Allegato 2b

NSIP
Manutenzione MAC e MEV

Sistema di sviluppo

Sistema di sviluppo software multiambiente

MANUALE PER IL PROGRAMMATORE

Documento	:	ISYDPRG0
Aggiornamento	:	0
Data	:	13 febbraio 1995

Sommario

Generalità

Una parte consistente dello sviluppo di applicazioni o di sistemi informativi è costituita dalle fasi di realizzazione e test. Mentre molti prodotti CASE coprono in modo eccellente le fasi iniziali del progetto (di analisi funzionale di dati e processi), per le fasi finali (dall'analisi tecnica in poi) esistono una molteplicità di tecnologie disponibili ognuna delle quali punta a raggiungere uno o più obbiettivi particolari.

Con la realizzazione di questo sistema di sviluppo si è voluto risolvere in modo autonomo e sostanzialmente diverso questa parte della metodologia.

In particolare si vogliono raggiungere i seguenti principali obbiettivi:

- Riduzione del tempo di sviluppo dei programmi (codifica e test)
- Elevata standardizzazione (omogeneità) dei programmi sia dal punto di vista strutturale, sia dal punto di vista formale
- Interfacciamento ed integrazione dei servizi disponibili (es. menù, help on line, ecc.)
- Trasportabilità delle applicazioni su ambienti diversi
- Utilizzazione di prodotti di base standard pur non vincolandosi ad alcuno di questi
- Integrazione stretta con il metodo di scrittura delle specifiche di programma
- Integrazione nella metodologia complessiva di sviluppo dei sistemi
- Adattabilità dell'interfaccia utente alla disponibilità del mercato

Collocazione nel ciclo di sviluppo del software

Il sistema di sviluppo presente si propone come fase finale del ciclo di sviluppo del software.

L'uso eventuale di particolari strumenti CASE dovrà essere limitato alla parte "alta" dell'analisi, fino alla scomposizione del progetto nei dati, nelle funzioni e nei programmi che realizzano quest'ultime.

A questo punto dovrà essere effettuata la stesura delle specifiche tecniche di realizzazione. Queste verranno in parte preparate utilizzando il sistema di sviluppo stesso seguendo il modello descritto nel documento XXXDSPE0, che, per quanto riguarda le specifiche dei programmi, prevede la generazione automatica delle stesse a partire dalle *Specifiche di base* e dai *Blocchi di documentazione* (vedi paragrafi appositi).

Concetti base

Nella spiegazione dei concetti base utilizzati nel sistema di sviluppo si farà riferimento principalmente a sistemi interattivi, prima di tutto per la maggiore complessità di queste realizzazioni, e poi perché la realizzazione delle applicazioni tenderà sempre più ad essere effettuata secondo questa modalità.

Ragionando in maniera astratta si può dire che i programmi possono essere classificati in un numero abbastanza ristretto di categorie. Ognuna di queste categorie richiederà la presenza di determinate funzionalità all'interno dei programmi, cioè di particolari modi di agire a fronte di condizioni fornite dall'utente o dai dati. Questo vuol dire che in ognuna di queste tipologie di programmi dovranno esistere dei blocchi di istruzioni pressoché uguali per realizzare analoghe funzioni. D'altro canto ogni categoria di programma presenterà anche delle modalità operative identiche.

Per esemplificare, un programma che realizza una funzione di acquisizione dati sarà sempre basato su un ciclo che prevede l'invio del pannello, la ricezione dei dati digitati, il loro controllo con l'emissione dei messaggi di errore e la ripetizione del ciclo in caso di errore o l'esecuzione dell'acquisizione in caso di dati validi.

La differenza fra l'aspetto del programma ed un altro appartenente alla stessa categoria sarà quindi limitata al pannello di acquisizione, all'I/O, a controlli particolari sui dati, a funzioni aggiuntive specifiche.

Le **specifiche** che dovranno essere fornite per la realizzazione dei programmi possono essere quindi limitate alla definizione della categoria di appartenenza del programma stesso e di tutti gli aspetti particolari che sono richiesti.

Le caratteristiche generali dei programmi appartenenti ad una categoria incideranno sulla sua struttura realizzativa e, tradotte in istruzioni, saranno lo **schema scheletro** su cui si dovranno appoggiare le specifiche per realizzare il programma applicativo completo.

Ci sono quindi tanti schemi scheletro per quante sono le categorie di programmi necessarie e ciascuno conterrà tra l'altro in forma standard, completa, ordinata ed efficiente tutte le istruzioni che altrimenti dovrebbero essere riscritte, e spesso anche ritestate, in ogni programma applicativo.

Poiché solamente nello schema sono anche presenti le particolarità operative legate agli ambienti di esercizio diversi, per ogni categoria ci sono tanti schemi scheletro per quanti sono i sistemi operativi previsti dal sistema di sviluppo.

Gli schemi scheletro comunque, sono scritti una tantum e rimangono sempre gli stessi.

La fusione tra lo schema scheletro e la specifica scritta dal progettista viene effettuata da un **programma generatore** che leggendo in input scheletro e specifica, produce in output il programma COBOL da compilare.

Può anche accadere che una stessa specifica possa essere fusa con più schemi diversi, per produrre differenti programmi, come nell'esempio di un programma di stampa che può 'girare' in ambiente batch o in on-line.

Le modalità di funzionamento del programma generatore sono desumibili dalla descrizione, riportata nel Manuale di sistema (ISYDSYS0), delle funzioni presenti nello schema scheletro.

Contenuto del file specifiche

Il file specifiche è diviso logicamente in due parti: la prima è riservata alla definizione delle variabili ed al disegno dei pannelli e dei tabulati (*zona delle specifiche di base*), la seconda è riservata alle routine specifiche del programma o comunque a blocchi di istruzioni (*zona delle specifiche di routine*).

Il passaggio dalla prima alla seconda parte è individuato dalla riga contenente la parola **&ENDVAR**.

In entrambe le porzioni del file di specifiche possono essere inserite righe di commento: esse sono caratterizzate dalla presenza dei simboli /* a colonna 1 e 2 della medesima riga.

```
/*      Esempio di riga di commento
```

Qui di seguito vengono descritte le tipologie di righe che possono essere presenti nel file specifiche. La presenza o meno di un blocco è legata alla tipologia del programma in quanto il contenuto informativo del blocco può non avere significato in alcune tipologie di programmi e quindi non avere corrispondenza nello scheletro.

ZONA DELLE SPECIFICHE DI BASE

È la parte iniziale del file specifiche. È suddivisa in una serie di blocchi riservati alla definizione di aspetti particolari del programma.

