Il nuovo valore di TOT1 togliendo 3 è 12.00000  
  
Il nuovo valore riassegnato a TOT1 è 50.00000  
Il precedente valore era 12.00000  
  
Ora TOT1 vale 0.00000
```

2.5 Particolarità nella gestione delle variabili con campi di tipo data

Se si eseguono operazioni o si inizializzano variabili numeriche con dei campi di tipo data, è necessario considerare che queste vengono trattate come un numero progressivo che identifica il numero di giorni a partire dall'anno 0.

Per ottenere il risultato dell'operazione in formato data utilizzare il formato [D].

Esempio:

@#CODIMP#

La data di iscrizione della ditta è #DCCIAA#

@Aggiungo 30 giorni alla data di iscrizione e la salvo nella variabile TOT1

@\#DCCIAA#+7\|++1\

@Stampo il risultato in formato data

La ditta dovrà presentare documentazione integrativa entro il giorno #TOT1[D]#

Se non uso il formato D viene stampato #TOT1#

Il risultato del file composto sarà:

La data di iscrizione della ditta è 01.07.2004

La ditta dovrà presentare documentazione integrativa entro il giorno 11.07.2004

Se non uso il formato D viene stampato 366895.00000

3 Gestione delle variabili di tipo stringa

3.1 Variabili di tipo stringa o alfanumeriche

Oltre alle variabili di tipo numerico (TOTnn) sono disponibili anche variabili di tipo alfanumerico o variabili stringa (STRnn) e possono venire utilizzate per la registrazione temporanea di stringhe di caratteri o del contenuto dei mnemonici. L'assegnazione di un valore a queste variabili si ottiene mediante la funzione [TOSTR].

FUNZIONE [TOSTR]

Sintassi: [TOSTR]#STRnn#="....."

Esempi: [TOSTR]#STR1#="stringa"

Note: La stringa deve essere indicata tra doppio apice.
Il comando deve essere scritto ad inizio riga.

Esempio:

```
@#CODIMP#  
  
[TOSTR]#STR1#=#NOMIMP#  
@Assegna alla variabile STR1 il nome dell'impresa  
  
[TOSTR]#STR2#="Il nome dell'impresa corrente è "  
@Assegna il testo alla variabile STR2  
  
#STR2##STR1#
```

Il risultato del file composto sarà:

```
Il nome dell'impresa corrente è Fedele Costruzioni snc
```

3.2 Cambiare formato alle variabili alfanumeriche

È possibile applicare le funzioni di cambio formato dei mnemonici anche alle stringhe, sia in fase di assegnazione che di stampa.

Si noti che è possibile assegnare ad una stringa il codice/numero di un campo tabellato mediante la funzione .N, mentre non è possibile farlo applicando il formato direttamente alla stringa (l'istruzione #STR3.N# non porterebbe ad alcuna conversione di valore).

Esempio:

```
@#CODIMP#  
  
[TOSTR]#STR1#=#NOMIMP#  
[TOSTR]#STR2#=#NOMIMP[A1]#  
[TOSTR]#STR3#=#PROIMP#  
[TOSTR]#STR4#=#PROIMP.N#  
  
STR1 contiene il nome dell'impresa #STR1#.  
L'iniziale del nome dell'impresa è #STR1[A1]#.  
L'iniziale del nome dell'impresa è registrata in STR2 come #STR2#.  
  
STR3 contiene la descrizione della provincia #STR3#  
STR3.N non funziona #STR3.N#  
STR4 contiene la sigla della provincia #STR4#
```

Il risultato del file composto sarà:

*STR1 contiene il nome dell'impresa Fedele Costruzioni snc
L'iniziale del nome dell'impresa è F.
L'iniziale del nome dell'impresa è registrata in STR2 come F.*

*STR3 contiene la descrizione della provincia Treviso
STR3.N non funziona Treviso
STR4 contiene la sigla della provincia TV*

3.3 Riassegnare o azzerare una variabile alfanumerica

L'esempio seguente mostra come riassegnare (ad esempio per riutilizzare per altri scopi) un valore ad una variabile stringa.

Esempio:

*@#CODIMP#
[TOSTR]#STR1#=#NOMIMP#
STR1 contiene il nome dell'impresa #STR1#
[TOSTR]#STR1#=#PROIMP#
STR1 ora contiene la provincia #STR1#
[TOSTR]#STR1#=""
STR1 ora è vuoto #STR1#*

Il risultato del file composto sarà:

*STR1 contiene il nome dell'impresa Fedele Costruzioni snc
STR1 ora contiene la provincia Treviso
STR1 ora è vuoto*

3.4 Concatenare più stringhe

Una funzione che può risultare utile in diverse occasioni è la funzione di concatenazione (unione) di stringhe, mnemonici, costanti o totalizzatori.

La funzione può essere utile in particolare combinata all'istruzione di condizione (descritta in seguito) per cambiare il contenuto di una stringa in funzione della condizione stessa.

Questa funzione è definita [CATSTR].

FUNZIONE [CATSTR]

Sintassi: [CATSTR]#STRnn#=#MNEMONICO1#,#MNEMONICO2#

Esempi: [CATSTR]#STR1#="stringa1","MNEMONICO1"
[CATSTR]#STR2#="Il codice impresa è ",#CODIMP#

Esempio:

