

# 2802 - ULL IKEA con NiceLabel

[Link al progetto originale Messersì](#)

## ANALISI AgentStampa

### AGENTPRINTMSI

Richieste che effettua ad AGENTSTAMPA:

- 1) recprint -> ottiene il rec da stampare e lo salva sul database
- 2) printRecFromId -> ottiene il rec da stampare tramite query a etichetw
- 3) stampamanu -> effettua la stampa

**AGENTSTAMPA** (/code/cpp/module/service/palctrl/stampa/agentstampa.cpp)

L'agente si occupa di eseguire la stampa di etichette IKEA. Può stampare in due modalità:

1. stampa automatica: richiesta dall'agente java
2. stampa manuale: richiesta dall'interfaccia

Il p.to 1) e 2) sono eseguiti attraverso la tell, rispettivamente:

1. request=**stampaauto**, p1=rec(codlotp), ritorna true se la stampa è avvenuta correttamente
2. request=**stampamanu**, p1=rec(codart,peso,npz,yyww,codlotp,codsup,barcode,npzfmt), ove codart=codice articolo Ikea, peso=peso del bancale, npz=numero pezzi, yyww=anno/settimana, codlotp=codice lotto bancale; il nome del rec (rec.name=path\_file) è il path completo del file del layout da stampare; nel nume va indicato il numero di etichette da stampare

Il rec da utilizzare per la stampa manuale può essere richiesto all'agente stesso attraverso la ask con:

1. request=**recprint**, p1=rec(codart,peso,npz,codlotp,codsup), p2=path\_del\_file l'agente restituisce il rec con tutti i dati compilati

### RECPRINT

```
if(msg->request=="recprint"){
    rv = recprint(IREC_GETVALUE(p1, "codart"), IREC_GETVALUE(p1, "codsup"),
atof(IREC_GETVALUE(p1, "pesotot"), atof(IREC_GETVALUE(p1, "peso")),
atoi(IREC_GETVALUE(p1, "npz"), IREC_GETVALUE(p1, "codlotp"), p2->value(),
atoi(IREC_GETVALUE(p1, "xinterfa"), atoi(IREC_GETVALUE(p1, "yinterfa")),
atof(IREC_GETVALUE(p1, "stackcap")));
}
```

**P1(REC):** codart, codsup, pesotot, peso, npz, codlotp, xinterfa, yinterfa, stackcap

**P2 (String):** layout

1. Ottiene il numeratore attraverso una getnume della tabella 'ikeapiano' sul campo 'codice' per ottenere un codice per il PIANO del pallet
2. Prepara un REC con i seguenti campi:
  - 0, "codart"
  - 1, "peso"

- 2, "npzfmt"
- 3, "yyww"
- 4, "codlotp"
- 5, "codsup"
- 6, "barcode"
- 7, "npz"
- 8, "idetic"
- 9, "layout"
- 10, "tm"
- 11, "pesotot"
- 12, "xinterfa"
- 13, "yinterfa"
- 14, "stackcap"

3. Popola il REC con i valori ottenuti da parametro. Aggiunge 'npzfmt' con il npz formattato a 3 cifre e 'yyww' basata sulla data odierna
4. Elabora il codice del piano nel seguente modo:

```
int totlen=17;
int bcforlen=m_bcforinfo->size();
char fmt[51];
sprintf(fmt, "%0%dd", totlen-bcforlen);
sprintf(copas, fmt, copa);
```

5. "idetic" → numeratore ottenuto nel punto 1
6. Prepara il codice a barre in questo modo:

```
sprintf(s, "%s%s", m_bcforinfo->value(), copas);
```

dove 'm\_bcforinfo' -> dato fisso del fornitore per il codice a barre EAN128

7. !!! FA UNA DOWNLOAD DEL REC SUL DATABASE !!!
8. Imposta il nome del REC con il valore del 'layout'
9. Ritorna il REC creato

## RECPRINTFROMID

```
if(smsg->request=="recprintfromid"){
    rv = recprintfromid(atoi(IREC_GETVALUE(p1, "idetic")), p2->value());
}
```

**P1(REC):** idetic **P2 (String):** layout

1. Mi estraggo dal database una riga di etichetw tramite il codice di idetic ricevuto nel p1
2. Imposta il nome del REC con il layout ricevuto nel p2
3. Ritorna il REC creato

## STAMPAMANU

```
if(smsg->request=="stampamanu"){
    rv = excprint(p1, smsg->nume);
}
```

Presente un #define che distingue le funzionalità della funzione excprint

## PRINT\_CODESOF

### TRUE:

1. Connessione con agentRemoteCS per stampa con codesoft
2. Richiesta del REC per la stampa all'agentRemoteCS
3. Popolazione del REC:
  1. filename = nome del REC p1
  2. codedb = p1["identific"]
  3. qtalotti = nume ricevuto
  4. qtacopie = "1" FISSO
  5. printer = parametro agente m\_printer
4. Richiesta di stampa inviata ad agentRemoteCS con request "print" passando il REC creato come parametro
5. Impostato un wait di 250 ms
6. Ritorna un booleano in base alla riuscita o meno della stampa

### FALSE:

1. Apertura\scrittura del file con il nome del file uguale al nome del REC ricevuto nel p1
2. Determina la dimensione del file, aggiunge 512 byte per eventuali sostituzioni, e alloca un buffer per contenere i dati del file.
3. Legge il contenuto del file nel buffer e chiude il file.
4. Preparazione dei tag (5 tag)
5. Crea un oggetto CTag con un numero di tag pari al numero di campi di rp meno 2.
6. Imposta i tag e i valori corrispondenti ottenuti dai campi di rp.
7. Apre il file "/code/etichetta.txt" e scrive il contenuto del buffer su questo file
8. Crea un oggetto IConnSock e tenta di connettersi a un socket utilizzando m\_port e m\_ipaddr
9. Se la connessione ha successo, invia il contenuto del buffer n (nume ricevuto) volte, attendendo 200 millisecondi tra ogni invio.
10. Libera la memoria del buffer e dell'oggetto CTag.
11. Restituisce true se almeno un invio è avvenuto con successo, altrimenti false.