Blocco di definizione delle variabili

In questo blocco iniziale vengono forniti i valori per le variabili di base del programma. L'associazione del valore alla variabile viene ottenuto ponendo il codice della variabile **&xxxxxxx** all'inizio di una riga ed il valore ad essa associato nella stessa riga a partire dalla *decima* posizione.

Le variabili base sono raggruppabili in categorie secondo le loro funzioni e caratteristiche. Di seguito è riportato, categoria per categoria, il significato di ciascuna variabile:

Variabili di documentazione e nomenclatura

&LANG

Linguaggio delle istruzioni delle specifiche

&PROGET

sigla identificativa del progetto, alfabetica di tre lettere

&PROGRAM

sigla identificativa del programma all'interno del progetto, alfanumerica di tre caratteri

&TRAN

sigla identificativa della transazione eventualmente associata al programma

&SCHEMA

sigla identificativa dello schema (principale) previsto per la specifica in oggetto. Le sigle per ciascuno degli schemi disponibili sono:

- B PRBISY
- L PRTISY
- M MAPPA
- N MNTISY
- T MLSISY
- V VISISY
- P MPNISY
- C MCOISY

&VERSPE

versione progressiva della specifica (numero da 0 a 9)

Va sottolineato che il programma generatore, una volta invocato con l'apposita procedura, crea automaticamente un sorgente il cui nome viene costruito sulla base dei valori indicati nelle quattro variabili precedenti, secondo il seguente meccanismo:

nome-programma = &PROGET+&SCHEMA+&PROGRAM+&VERSPE.

Il nome della mappa (quando serve) viene costruito dinamicamente dal generatore:

nome-mappa = &PROGET+'M'+&PROGRAM

e messo nella variabile &MAPNAME

&PGMNAME

specificando il nome del programma in questa variabile, esso viene assunto come prevalente rispetto a quello generato automaticamente.

&VERSCH

questa variabile, rappresentata da una cifra di due posizioni, è necessaria per generare applicazioni già sviluppate con una versione dello schema passata e la cui funzionalità sarebbe compromessa da una rigenerazione con uno schema aggiornato. Il generatore controlla la corrispondenza tra il valore di questa variabile e di quella passata come parametro alla procedura di generazione: se non sono uguali emette un messaggio di avvertimento

&PGMDES

definizione breve (massimo 40 caratteri) del programma

Variabili per formato stampe

&PAGELNT

Dimensione fisica in righe di una pagina di tabulato per i programmi di stampa

&TESTATA

Specificando il numero di colonne del tracciato di stampa consente di centrare rispetto ad esso una testata standard gestita dal sistema. Se specificato NO, la testata generata automaticamente non viene prodotta

Variabili di ambiente

&SQLDS

genera o meno gli statements necessari alla definizione della DECLARE SECTION e ad altri aspetti che distinguono applicazioni che utilizzano il dbms SQL/DS da quelle che utilizzano il DB2

&DB2

ha le stesse funzioni del precedente, evidentemente invertendo i dbms

&ISYRIPM e &ISYRCCD

specificano i nomi effettivi dei programmi di cui deve essere effettuata la CALL e che, in ambiente VSE sono rispettivamente le routine Assembler ISYRIPM0 E ISYRCCD0 le quali effettuano un LOAD dinamico dei programmi ISYBIPM0 e ISTBCCD0), in ambiente MVS sono direttamente i programmi COBOL ISYBIPM0 e ISYBCCD0 dei quali viene effettuata una CALL dinamica.

Switches funzionali

&ONLYVIS

permette di indicare in specifiche del tipo MLSISY se la lista generata deve essere in sola visualizzazione oppure consentire aggiornamenti

&I-O-CAN

inibisce/abilita il tasto di cancellazione in programmi generati da specifiche di tipo MNTISY

&I-O-NXT

inibisce/abilita il tasto di scorrimento archivio in programmi generati da specifiche di tipo MNTISY

&INSERT

inibisce/abilita l'inserimento nei programmi di maintenance

&UPDATE

inibisce/abilita l'aggiornamento nei programmi di maintenance

&PARAMET

consente di escludere la fase di acquisizione dei parametri di attivazione qualora questi non siano previsti dalla particolare applicazione di MNTISY

&UCTRAN

effettua o meno la conversione dei caratteri minuscoli in maiuscolo nelle transazioni conversazionali (sviluppate tramite schema MCOISY)

Altre variabili

&TYPE

definisce, nello schema MPNISY, la sequenza dei programmi che fanno parte dello stesso gruppo, e può valere FIRST, SECOND,..., LAST

&NXTPGM

definisce, nello schema MPNISY, il programma che deve essere attivato successivamente in sequenza

Queste righe, come esempio, sono generalmente le prime in assoluto del file specifiche.

```
&LANG          COBOL
&PROGET        IDS
&PROGRAM       002
&SCHEMA        N
&VERSPE        0
&PGMDES        AGGIORNAMENTO STRUTTURA DOCUMENTAZIONE
&SQLDS               NO
```

Per ogni progetto esiste un file nell'ambiente di sviluppo che contiene l'impostazione di default di alcune delle variabili suddette. Tale file viene letto dal generatore prima del file 'specifiche' dove il valore default delle variabili può sempre essere sostituito da una ridefinizione delle stesse. Normalmente i valori di default sono quelli definiti di seguito, ma è opportuno verificarli con il responsabile del sistema.

&SQLDS YES

&DB2 NO

&ONLYVIS NO

&I-O-CAN YES

&I-O-NXT YES

&INSERT YES

&UPDATE YES

&PARAMET YES

&UCTRAN NO

&ISYRIPM ISYRIPM0

&ISYRCCD ISYRCCD0

Blocchi di definizione dei dati

Il blocco di righe riservato alla definizione dei dati è limitato a seconda dei casi dalle seguenti coppie di parole:

definizione di un pannello generico e dei dati relativi

&MAPFMT

&ENDFMT

definizione di un pannello di visualizzazione a lista o di un tabulato e dei dati relativi

&LSTFMT

&ENDLST

definizione dei dati per un pannello di acquisizione standard

&INPUT

&ENDINP

Le righe all'interno di questo blocco possono essere ulteriormente suddivise in generale in tre sottoblocchi:

- il primo per la *definizione degli attributi* e cioè dei caratteri necessari per la loro rappresentazione (presente solo nel caso di pannelli generici)
- il secondo per il *disegno del pannello* (assente nel caso di acquisizione standard)
- il terzo per la *descrizione dei campi* (sempre presente)

Questi sottoblocchi concorrono, tra l'altro, alla generazione della mappa, che sarà prodotta *fondendo* il file specifiche con uno schema già appositamente predisposto (denominato MAPPA).

