

## “Compositore Java” versus “Compositore C”

Vengono qui evidenziate particolarità e comandi non più supportati dal nuovo compositore java.

### **Formato RTF, DOCX e ODT**

Il nuovo compositore ha un ottimo supporto dei formati docx e odt con cui vengono superate varie limitazioni dei file RTF, in particolare non è afflitto dai tipici problemi di decodifica dei file RTF che spesso comparivano in caso di formattazione della sintassi.

Il nuovo compositore supporta anche il formato RTF, tuttavia non è certificato il funzionamento con tale formato; va evitato l'uso di questo formato con il nuovo compositore.

### **Relazioni implicite e JOIN**

Il nuovo compositore riconosce la sequenza di mnemonici quando appartenenti ad entità padre – figlio sono se posizionati nell'ordine gerarchico e se tale relazione è stata pre-definita nel sistema. In caso di problemi esplicitare sempre le JOIN per navigare da un'entità all'altra.

### **Nuova sintassi per i costrutti IF e LOOP**

Il nuovo compositore è retro-compatibile, cioè supporta la sintassi del compositore C.

È possibile quindi usare la nuova o la vecchia sintassi.

Nuova sintassi per i costrutti IF e LOOP.

```
%I1 %condizione%
  primo if
%E1
  else del primo if
%I2 %condizione%
  if nidificato
%F2
%F1
```

alla nuova notazione

```
[IF]%condizione%
  primo if
[ELSE]
[IF]%condizione%
  if nidificato
[ENDIF]
[ENDIF]
```

Con questa notazione non è più necessario indicare il numero di nidificazione che viene a perdere di significato. per migliorare la lettura si può utilizzare anche una nidificazione in questo modo

```
@ [IF] %condizione%
  primo if
@[ELSE]
@   [IF] %condizione%
    if nidificato
@   [ENDIF]
```

```
@[ENDIF]
```

Per quanto riguarda i loop si passa dalla vecchia notazione:

```
$$IND=1,100  
    riga da stampare  
$$IND
```

alla nuova notazione

```
[FOR]IND=1,100  
    riga da stampare  
[NEXT]IND
```

Per tutti i comandi vale la possibilità di distanziarli dall'inizio riga con il carattere @ seguito da spazio o tab, in caso di loop nidificati possiamo avere quindi

```
@ [FOR]IND=1,100  
    riga da stampare  
@   [FOR]I=100,1,-1  
    riga interna da stampare  
@   [NEXT]I  
@ [NEXT]IND
```

### ***JOIN implicita***

Non è più supportata la JOIN implicita, cioè quella che prevede di passare il valore della chiave al mnemonico.

Esempio: #NOMIMP (STR1) #

Il "Compositore Java" restituisce <vuoto>

### ***Invio mail***

Il comando [MAILTO] non è più supportato.

### ***Concatenazione stringhe***

Non è più 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"
```

## ***Mnemonici consecutivi***

Non è possibile riportare nel modello due “cancelletti” consecutivi

Esempio: #STR1##STR2#



Si risolve così:

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

E si riporta nel modello #STR3#

## ***Registrazione dati***

Non sono più supportate le istruzioni per l’aggiornamento della base dati

§&#STR1#&#MNEMONICO# oppure “{X\_EXECAGG}”

## ***Comandi di inclusione testo***

Il “Compositore Java” non supporta la gestione dei comandi “INCLUDE” e “INCLUDEMOD”.

## ***Conteggio lunghezza del contenuto di un mnemonico***

Non è più supportato il comando LENGHT che restituiva il numero di caratteri relativi al dato del mnemonico

```
[LENGTH]#nome_mnemonico#,#TOTnn#
```

## ***Richiami dopo la fine del corpo (#FINETESTO#):***

Non è più supportata la gestione della sezione FINETESTO che permetteva di indicare delle costanti, dei mnemonici o delle operazioni non direttamente nel loro punto di utilizzo, ma nella riga n-esima dopo fine del corpo.

## ***Funzioni speciali***

Non sono supportate le funzioni “speciali” seguenti:

- CO, UL, PR, NO  
Es.: #CO@NAPPAL#, riportava il numero delle occorrenze dell’entità a cui appartiene il mnemonico.  
Es.: #UL@NAPPAL#, riportava il contenuto del campo dell’ultima occorrenza prelevata.  
Es.: #PR@NAPPAL#, riportava il contenuto del campo della prima occorrenza prelevata.  
Es.: #NO@NAPPAL#, riportava il numero dell’occorrenza corrente dell’entità a cui appartiene il campo.
- Comandi C  
Non sono più supportati i comandi “CEldasoft”, ovvero il comando [EXECE]
- Comandi XML  
Non sono più supportati i comandi XML (es. #STR1{X\_XMLOpen}#, #STR2{X\_XMLGoto}#, ecc.)

## ***Debug***

È possibile configurare la webapp WSCompositore con il livello di debug da INFO e DEBUG (log4j.properties oppure agendo nella variabile CATALINA\_OPTS\_LOG\_LEV del container) per ottenere una tracciatura fine della composizione dei modelli.

Il nuovo compositore java effettua la tracciatura direttamente in database:

- nella tabella **rpa\_session** memorizza un record per ogni sessione, cioè ogni volta che si esegue una composizione viene assegnata una sessione con un determinato "id"
- nella tabella **rpa\_log** vengono tracciate tutte le operazioni effettuate dal compositore, per ogni sessione "id\_session", per ogni riga di codice analizzata del modello (colonna "text"), con annotate le eventuali anomalie rilevate (colonna "error\_stack\_trace")

Per estrarre i dati di log dell'ultima composizione in corso (o appena eseguita):

```
select * from rpa_log where id_session = (select max(id) from rpa_session)
```

Prestare attenzione a riportare il livello di debug ad INFO terminate le operazioni di analisi per non occupare spazio inutilmente e rallentare le performance.

## ***Esecuzione del nuovo compositore Java***

**Accertarsi che l'utente esecutore del tomcat nel quale viene deployato WSCompositore, esegua java da riga di comando** (provare il comando java -version in un shell aperta utilizzando tale utente).

Nel caso in cui non vada in esecuzione, accertarsi di impostare la variabile d'ambiente JAVA\_HOME e poi la variabile PATH in modo da contenere il riferimento a JAVA\_HOME/bin, quindi riprovare l'esecuzione di java.

## ***Licenza d'uso***

Si precisa che il nuovo compositore java include delle librerie acquisite in licenza d'uso da Aspose (<https://www.aspose.com/>).

Queste vengono distribuite ai clienti Maggioli SpA come run-time all'interno del "compositore java" nell'ambito della licenza acquisita da Maggioli SpA.

ATTENZIONE: nel caso di riuso, quando il prodotto non sia più in licenza d'uso Maggioli SpA, l'utente finale è tenuto all'acquisto delle licenze Aspose.

Nel caso considerare che il compositore C rimane disponibile, se pur deprecato.