```
@#CODIMP#  
  
[CATSTR]#STR1#=#CODIMP#," - - - "  
@Assegna alla variabile STR1 il codice dell'impresa e aggiunge tre trattini  
  
[CATSTR]#STR1#=#STR1#,#NOMIMP#  
@Aggiungo a STR1 il nome dell'impresa  
  
#STR1#  
@Stampo STR1
```

Il risultato del file composto sarà:

```
00179 - - - Fedele Costruzioni snc
```

ATTENZIONE:

Non è possibile concatenare due stringhe in questo modo:

```
[CATSTR]#STR1#="stringa1","stringa2"
```

Se fosse necessario, è possibile procedere così:

```
[TOSTR]#STR1#="stringa1"  
[CATSTR]#STR1#=#STR1#,"stringa2"
```

Non è possibile riportare nel modello due "cancelletti" consecutivi, ad es.: #STR1##STR2#
Risolvere con:

```
[CATSTR]#STR3#=#STR1#,#STR2#
```

E riportare nel modello #STR3#



4 Funzioni di conversione delle variabili

4.1 Conversione di stringhe in numeri

È possibile convertire una variabile stringa in una variabile numerica utilizzando la funzione di inizializzazione di un totalizzatore.

Se la stringa non contiene solo numeri, ma anche testo il risultato sarà 0.

Si presti attenzione anche all'uso del separatore delle migliaia e della virgola. La conversione avviene solo se la stringa contiene numeri interi senza separatori delle migliaia per la parte intera e se viene usato il punto come separatore dei decimali.

Esempio:

```
@#CODIMP#

[TOSTR]#STR1#="1297.89"
STR1 contiene #STR1#
@converto la stringa STR1 in TOT1 \#STR1#\|++1\
TOT1 contiene #TOT1#

[TOSTR]#STR1#="1297,89"
STR1 contiene #STR1#
@converto la stringa STR1 in TOT1 \#STR1#\|++1\
TOT1 contiene #TOT1#

[TOSTR]#STR1#=" 1297.89  "
STR1 contiene #STR1#
@converto la stringa STR1 in TOT1 \#STR1#\|++1\
TOT1 contiene #TOT1#

[TOSTR]#STR1#="1297. 89"
STR1 contiene #STR1#
@converto la stringa STR1 in TOT1 \#STR1#\|++1\
TOT1 contiene #TOT1#

[TOSTR]#STR1#="abc1297.89"
STR1 contiene #STR1#
@converto la stringa STR1 in TOT1 \#STR1#\|++1\
TOT1 contiene #TOT1#
```

Il risultato del file composto sarà:

```
STR1 contiene 1297.89
TOT1 contiene 1297.89000

STR1 contiene 1297,89
TOT1 contiene 0.00000

STR1 contiene 1297.89
TOT1 contiene 1297.89000

STR1 contiene 1297. 89
TOT1 contiene 1297.89000

STR1 contiene abc1297.89
TOT1 contiene 0.00000
```

4.2 Conversione di numeri in stringhe

È possibile convertire una variabile numerica in una variabile stringa assegnando alla stringa il totalizzatore.

È possibile anche assegnare alla stringa il valore numerico sfruttando le funzioni di formattazione e conversione dei mnemonici.

Esempio:

```
@#CODIMP#  
  
@\9256.589\\++1\  
La variabile numerica TOT1 vale #TOT1#  
  
[TOSTR]#STR1#=#TOT1#  
Conversione semplice #STR1#  
  
[TOSTR]#STR1#=#TOT1[F10.2]#  
Conversione con formato F10.2 #STR1#  
  
[TOSTR]#STR1#=#TOT1[Z15]#  
Conversione con formato Z15 #STR1#  
  
[TOSTR]#STR1#=#TOT1{MONEY}#  
Conversione con dominio MONEY #STR1#  
  
[TOSTR]#STR1#=#TOT1[D]#  
Conversione con formato D #STR1#
```

Il risultato del file composto sarà:

```
La variabile numerica TOT1 vale 9256.58900  
  
Conversione semplice 9256.58900  
  
Conversione con formato F10.2 9256.59  
  
Conversione con formato Z15 000000000009256  
  
Conversione con dominio MONEY 9'256.59  
  
Conversione con formato D 06.05.1025
```

5 L'istruzione condizionale

Durante composizione di un modello può essere necessario dover riportare un dato piuttosto che un altro a seconda della veridicità o meno di una o più condizioni.

Questa esigenza viene soddisfatta da una serie di comandi che permettono il confronto logico tra mnemonici, stringhe o costanti, eseguendo in base al risultato o meno una o più operazioni.

È possibile anche inserire una condizione all'interno di un'altra (condizioni annidate); in questo caso la verifica della seconda condizione avviene solo nel caso in cui sia vera la prima.

La sintassi del comando prevede l'utilizzo del comando [IF] come descritto in seguito.

ISTRUZIONE CONDIZIONALE

Sintassi:

```
[IF] %condizione1%  
  
... testo ed istruzioni se verificata la condizione1  
  
[ELSEIF] %condizione2%  
  
... testo ed istruzioni se verificata la condizione2 in alternativa  
    alla condizione1  
  
[ELSEIF]  
  
... testo ed istruzioni se non sono verificate la condizioni  
    precedenti  
  
[ENDIF]
```

Nota 1: Ogni istruzione ha un determinato livello [IF]; per ogni livello possono essere impostate numerose condizioni alternative [ELSEIF], ma può esistere una sola istruzione alternativa [ELSE], e deve esistere l'istruzione di fine condizione [ENDIF].

Nota 2: All'interno di un tratto di testo compreso tra 2 righe di [IF], [ELSEIF], [ELSE] o [ENDIF] può essere inserita una ulteriore istruzione condizionale completa [IF], [ELSEIF], [ELSE] o [ENDIF]; Questo annidamento può essere ripetuto quante si voglia volte.

L'errore più comune che viene commesso quando si usa l'istruzione di condizione è dimenticare di chiudere la condizione con l'istruzione [ENDIF] , o di intrecciare livelli di [IF] annidate.

Si suggerisce pertanto di scrivere subito l'istruzione [ENDIF] per ciascun livello [IF] aperto, e completare successivamente il testo tra le due righe.

È anche possibile indentare le [IF] sfruttando il comando @ per rendere più leggibili i livelli, ad esempio:

```
@ [IF] %condizione1%  
  
@      [IF] %condizione2%  
  
@          [IF] %condizione3%  
  
... testo istruzioni se verificano tutte le condizioni  
  
@          [ENDIF]  
  
@      [ENDIF]  
  
@ [ENDIF]
```

La sintassi delle condizioni si basa sugli operatori logici di seguito descritti che permettono il confronto tra mnemonici, stringhe o costanti.

Esempio:

```
@#CODIMP#  
  
[IF] %#GFIIMP.N#_EQ_1%  
@Se il grado di fiducia dell'impresa è 1, stampa la seguente frase:  
  
L'impresa #NOMIMP# è di fiducia.  
  
[ELSEIF] %#GFIIMP.N#_EQ_2%  
@Altrimenti, se il grado di fiducia dell'impresa è 2, stampa quest'altra frase:  
  
L'impresa #NOMIMP# partecipa per la prima volta.  
  
[ELSE]  
@Altrimenti, se non si verificano le prime due condizioni, stampa la frase:  
  
L'impresa #NOMIMP# è sconosciuta.  
  