Definizione degli attributi

È racchiuso tra due schede **)DA**. Contiene la definizione dei caratteri che devono rappresentare gli attributi dei campi di mappa, con il tipo di protezione e la intensità luminosa.

Per ogni attributo devono essere definite:

- protezione

  A per askip

  P per protetto

  U per non protetto;

- luminosità

  B per luminoso

  N per normale

  D per dark.

Si può usare qualsiasi carattere per definire un attributo, ma si consiglia di servirsi dei caratteri speciali della tastiera (ad es.: à, è, ì, ò, ù, ç, |, ^, @, ecc.), poiché questi non sono normalmente utilizzati sui pannelli.

Disegno pannello

È racchiuso fra due schede **)FM**. Contiene il disegno del pannello su cui è basato il programma, con tutti i campi normalmente preceduti da uno degli attributi che sono stati precedentemente definiti e con i campi variabili individuati dal carattere "_".

I campi, così definiti, possono essere contigui o, se si tratta di campi variabili, separati da almeno due spazi. Ciò è dovuto alla necessità di interporre fra i due campi lo stopper field e l'attributo.

Tutti i campi, sia quelli fissi che quelli variabili, non possono essere più lunghi di 132 caratteri.

La somma del numero di campi fissi e del numero di campi variabili non può comunque superare il limite massimo di 400.

All'inizio di questa zona può essere utilizzata l'istruzione speciale:

.CO x **carattere di continuazione**

dove x è un qualsiasi carattere che si vuole utilizzare come riempimento per far considerare unitariamente i campi fissi eventualmente formati da più parti.

Poiché le prime ed ultime due righe di ogni pannello sono riservate ed utilizzate in forma standard, la prima riga disegnata corrisponderà alla terza riga di pannello ed *ogni pannello potrà comprendere al massimo venti righe.*

Tutti i campi definiti devono essere racchiusi, per gruppi, dalla iniziale e finale di una delle seguenti istruzioni speciali, a seconda del tipo di transazione e quindi dello schema che tali specifiche dovranno utilizzare.

.RG nnnn **riga iniziale di mappa**

dove nnnn è un numero di riga di quattro cifre (es. 0010) che permette di iniziare la definizione del pannello ad una riga diversa dalla terza evitando di lasciare righe bianche nel file specifiche. Questa istruzione può essere utilizzata anche nella parte di descrizione dei campi di acquisizione dei dati in input (vedi esempio).

.HP 0001 xxxx **definizione righe di testata**

dove xxxx è il codice identificativo del blocco di testata ("HEAD" nelle specifiche di programmi di visualizzazione a lista); le righe di testata sono terminate dalla scheda **.HP**

.DT 0001 xxxx **definizione righe di dettaglio**

dove xxxx è il codice identificativo del blocco di dettaglio ("DETT" nelle specifiche di programmi di visualizzazione a lista); le righe di dettaglio sono terminate dalla scheda **.DT**

.MP nnnn **definizione di nnnn elementi uguali di righe (moltiplicatore)**

dove nnnn è il numero di volte che gli elementi contenenti le righe di seguito definite, vengono sviluppati sul programma generato. Le righe definite sono terminate dalla scheda **.MP**. Tale moltiplicatore deve essere utilizzato nelle specifiche che prevedono di utilizzare schemi di maintenance a lista. Il primo campo della prima riga non può partire ad una colonna inferiore a 3, perché automaticamente viene sviluppato un campo di un carattere per permettere la selezione di uno degli elementi per la funzione di cancellazione o altro.

Inoltre, solamente per la definizione di tabulati, valgono le seguenti istruzioni speciali:

.EG 0001 xxxx **definizione righe di gruppo (riepilogo parziale)**

dove xxxx è il codice identificativo del gruppo; le righe di gruppo sono terminate dalla scheda **.EG**. I gruppi sono utilizzati prevalentemente per realizzare riepiloghi parziali in tabulati o liste, al verificarsi di rotture negli elementi secondo i quali tali liste sono ordinate. Tali rotture possono avvenire a vari livelli (max 9) e i campi sui quali le stesse vengono controllati sono tra loro raggruppati adeguatamente nel sottoblocco di descrizione dei campi.

.EL 0001 xxxx **definizione righe di fine lista (riepilogo finale)**

dove xxxx è il codice identificativo della lista; le righe di fine lista sono terminate dalla scheda **.EL**. Tali righe sono utilizzate prevalentemente per realizzare riepiloghi finali di tabulati e liste.

Descrizione campi

È racchiuso fra due schede **)DF**. Ogni riga contiene la denominazione del campo variabile corrispondente nel disegno del pannello. La corrispondenza è posizionale, cioè la prima denominazione è relativa al primo campo variabile del pannello, la seconda al secondo e così via, considerando le posizioni consecutive del pannello in modo naturale da sinistra a destra e dall'alto in basso.

Alla fine dei campi corrispondenti uno a uno con quelli del pannello definito, possono essere definiti altri campi.

Per ogni campo viene definito (secondo il tracciato chiarito dagli esempi che seguono):

- **nome**
- **caratteristiche**
  - *obbligatorietà del campo* su mappa oppure *campo di Working Storage* che viene gestito come un campo di mappa per quanto riguarda il salvataggio nella conversazionalità della transazione oppure *campo chiave* o *campo di gruppo* nella gestione di liste
    - **O** campo obbligatorio
    - **W** campo di WS
    - **K** campo chiave nelle liste
    - **G** campo di gruppo nelle liste
  - *appartenenza* del campo a blocchi particolari
    - **W** campo di Working Storage (senza salvataggi)
    - **S** campo di una declare section SQL
    - **R** campo presente in tracciato record o altra copy
  - modalità di definizione del *controllo formale*
    - **D** controllo tramite dizionario dati
    - **F** controllo secondo la funzione indicata
    - **M** campo da spostare con un semplice MOVE
    - **Z** descrizione da tabella di decodifica
  - routine di *controllo logico*, che il progettista deve predisporre ed il cui nome è 800-CTRL-nome-del-campo, o *numero di livello* nei raggrupamenti di campi
    - **L** attivazione automatica di suddetta routine
    - **n** numero di livello (max 9)
- **funzione di controllo** se il controllo formale è F o il nome del campo sul dizionario (se è stata indicata la D) ed il nome specificato non coincidono.

- **identificatore** del campo, cioè stringa di caratteri che deve comparire sul pannello di richiesta dei dati di attivazione come descrizione del campo stesso.

Caratteristiche

Mentre i campi definiti appartenenti a Working Storage (W), o a Declare Section SQL (S) vengono generati nella loro definizione COBOL (sia nella tipologia che nelle dimensioni) ed inseriti nelle zone opportune, i campi di tracciato record o copy (R) vengono considerati come disponibili nella copy di cui sarà cura del progettista richiedere l'inserimento nel programma indicandola nella zona del file specifiche a ciò destinata. Il progettista dovrà tenere presente che i nomi di questi campi saranno utilizzati per creare i campi di interfaccia verso il pannello di acquisizione.

Questo avverrà semplicemente generando un altro campo con le stesse caratteristiche di quello originale ed avente un nome composto dal nome del campo originale con il suffisso "-I". In questi campi il progettista potrà trovare i dati, in formato interno, da utilizzare in seguito per tutti gli usi che riterrà opportuno farvi.

I campi di working storage (indicatore W nella prima posizione delle opzioni) devono essere definiti sempre dopo quelli corrispondenti a campi di mappa.

Definendo un campo con questa opzione, il valore che esso contiene viene salvato, al pari dei campi di mappa, in una apposita coda utilizzata per il salvataggio delle informazioni, ed e' pertanto disponibile nell'arco dell'intera unita' logica di lavoro.

I campi la cui funzione di controllo è di tipo tabellare TBxx, e che sono quindi costituiti normalmente da codici, possono essere seguiti dalla rispettiva descrizione, che si vuole mostrare sulla mappa. In questo caso il campo codice sarà immediatamente seguito da un campo con un controllo formale Z e una funzione di controllo del tipo ALnn, dove nn rappresenta la lunghezza del campo contenente la descrizione.

Quando per un campo si specifica l'opzione M come modalità di controllo formale, non viene richiesta per esso alcuna conversione da parte del programma di controllo, ma viene operata unicamente una MOVE dal campo interno al campo esterno. In particolare, questa opzione può essere utilizzata per quei campi alfanumerici di output (in visualizzazione o in stampa) che non necessitano di controllo formale ma che possono non essere valorizzati. In questo modo il campo non viene editato, e conseguentemente nel caso sia valorizzato a SPACE non vengono visualizzati o stampati i caratteri '_'.

Funzioni di controllo

Funzione di acquisizione per campo alfanumerico

Questa funzione viene richiesta con il codice "ALll" dove "ll" rappresenta la lunghezza del campo.

Questa funzione è quella più semplice in quanto effettua unicamente l'acquisizione del campo di ingresso dopo aver convertito eventuali LOW-VALUE o caratteri "_" in spazi.

I campi ai quali è associato questo controllo devono avere PICTURE X(ll).

Funzione di gestione per campi numerici

Questa funzioni effettuano il controllo e l'editing per campi numerici qualunque sia il loro formato.

Viene effettuato il controllo di numericità, il controllo del numero di cifre rispetto alla lunghezza del campo interno, vengono trattati i decimali.

Il numero può essere scritto in qualunque punto del campo di ingresso e con qualsiasi numero di decimali (purché non superiore a quello previsto).

L'editing prevede la presenza del carattere "," come separatore dei decimali e del "." per le migliaia.

Per le funzioni specifiche viene inoltre prevista la presenza del segno come carattere iniziale del numero da controllare o editato. Il segno viene visualizzato solo nel caso di valori negativi, mentre si assume per default che i valori senza segno sono positivi.

In particolare le funzioni sono le seguenti:

NZ*lld* Acquisizione/Controllo di un campo numerico con trasformazione in formato ZONED

NP*lld* Acquisizione/Controllo di un campo numerico con trasformazione in formato PACKED

NB*lld* Acquisizione/Controllo di un campo numerico con trasformazione in formato BINARIO

SZ*lld* Come per il controllo NZ con in più la gestione del segno.

SP*lld* Come per il controllo NP con in più la gestione del segno

SB*lld* Come per il controllo NB con in più la gestione del segno

In tutti *"ll"* rappresenta la lunghezza del campo in formato interno, cioè il numero dei byte occupati da questo, e *"d"* il numero dei decimali. *"ll"* comprende quindi anche i decimali.

Il numero dei decimali può essere omesso, cioè può assumere il valore SPAZIO. In questo caso viene assunto che il numero dei decimali è zero. L'editing di un campo così definito differisce da quello con "0" sullo stesso campo per la mancanza dei punti di separazione delle migliaia.

Nel caso si tratti di un campo decimale di un archivio SQL, la lunghezza 'll' deve essere il numero dispari immediatamente superiore alla lunghezza con cui è stato definito il campo stesso in tabella (a meno che non sia già di lunghezza dispari).

Il segno è visualizzato soltanto nel caso di numeri negativi.

Funzioni di gestione date

Queste funzioni effettuano il controllo e l'editing di campi data qualunque sia il loro formato.

Il controllo data è completo anche della gestione degli anni bisestili.

Il dato in ingresso può essere scritto sia nel formato con le barre che compattato senza le barre. Se nel formato interno è presente anche il secolo la data esterna può essere digitata anche senza di questo, essendo assunto 1900.

I formati interni possibili sono i seguenti:

DT02 Acquisizione/Controllo e Visualizzazione di una data con conversione in progressivo giorni dal 1900

DT04 Acquisizione/Controllo e Visualizzazione di una data con conversione in formato PACKED GGMMAA

DT05 Acquisizione/Controllo e Visualizzazione di una data con conversione in formato PACKED GGMMAAAA

DT06 Acquisizione/Controllo e Visualizzazione di una data con conversione in formato ZONED GGMMAA

DT08 Acquisizione/Controllo e Visualizzazione di una data con conversione in formato ZONED GGMMAAAA

DI04 Acquisizione/Controllo e Visualizzazione di una data con conversione in formato PACKED AAMMGG

DI05 Acquisizione/Controllo e Visualizzazione di una data con conversione in formato PACKED AAAAMMGG

DI06 Acquisizione/Controllo e Visualizzazione di una data con conversione in formato ZONED AAMMGG

DI08 Acquisizione/Controllo e Visualizzazione di una data con conversione in formato ZONED AAAAMMGG

DI10 Acquisizione/Controllo e Visualizzazione di una data con conversione in formato standard ISO (AAAA-MM-GG).

DE10 Acquisizione/Controllo e Visualizzazione di una data con conversione in formato standard EUR (GG.MM.AAAA).

Inoltre il campo CCD-DT-GIORNO restituisce il giorno della settimana corrispondente alla data fornita.

Per i campi di tipo DT02 la PICTURE di definizione deve essere X(2). Per il controllo DT06 e DI06 il formato è S9(6). Il controllo DT08 e DI08 vuole il formato S9(8). Per il controllo DT04 e DI04 il formato è S9(6) COMP-3, mentre è S9(8) COMP-3 per i controlli DT05 e DI05. Infine, per i controlli DI10 e DE10 il formato e' X(10).