[ENDIF]  
@Fine della serie di condizioni di livello 1
```

Il risultato del file composto sarà, se si verifica la prima condizione:

L'impresa Fedele Costruzioni snc è di fiducia.

se si verifica la seconda condizione:

L'impresa Fedele Costruzioni snc partecipa per la prima volta.

se non si verifica nessuna delle due condizioni:

L'impresa Fedele Costruzioni snc è sconosciuta.

Come già detto, un'istruzione condizionale può contenere a sua volta un'altra istruzione condizionale. In questo caso deve essere specificato un livello diverso:

Esempio:

```
@#CODIMP#  
  
[IF] %#GFIIMP.N#_EQ_1%  
@    [IF] %#PROIMP.N#_EQ_TV%  
L'impresa #NOMIMP# è di fiducia e risiede in provincia .  
@    [ELSE]  
L'impresa #NOMIMP# è di fiducia e ma risiede fuori provincia  
@    [ENDIF]  
[ELSE]  
L'impresa #NOMIMP# partecipa per la prima volta.  
[ENDIF]
```

Il risultato del file composto sarà, se si verifica la prima condizione di livello 1 e la prima di livello 2:

```
L'impresa Fedele Costruzioni snc è di fiducia e risiede in provincia.
```

5.1 Operatori logici di confronto

Negli esempi precedenti, all'interno delle condizioni, è stato utilizzato l'operatore di confronto `_EQ_` che permette di verificare l'uguaglianza tra due dati (mnemonici, costanti, totalizzatori, stringhe, ecc...). Sono disponibili anche altri operatori di confronto, di seguito elencati, utili in altre situazioni:

<code>_EQ_</code>	Uguale (dall'inglese "equal")
<code>_NE_</code>	Diverso (dall'inglese "not equal")
<code>_GT_</code>	Maggiore (dall'inglese "greater than")
<code>_LT_</code>	Minore (dall'inglese "less than")
<code>_GE_</code>	Maggiore o uguale
<code>_LE_</code>	Minore o uguale

Per confrontare due dati è necessario che questi abbiano formato e tipo compatibili.

5.2 Relazioni tra condizioni

È possibile utilizzare all'interno della stessa riga più condizioni purché queste siano separate da un operatore di relazione. Questi operatori permettono di verificare la veridicità di tutte le condizioni o di almeno una di esse senza dover utilizzare livelli diversi di condizione. Sono a seguito riportati:

<code>_AND_</code>	E: la condizione è vera se tutte e due le condizioni tra le quali è posto l'operatore <code>_AND_</code> sono verificate contemporaneamente.
<code>_OR_</code>	O: la condizione è vera se almeno una delle due condizioni tra le quali è posto l'operatore <code>_OR_</code> è verificata.

Esempio:

```
@#CODIMP#  
  
[IF] %#GFIIMP.N#_EQ_1_AND_#PROIMP.N#_EQ_TV%  
L'impresa #NOMIMP# è di fiducia e risiede in provincia .  
[ELSEIF] %#GFIIMP.N#_EQ_1_AND_#PROIMP.N#_NE_TV%  
L'impresa #NOMIMP# è di fiducia e ma risiede fuori provincia  
[ELSE]  
L'impresa #NOMIMP# partecipa per la prima volta.  
[ENDIF]
```

Il risultato del file composto sarà, se si verifica la prima condizione:

```
L'impresa Fedele Costruzioni snc è di fiducia e risiede in provincia.
```

5.3 Particolarità dell'istruzione di condizione alternativa

Prestare attenzione nel caso di utilizzo della condizione alternativa [ELSEIF].
Considerare l'esempio seguente:

Esempio:

```
...  
Assegno a TOT1 il valore \25\\++1\  
[IF] %#TOT1#_GT_0%  
TOT1 è maggiore di 0  
[ELSEIF] %#TOT1#_GT_10%  
TOT1 è maggiore di 10  
[ELSEIF] %#TOT1#_LT_30%  
TOT1 è minore di 30  
[ELSEIF] %#TOT1#_GT_50%  
TOT1 è maggiore di 50  
[ELSE]  
TOT1 è uguale o miniore di 0  
[ENDIF]  
....
```

Il risultato della composizione sarà:

```
...  
Assegno a TOT1 il valore 25.00000  
TOT1 è maggiore di 0  
...
```

Pur essendo vere anche le condizioni TOT1 è maggiore di 0, TOT1 è maggiore di 10 e TOT1 è minore di 30, viene stampata solo la prima che soddisfa la condizione e trascurate le successive.

Se si desidera che tutte le condizioni vengano modificate, è necessario cambiare il testo come nell'esempio che segue.

Esempio:

```
@#CODIMP#  
  
Assegno a TOT1 il valore \25\\++1\  
[IF] %#TOT1#_GT_0%  
TOT1 è maggiore di 0  
[ENDIF]  
[IF] %#TOT1#_GT_10%  
TOT1 è maggiore di 10  
[ENDIF]  
[IF] %#TOT1#_LT_30%  
TOT1 è minore di 30  
[ENDIF]  
[IF] %#TOT1#_GT_50%  
TOT1 è maggiore di 50  
[ENDIF]  
[IF] %#TOT1#_LE_0%  
TOT1 è uguale o miniore di 0  
[ENDIF]
```

Il risultato del file composto questa volta sarà:

```
Assegno a TOT1 il valore 25.00000  
TOT1 è maggiore di 0  
TOT1 è maggiore di 10  
TOT1 è minore di 30
```

5.4 Particolarità di condizioni su mnemonici tabellati

Si consiglia vivamente di utilizzare il formato .N per fare il confronto con il codice/numero dei mnemonici tabellati anziché con la loro descrizione.

Questo perché le descrizioni possono cambiare nel tempo (anche per banali correzioni ortografiche), mentre rimangono normalmente immutati i rispettivi codici/numeri.

La scrittura della condizione è inoltre più semplice (si usano numeri o codici), diversamente è necessario scrivere il testo rispettando anche maiuscole e minuscole.

L'esempio seguente mostra la medesima condizione scritta usando il formato .N e senza.

Esempio:

```
@#CODIMP#  
[IF] %#PROIMP.N#_NE_TV%  
L'impresa #NOMIMP# risiede fuori dalla provincia di Treviso.  
[ENDIF]  
  
[IF] %#PROIMP.N#_NE_Treviso%  
L'impresa #NOMIMP# risiede fuori dalla provincia di Treviso.  
[ENDIF]
```

5.5 Particolarità di condizioni su mnemonici di tipo Si/No

Particolare attenzione va rivolta ai confronti all'interno dell'istruzione condizionale nel momento in cui uno dei termini di confronto è un campo di tipo Si/No in quanto il valore del campo varia in base all'utilizzo dello stesso.

Se si stampa semplicemente il mnemonico del campo, nel testo composto potrà essere mostrata la dicitura Si, No o nulla (caso particolare) a seconda del valore del campo.

Se lo si usa all'interno di una condizione questo viene convertito automaticamente in forma numerica e in tal caso vale 1 se è Si (quindi non va confrontato con "_EQ_Si", ma con "_EQ_1"), oppure 0 se è No

Per praticità e per evitare errori nei confronti, si consiglia di applicare sempre la funzione .X considerando che il valore del campo in questo caso sarà sempre 1 quanto vale Si e può essere considerato No in tutti gli altri casi.

Esempio:

```
@#CODIMP#  
  
Valore del menmonico #G_ISCNOS#  
Valore del menmonico con .X #G_ISCNOS.X#  
[IF] %#G_ISCNOS.X#_EQ_1%  
L'impresa è iscritta  
[ELSE]  
L'impresa non è iscritta  
[ENDIF]
```

Il risultato del file composto questa volta sarà:

```
Valore del menmonico Si  
Valore del menmonico con .X 1  
L'impresa è iscritta
```

5.6 Particolarità di condizioni su mnemonici di tipo data

Nella necessità in cui si debbano confrontare dei mnemonici con campi di tipo data, all'interno di un'istruzione condizionale IF, deve essere utilizzato il formato di stampa [D]. Il formato deve essere applicato ad almeno uno dei due termini di confronto.

Esempio:

```
@#CODIMP#  
@Tolgo 365 giorni dalla data odierna \#DATAOGGI#-365\|++1\  
Considero la data di riferimento #TOT1[D]#  
La data di iscrizione è #DCCIAA#  
  
[IF] %#DCCIAA[D]#_GT_#TOT1[D]#%  
La ditta #NOMIMP# è iscritta alla CCIAA da meno di un anno  
[ELSE]  
La ditta #NOMIMP# è iscritta alla CCIAA da oltre di un anno  
[ENDIF]
```

Il risultato compilato sarà, se le date sono uguali:

```
Considero la data di riferimento 23.08.2006  
La data di iscrizione è 01.07.2004  
  