Per tutti i formati di data suindicati è inoltre prevista una ulteriore possibilità. Se la funzione è del tipo DTnnSYSTEM, cioè in essa è contenuta la parola SYSTEM, viene trascurato qualsiasi dato presente nel campo e viene utilizzata la data di sistema, portandola nel campo interno nel formato previsto.

Funzioni di gestione tempo

Queste funzioni effettuano il controllo e l'editing di campi contenenti dati rappresentanti tempo in ore, minuti e secondi.

Il dato in ingresso può essere scritto sia separando con il carattere ":" le ore dai minuti che questi dai secondi, che digitando l'intera stringa senza separatori. In ogni caso il tempo deve essere fornito anche con gli zeri non significativi, fornendo cioè 05 e non 5.

I formati interni possibili sono i seguenti:

TM04 Acquisizione/Controllo e Visualizzazione di un'ora con conversione in formato PACKED HHMMSS

TM06 Acquisizione/Controllo e Visualizzazione di un'ora con conversione in formato ZONED HHMMSS.

Per i campi di tipo TM04 la PICTURE di definizione deve essere 9(6) COMP-3. Il controllo TM06 vuole il formato 9(6).

Anche per il campo tempo, come per quello data, esiste la possibilità di ottenere automaticamente il tempo corrente. Se la funzione è del tipo TMnnSYSTEM, cioè in essa è contenuta la parola SYSTEM, viene trascurato qualsiasi dato presente nel campo e viene utilizzato il time di sistema, portandolo nel campo interno nel formato previsto.

Funzioni di gestione tabelle

Queste funzioni permettono di ottenere il controllo di campi di ingresso secondo valori tabellari e di ottenere informazioni legate a questi.

La funzione viene richiesta con il formato TBllxxxxxx dove ll è la lunghezza del campo e xxxxxx è la sigla identificativa della tabella di valori accettabili.

I campi ai quali è associato questo controllo devono avere PICTURE di tipo X.

Con tale funzione viene estratta anche la descrizione del campo che viene eventualmente posta nel campo associato al controllo formale Z.

Funzioni di gestione campi per singolo bit

Questa funzione è stata sviluppata per permettere di utilizzare i singoli bit di un byte come indicatori. In forma generale questa funzione espande / comprime i singoli bit di un campo di un byte in un campo di 8 byte.

La funzione viene richiesta con il formato BT01xxxxxx dove xxxxxx è la sigla identificativa di una tabella contenente il significato dei singoli bit. In questa tabella ad ogni bit viene associato un carattere che lo rappresenta esternamente.

I campi ai quali è associato questo controllo devono avere PICTURE X.

TABELLA RIASSUNTIVA FUNZIONI ISYxCCD0		
funzione	**PICTURE campo record**	**campi interfaccia (CCD-...)**
ALll	X(ll)	-AL-INTERNO
NZll(d)	9(ll-d)V9(d)	-NRd
NPll(d)	9(2*ll-1-d)V9(d) COMP-3	-NPd
SPll(d)	S9(2*ll-1-d)V9(d) COMP3	-NPd
NB01(d)	X	-NBX1-C
SB01(d)	X	-NBX1-C
NB02(d)	9(4-d)V9(d)	-NBd
SB02(d)	S9(4-d)V9(d)	-NBd
NB03(d)	X(3)	-NBX3-C
SB03(d)	X(3)	-NBX3-C
NB04(d)	9(9-d)V9(d)	-NBd
SB04(d)	S9(9-d)V9(d)	-NBd
DT02(SYSTEM)	XX	-DT02 -DTGIORNO -DTMESE
DT04(SYSTEM)	9(6) COMP-3 ggmmaa	-DT04 -DTGIORNO -DTMESE
DT05(SYSTEM)	9(8) COMP-3 ggmmaaaa	-DT05 -DTGIORNO -DTMESE
DT06(SYSTEM)	9(6) ggmmaa	-DT06 -DTGIORNO -DTMESE
DT08(SYSTEM)	9(8) ggmmaa	-DT08 -DTGIORNO -DTMESE
DI04(SYSTEM)	9(6) COMP-3 aammgg	-DI04 -DTGIORNO -DTMESE
DI05(SYSTEM)	9(8) COMP-3 aaaammgg	-DI05 -DTGIORNO -DTMESE
DI06(SYSTEM)	9(6) aammgg	-DI06 -DTGIORNO -DTMESE
DI08(SYSTEM)	9(8) aaaammgg	-DI08 -DTGIORNO -DTMESE
TM04(SYSTEM)	9(6) COMP-3 HHMMSS	-TM04
TM06(SYSTEM)	9(6) HHMMSS	-TM06
TBllxxxxxx	X(11)	-TB-INTERNO -TB-DESCRIZ -TB-CAMPI
BTllxxxxxx	X	-BT-INTERNO -BT-ESPANSO -BT-EDITATO -BT-CAMPI
ER07	non previsto	-ER-CODICE -ER-VARIABILE -ER-MESS -ER-MESS-FULL

Tabella riassuntiva dei tipi di controllo

Controllo tramite dizionario

Qualora nel campo disponibile per la modalità di controllo formale venga indicata una D, il controllo viene effettuato tramite un dizionario dati.

Se il campo definito è presente sul dizionario, non va riportato nulla nello spazio destinato al nome della funzione di controllo, in quanto automaticamente viene attivato il controllo formale che sul dizionario è previsto per tale campo.

Viceversa, se il campo non è presente sul dizionario, nello spazio destinato alla funzione di controllo deve essere riportato il nome di un campo del dizionario. In questo caso, sul campo in oggetto viene effettuato il controllo formale previsto per il campo di dizionario indicato.

Evidentemente i tipi di controllo previsti sono gli stessi elencati sopra.Nelle pagine seguenti vengono mostrati alcuni esempi di combinazioni degli elementi descritti in precedenza, per la definizione dei campi e dei dati, per :

- pannello per maintenance
- pannello di visualizzazione a lista
- pannello di acquisizione dati
- pannello di maintenance a lista
- tabulato di stampa con gestione di livelli riepilogativi
- job stream di esecuzione batch con acquisizione dei parametri di attivazione

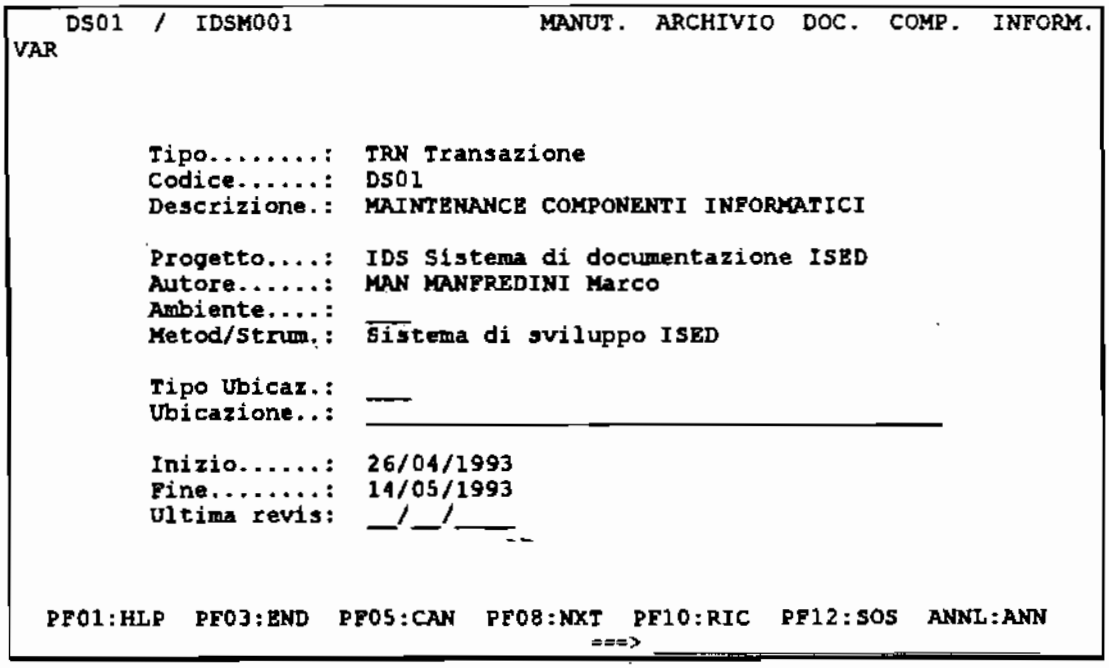
e vengono riportati, come ulteriore chiarimento i pannelli generati o le stampe prodotte.

```
&MAPFMT
/* -------------- PANNELLO PER MAINTENANCE ---------------- *
)DA
/*       ATTRIBUTI DEI CAMPI SU MAPPA
/*       +------ carattere da utilizzare
/*       | +---"U"nprot,"A"skip,"P"rot
/*       | |+--"N"ormal,"B"right,"D"ark
/*       | ||
/*       | ||
/*       V VV
/*       - --
         # AB
         # AN
         % DN
)DA
)PN
/*                    DISEGNO DELLA MAPPA
.CO &
.NZ 0005
       #Tipo.......: $___$_____________________________
       #Codice.....: $_____
       #Descrizione.: %_____________________________

       #Progetto...: %___$_____________________________
       #Autore.....: %___$_____________________________
       #Ambiente...: %___$_____________________________
       #Metod/Strum.: %___$_____________________________

       #TipoUbicaz.: %___$_____________________________
       #Ubicazione.: %_____________________________

       #Inizio.....: %_________
       #Fine.......: %_________
       #Ultimarevis: %_________
)PN
)DF
/*                  CARATTERISTICHE DEI CAMPI SU MAPPA
/*                        +---- "O"BBLIGATORIO
/*                        }+--- "S"QL,"W"ORKING,"R"ECORD
/*                        }}+-- "F"UNZIONE,"D"IZIONARIO,"M"OVE
/*                        }}}   Z = DESCRIZIONE
/*                        }}}+- "L"OGIC CHECK
/*                        }}}}
/*       NOME CAMPO RECORD  VVVV  FUNZIONE CONTROLLO
/*       ------------------ ----- ------------------
         CIN-TIPO            SF   TB03ID8001
         DES-TIPO             Z   AL35
         CIN-CODICE          SM   AL08
         CIN-DESCRIZIONE     OSF  AL40
         CIN-PROGETTO        OSF  TB03ID8002
         DES-PROGETTO         Z   AL35
         CIN-AUTORE          SF   TB03ID8003
         DES-AUTORE           Z   AL35
         CIN-AMBIENTE        SF   TB03ID8004
         DES-AMBIENTE         Z   AL35
         CIN-METOD-STRUM     SF   TB03ID8005
         DES-METOD-STRUM      Z   AL35
         CIN-TIPO-UBICAZ     SF   TB03ID8006
         DES-TIPO-UBICAZ      Z   AL35
         CIN-UBICAZIONE      SF   AL40
         CIN-INIZIO          SF   DI10
         CIN-FINE            SF   DI10
         CIN-ULTIMA-REVIS    SF   DI10
)DF
&ENDFMT
```