La ditta Fedele Costruzioni snc è iscritta alla CCIAA da oltre di un anno
```

Nel confronto tra date, in quanto gestito all'interno dell'istruzione condizionale, possono essere utilizzati tutti gli operatori di confronto.

6 Troncare e arrotondare le cifre con l'uso di formati e condizioni

Può essere necessario disporre o sommare delle cifre con decimali senza dover considerare gli arrotondamenti, oppure può essere utile effettuare un particolare arrotondamento.

L'esempio seguente mostra come è possibile cambiare il valore di una variabile numerica applicando un arrotondamento sfruttando le funzioni di formattazione dei mnemonici sui totalizzatori e sulle operazioni.

Esempio:

```
@#CODIMP#  
La superficie occupata è di metri 1259.655  
  
@inializzo TOT1 con il valore della superficie \1259.655\|++1\  
Applicando il formato N5 viene riportata la superficie #TOT1[N5]# senza arrotondamento  
Applicando il formato N5 viene riportata la superficie #TOT1[N5]# senza arrotondamento  
Applicando il formato F10.2 viene riportata la superficie #TOT1[f10.2]# con arrotondamento  
Ma la superficie registrata in TOT1 è #TOT1#  
  
Applico una regola per arrotondare l'unità per difetto se il primo decimale è inferiore a 6  
@ Ricavo in TOT2 il primo decimale di TOT1  
@ \(#TOT1#-#TOT1[N10]#)*10\|++2\  
Il primo decimale di TOT1 è #TOT2[N1]#  
@ la cifra rimane con l'unità presente se se il primo decimane è miniore di 6  
[IF] %#TOT2#_LT_6%  
@ ricalcolo TOT1 togliendo solo i decimali \#TOT1[N10]#\|++1\  
La superficie arrotondata con questo criterio è #TOT1[N10]#  
TOT1 ora vale #TOT1#  
@ la cifra va sommata di 1, quindi all'unità superiore, in caso contrario  
[ELSE]  
@ ricalcolo TOT1 aggiungendo l'unità \#TOT1[N10]#+1\|++1\  
La superficie arrotondata con questo criterio è #TOT1[N10]#  
TOT1 ora vale #TOT1#  
[ENDIF]
```

Il risultato del file composto sarà:

```
La superficie occupata è di metri 1259.655  
  
Applicando il formato N5 viene riportata la superficie 1259 senza arrotondamento  
Applicando il formato F10.2 viene riportata la superficie 1259.66 con arrotondamento  
Ma la superficie registrata in TOT1 è 1259.65500  
  
Applico una regola per arrotondare l'unità per difetto se il primo decimale è inferiore a 6  
Il primo decimale di TOT1 è 6  
La superficie arrotondata con questo criterio è 1260  
TOT1 ora vale 1260.00000
```

7 Istruzione di ripetizione (ciclo, loop)

Questa istruzione ha il compito di ripetere nel testo un gruppo di righe o istruzioni per un numero qualsiasi di volte, espandendosi in senso verticale, al fine di ottenere ad esempio delle liste o delle tabelle.

Questa istruzione è indispensabile estrarre i dati da entità "figlie", ad esempio per riportare nel testo l'elenco di tutti i legali rappresentanti di un'impresa o l'elenco delle categorie a cui è iscritta.

ISTRUZIONE [FOR] [NEXT]

Sintassi: [FOR]Ind = LimInf, LimSup, Passo | Ordinamento
Istruzioni o righe da ripetere
[NEXT]Ind

dove **Ind** è una variabile numerica che rappresenta l'indice del ciclo, o contatore, e viene incrementata ogni volta che lo stesso viene ripetuto.

LimInf e **LimSup** fissano il valore di partenza e di fine del ciclo stesso. Il ciclo parte dal valore di LimInf e si interrompe nel momento in cui il contatore Ind raggiunge o supera il valore di LimSup. LimInf e LimSup possono essere numeri, mnemonici o totalizzatori.

Il **Passo** indica il valore del quale il contatore viene incrementato ad ogni ciclo. Se non specificato diversamente, di default, il contatore viene incrementato di 1. Non è un dato obbligatorio.

L'**Ordinamento**, è un mnemonico, dell'entità di riferimento, per il quale si vuole ordinare la stampa o l'esecuzione delle istruzioni contenute nel ciclo. Specificando dopo il mnemonico il simbolo "+" o "-" viene impostato, rispettivamente l'ordine crescente o decrescente. Nel caso non sia specificato l'ordinamento, vengono estratti i dati nell'ordine in cui sono immagazzinati. Possono essere inseriti anche più mnemonici separati dal carattere |.

Note: Ad ogni ciclo viene assegnata un'entità che sarà quella del primo mnemonico incontrato dopo la riga [FOR]. È quindi consigliabile specificare il mnemonico chiave dell'entità, preceduto dal carattere @ come prima riga del ciclo.

All'interno del ciclo, la variabile Ind può essere usata come un normale mnemonico che conterrà il valore del contatore, richiamata con la sintassi normale del mnemonico.

Sono consentiti cicli uno all'interno di un altro purché abbiano identificativi del contatore diversi.

Esempio:

```
@#CODIMP#  
Le categorie di iscrizione della ditta #NOMIMP# sono:  
@ciclo sulle categorie in ordine di importo di iscrizione inverso  
[FOR]AA=1,999|#IMPISC#-  
@#CODIMP1#  
#AA#) Categoria #CATISC# Importo #IMPISC#  
[NEXT]AA
```

Il risultato del file composto sarà:

```
Le categorie di iscrizione della ditta Fedele Costruzioni snc sono:  
1) Categoria OS24 Importo 100'000.00  
2) Categoria OS18 Importo 50'000.00
```

Non conoscendo a priori il numero delle categorie è stato impostato il limite superiore (LimSup) come un numero sufficientemente grande (999), il ciclo si ferma comunque nel momento in cui sono finite le categorie presenti per l'impresa (ovvero quando non sono più presenti record di database e il campo chiave #CODIMP1# è nullo).

All'interno dell'istruzione di loop possono essere eseguite anche operazioni, funzioni o condizioni oltre ai normali richiami ai mnemonici, oppure annidati altri cicli purchè si cambi il nome dell'indice, ad esempio in #BB#).

Nella scrittura del codice, si consiglia inoltre di chiudere subito l'istruzione di loop con \$\$Ind e poi proseguire con la scrittura dell'ulteriore codice all'interno dell'istruzione. Si eviterà così di incorrere nell'errore "sono rimasti dei cicli aperti".

7.1 Particolarità dei cicli senza entità

Nel caso si abbia l'esigenza di ripetere del testo a prescindere dalla logica di estrazione dei dati, è possibile utilizzare nel ciclo il parametro #_NOENT_#.

L'esempio seguente mostra come ripetere per 4 pagine lo stesso testo. Senza l'opzione #_NOENT_# il ciclo si fermerebbe alla prima pagina in quanto opererebbe sempre sull'unica impresa su cui si stanno attingendo i dati.

Esempio:

```
@ #CODIMP#  
Si prega il legale rappresentante della ditta #NOMIMP# di firmare in tutte le pagine che seguono.  
  
[FOR]AA=1,4|#_NOENT_#  
@ #CODIMP#  
#NOMIMP# #NOMLEG#  
  
----- salto pagina  
[NEXT]AA
```

8 Parametri di input

8.1 Inserimento e configurazione dei parametri

Durante la fase di caricamento di un modello all'interno dell'applicativo, è possibile impostare dei parametri in input che, chiesti in fase di composizione del testo, vengono poi utilizzati all'interno del modello.

L'impostazione di questi parametri si attiva selezionando la linguetta "Parametri" nella maschera di Dettaglio modello e cliccando il tasto "Aggiungi parametro".

Home » Ricerca modelli » Lista Modelli » Dettaglio Modello - Verba se...

Dettaglio modello - "Verba seduta di gara"

Dettaglio Parametri

Codice (*)	MOD_INPUT1
Obbligatorio	<input type="checkbox"/>
Descrizione per inserimento (*)	Inserire MOD_INPUT15
Descrizione	<div></div>
Tipo (*)	<div></div>

Salva Annulla

Viene data in questo modo la possibilità all'utente di inserire i dati di configurazione del parametro che si sta creando descritti singolarmente qui di seguito:

- **Codice (*):** lasciare il codice proposto o impostare un codice alfanumerico per il parametro. Lo stesso codice inserito sarà lo stesso richiamato all'interno del modello.
- **Obbligatorio:** spuntare la casellina nel caso in cui il parametro è obbligatorio per la corretta produzione del testo. Nel caso, l'applicativo non permette di proseguire e propone un avviso che ricorda l'obbligatorietà dell'inserimento del dato.
- **Descrizione per inserimento (*):** è la descrizione breve che compare nel momento in cui il dato viene richiesto.
- **Descrizione:** inserire una descrizione estesa libera utile per gli utenti del modello.
- **Tipo (*):** selezionare attraverso il menu a tendina la tipologia del dato in input. In base al tipo selezionato, viene attivato un controllo sul formato del dato in fase di inserimento onde evitare anomalie poi durante la composizione del testo.

I tipi di dato vengono di seguito descritti:

- o **Data:** attiva in inserimento il tastino del calendario per facilitare l'inserimento della data richiesta. E' possibile comunque inserire manualmente la data nei formati standard accettati dall'applicativo
- o **Numero con virgola:** deve essere impostato un numero decimale con separatore ",". Nel caso il dato da inserire non preveda decimali si possono omettere.
- o **Intero:** aspetta in inserimento un numero intero, quindi senza virgola.
- o **Menu:** selezionando questa opzione si attiva un supplementare campo sottostante che permette l'inserimento delle opzioni del menu. Le voci che si intendono inserire nel menu devono essere inserite una di seguito all'altra divise dal separatore "|" ad esempio: Si|No. In fase di composizione appare un menu a tendina con le opzioni impostate.
- o **Stringa:** chiede in ingresso una stringa di caratteri alfanumerici (parole, frasi, ecc...)
- o **Note:** come stringa, ma permette di inserire più righe di testo
- o **Identificativo utente:** passa come valore l'identificativo dell'utente (SYSCON) collegato all'applicazione; è utile se si vogliono filtrare i dati elaborati in base all'utente
- o **Ufficio intestatario:** analogo al precedente; passa il codice dell'Ufficio o Stazione Appaltante (CODEIN) con a cui è collegato l'utente

Cliccando il bottone "Salva" viene registrato il parametro appena inserito e viene riproposta la lista dei parametri già inseriti. A questo punto è possibile inserire un ulteriore parametro sempre con il bottone Aggiungi parametro o procedere con altre operazioni.

8.2 Utilizzo dei parametri all'interno del modello

Nel caso in cui il Modello preveda dei parametri, per poterli gestire all'interno del testo, è necessario utilizzare una speciale funzione che permette di richiamare e operare con tali parametri inseriti dall'utente in qualsiasi punto del modello.

UTILIZZO DEI PARAMETRI

Sintassi: #Mnemonico{X_PAR_codiceparametro}#

Esempio: #Mnemonico{X_PAR_PARAMETRO1}#

Note: *Mnemonico* è la variabile della stessa tipologia del parametro (numero, stringa o data) e serve solo come appoggio per passare il valore inserito. Il codiceparametro è il codice che viene assegnato dall'utente nelle impostazioni del Dettaglio Modello all'interno dell'applicativo web in fase di aggiunta dei parametri.

In questo modo viene richiamato, all'interno del testo, il valore che è stato inserito dall'utente in fase di composizione.

Si consideri inoltre che per i parametri di tipo menu verrà restituito il numero dell'opzione scelta.


Esempio:

I parametri previsti per il modello sono:

PARAMETRO1: una data di riferimento

PARAMETRO2: un menu con tre opzioni

Parametri per la composizione

Inserire la data di riferimento	<input type="text" value="23/08/2007"/>	 (GG/MM/AAAA)
Vuoi mandare la comunicazione	<div><div></div><div>scrivi al legale rappresentante</div><div>scrivi al direttore tecnico</div><div>entrambi</div></div>	<div>Torna a elenco modelli</div>

Il file sorgente è così impostato:

```
@#CODIMP#  
  
Si informa la ditta #NOMIMP#, che  
@assegna a un TOT il valore del PARAMETRO2  
@ \#CAPIMP{X_PAR_PARAMETRO2}#\++1\  
@  
[IF] %#TOT1#_EQ_2%  
#NOMDTE#, direttore tecnico  
[ELSE]  
#NOMLEG#, legale rappresentante  
[ENDIF]  
che entro il giorno #DISCIF{X_PAR_PARAMETRO1}#...
```

Il risultato del file composto, nel caso si scelga l'opzione 2 del menu, sarà:

```
Si informa la ditta Fedele Costruzioni snc che  
Verdi Giuseppe, direttore tecnico  
entro il giorno 20.05.2007...
```