DS01 / IDSM001 MANUT. ARCHIVIO DOC. COMP. INFORM.
VAR
Tipo.....: TRN Transazione
Codice.....: DS01
Descrizione.: MAINTENANCE COMPONENTI INFORMATICI
Progetto....: IDS Sistema di documentazione ISED
Autore.....: MAN MANFREDINI Marco
Ambiente....:
Metod/Strum.: Sistema di sviluppo ISED
Tipo Ubicaz.:
Ubicazione..:
Inizio.....: 26/04/1993
Fine.....: 14/05/1993
Ultima revis: __/__/____
PF01:HLP PF03:END PF05:CAN PF08:NXT PF10:RIC PF12:SOS ANNL:ANN
===>

```
&LSTPMT
/* ------------------- PANNELLO DI VISUALIZZAZIONE A LISTA ----------
)PM
.HP 0001 HEAD
Progetto: ___ _____________________________
---------- ------------------------- ------------------------------------------------------------
Tipo Codice  Descrizione             Aut Amb M/S Ubicazione                                  Imp.
---------- ------------------------- ------------------------------------------------------------

.SP
.DT 0001 DETT
---- -------- ------------------------- --- --- --- --- ------------------------------------ --
.DT
)PM
)DF
/*                    CARATTERISTICHE DEI CAMPI SU TABULATO
/*                          +--- "K"EY
/*                          ]+-- "S"QL,"W"ORKING,"R"ECORD
/*                          ]]+- "F"UNZIONE,"D"IZIONARIO
/*                          ]]]  "M"OVE
/*                          ]]]
/*      NOME CAMPO INTERNO VVV   FUNZIONE CONTROLLO
/*      ------------------ ----- ------------------
        WSA-PROGETTO        SF   AL03
        WSA-DESCRIZIONE     SF   AL40
        CIN-TIPO            SF   AL03
        CIN-CODICE          SF   AL08
        CIN-DESCRIZIONE     SF   AL40
        CIN-AUTORE          SF   AL03
        CIN-AMBIENTE        SF   AL03
        CIN-METOD-STRUM     SF   AL03
        CIN-TIPO-UBICAZ     SF   AL03
        CIN-UBICAZIONE      SF   AL40
        WS-IND-IMPIEGO      WF   AL02
        CIN-TIPO           K M   AL03
        CIN-CODICE         K M   AL08
)DF
&ENDLST
```

```
DS03 / ISYOVIS          ELENCO DOCUMENTAZIONE COMP. INFORM.                VIS

  Progetto: IDS SISTEMA DI DOCUMENTAZIONE I.S.E.D.
  ---------- ------------------------- ------------------------------------
  Tipo Codice  Descrizione             Aut Amb M/S Ubicazione
  ---------- ------------------------- ------------------------------------

_ TRN DS01     MAINTENANCE COMPONENTI I MAN ___ ISY ___ _____________________
_ TRN DS02     MAINTENANCE STRUTTURE CO MAN VSE ISY ___ _____________________
_ TRN DS03     ELENCO COMPONENTI        MAN VSE ISY ___ _____________________
_ TRN DS04     STRUTTURA DI UN COMPONEN MAN VSE ISY ___ _____________________
_ TRN DS05     IMPIEGHI DI UN COMPONENT MAN VSE ISY ___ _____________________
_ TRN DS06     ESPLOSIONE SCALARE DI UN MAG VSE ISY ___ _____________________
_ TRN DS08     MAINTENANCE STRUTTURA DI MAN ___ ISY ___ _____________________

PF01:HLP  PF03:END  PF04:ATT  PF06:PRT  ANNL:ANN
                                        ===> _____________________________
```

```
&INPUT
/* ------------------- PANNELLO DI ACQUISIZIONE DATI ----------
/*
/*                    DEFINIZIONE DEI CAMPI DI ATTIVAZIONE
/*                              +---- "O"BBLIGATORIO
/*                              ]+--- "S"QL,"W"ORKING
/*                              ]]+-- "F"UNZIONE,"D"IZIONARIO
/*                              ]]]+- "L"OGIC CHECK
/*                              ]]]]
/*                              ]]]]
/*       NOME CAMPO             VVVV  FUNZIONE CONTROLLO IDENTIFICAZIONE
/*       ---------------------- ----- ------------------ ----------------
.RG 0010
       ATT-TIPO                 SF   TB03IDS001         TIPO COMPONENTE.:
.RG 0012
       ATT-CODICE               SF   AL08               COD. COMPONENTE.:
&ENDINP
```

```
DS01 / ISYMPAR          MANUT. ARCHIVIO DOC. COMP. INFORM.              SEL



TIPO COMPONENTE.:  ___

COD. COMPONENTE.:  _______




PF01:HLP  PF03:END  PF04:ATT
                                          ===>
```

```
/*            PANNELLO MAINTENANCE A LISTA
&MAPFMT
)DA
/*        ATTRIBUTI DEI CAMPI SU MAPPA
/*        +------ carattere da utilizzare
/*        | +---"U"nprot,"A"skip,"P"rot
/*        | |+--"N"ormal,"B"right,"D"ark
/*        | ||
/*        V VV
/*        - --
          & AB
          $ AN
          % UN
)DA
)PM
.CO  &
.RG  0003
&Padre:         &Tipo  &Codice    &Descrizione
                 $___  $________  $_____________________________________

&Figli: &Pos  &Tipo  &Codice    &Descrizione              &Espl  &Note
.MP 0015
        $___$  %___  %________  $_____________________%_    %________
.MP
)PM
)DF
/*                              CARATTERISTICHE DEI CAMPI SU MAPPA
/*                              +--- "O"bbligatorio,"W"orking St.con salvat.
/*                              |+-- "S"ql,"W"orking,"R"ecord
/*                              ||+- "F"unz.,"D"izion,"M"ove,"Z"descr.tabell/
/*                              |||+ "L"ogic check
/*                              ||||
/*                              ||||  FUNZIONE DI CONTROLLO
/*                              ||||  ++---------- controllo formale
/*                              ||||  ||++-------- lunghezza
/*                              ||||  ||||+------- decimali
/*                              ||||  |||||++++++- nome tabella
/*                              ||||  ||||||||||
/*        NOME CAMPO RECORD     VVVV  VVVVVVVVVV
/*        --------------------  ----- ----------
          SIN-TIPO-PADRE         SF   TB03IDS001
          SIN-CODICE-PADRE       SF   AL08
          CIN-DESCRIZIONE        SF   AL40
          SIN-POSIZIONE         OSF   NB02
          SIN-TIPO-FIGLIO       OSF   TB03IDS001
          SIN-CODICE-FIGLIO     OSFL  AL08
          WB-DESCRIZIONE         SF   AL40
          SIN-ESPLOSIONE         SF   TB01ISY006
          SIN-NOTE               SF   AL20
)DF
&ENDFMT
```

```
DS08 / IDSM008          TRANSAZIONE DI TEST                              VAR

Padre:          Tipo    Codice      Descrizione
                 PRG    IDS         SISTEMA DI DOCUMENTAZIONE I.S.E.D.

Figli:   Pos    Tipo    Codice      Descrizione              Espl    Note
_         10     DOC    IDSIST11    IDS - SISTEMA DI DOCUME  _
_         20     JOB    IDSCRTAB    CREAZIONE TABELLE SQL    _
_         30     JOB    IDSLDDIZ    CARICAMENTO DIZIONARIO   _
_         40     JOB    IDSLDTBD    CARICAMENTO TABELLE DI   _
_         50     JOB    IDSLDTRN    CARICAMENTO TRANSAZIONI  _
_         60     JOB    IDSLDHLP    CARICAMENTO HELP TRANSA  _
_         70     JOB    IDSLDERR    CARICAMENTO MESSAGGI ER  _
_         80     TRN    DS01        MAINTENANCE COMPONENTI   _
_         90     TRN    DS02        MAINTENANCE STRUTTURE C  _
_        100     TRN    DS03        ELENCO COMPONENTI        _
_        110     TRN    DS04        STRUTTURA DI UN COMPONE  _
_        120     TRN    DS05        IMPIEGNI DI UN COMPONEN  _
_        130     TRN    DS06        ESPLOSIONE SCALARE DI U  _
_        ___     ___    ________    ________________________ _
_        ___     ___    ________    ________________________ _

PF01:HLP  PF03:END  PF09:INS  PF10:RIC  PF12:SOS  ANNL:ANN
                                        ===>
```

```
DS08 / IDSM008          TRANSAZIONE DI TEST                              INS

Padre:          Tipo    Codice      Descrizione
                 PRG    IDS         SISTEMA DI DOCUMENTAZIONE I.S.E.D.

Figli:   Pos    Tipo    Codice      Descrizione              Espl    Note
_        ___     ___    ________    ________________________ _
_        ___     ___    ________    ________________________ _
_        ___     ___    ________    ________________________ _
_        ___     ___    ________    ________________________ _
_        ___     ___    ________    ________________________ _
_        ___     ___    ________    ________________________ _
_        ___     ___    ________    ________________________ _
_        ___     ___    ________    ________________________ _
_        ___     ___    ________    ________________________ _
_        ___     ___    ________    ________________________ _
_        ___     ___    ________    ________________________ _
_        ___     ___    ________    ________________________ _
_        ___     ___    ________    ________________________ _
_        ___     ___    ________    ________________________ _
_        ___     ___    ________    ________________________ _

PF01:HLP  PF03:END  PF08:NXT  PF09:INS  PF10:RIC  PF12:SOS  ANNL:ANN
ISY906I NESSUNA VARIAZIONE RICHIESTA      ===>
```

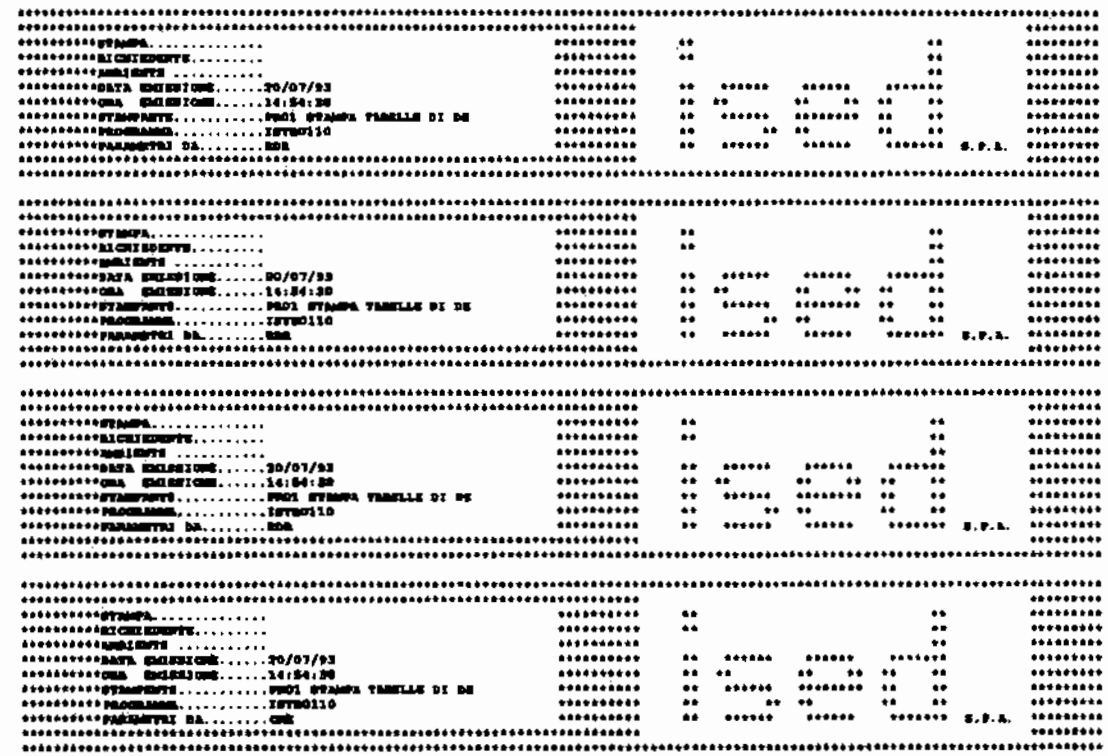
```
/*          TABULATO DI STAMPA CON GESTIONE DI LIVELLI RIEPILOGATIVI
&LSTFMT
)FM
.HP 0001 HEAD
____ / _____ _____________________________________ PAG. ___
                                ELABORAZIONE DEL _______

TABELLA _____ _____________________________________
------------ ------------------------------------ -------- --------
CODICE       DESCRIZIONE                          INSERIM. DISABIL.
------------ ------------------------------------ -------- --------
.HP
.DT 0001 DETT
_____________ ____________________________________ ________ ________
.DT
.BG 0001 EFG1

NUMERO ELEMENTI                                              _______
.EG
.EL 0001 EOFL

NUMERO TABELLE                                               _______
MEDIA ELEMENTI PER TABELLA                                   _______
.EL
)FM
)DF
/*                          CARATTERISTICHE DEI CAMPI SU TABULATO
/*                          +----"K"iave,"G"ruppo
/*                          |+---"S"ql,"W"orking,"R"ecord
/*                          ||+--"F"unzione,"D"izion.,"M"ove
/*                          |||+-"L"ogic check,"n"umero di livello
/*                          ||||
/*                          ||||  FUNZIONE DI CONTROLLO
/*                          ||||  ++---------- contr. formale
/*                          ||||  ||++-------- lunghezza
/*                          ||||  ||||+------- decimali
/*                          ||||  ||||+++++++- nome tabella
/*                          ||||  ||||||||||
/*       NOME CAMPO RECORD  VVVV  VVVVVVVVVV
/*       ------------------ ----- ----------
         ISY-CODICE-TRN       F   AL04
         ISY-CODICE-PGM       F   AL08
         ISY-DESCRIZ-TRN      F   AL35
         ISY-NUMERO-PAGINA    F   NB02
         ISY-DATA-CORRENTE    F   DI05
         TAB-SIGLA           RF   AL06
         WSA-TAB-DESCRIZ     WF   TX35
         TAB-CODICE          RF   AL13
         TAB-DESCRIZ         RF   TX35
         TAB-DATINS          RF   DI05
         TAB-DATDISA         RF   DI05
         WS-NR-ELEMENTI      WF   NB02
         WS-NR-TABELLE       WF   NB02
         WS-MEDIA-ELE-TAB    WF   NB02
         TAB-SIGLA           GRF1 AL06
)DF
&ENDLST
```

```
* $$JOB JNM=ISYB0110,CLASS=A,DISP=D,PRI=3,LDEST=(*,ISYLIB)
* $$LST CLASS=R,DISP=D,LST=X'00E'
* $$LST CLASS=R,DISP=D,LST=X'00F'
// JOB ISYB0110
/*
// UPSI 00000011
// ASSGN SYS040,X'00E'
// ASSGN SYS041,X'00F'
// ASSGN SYS042,IGN
// ASSGN SYS043,IGN
// ASSGN SYS044,IGN
// ASSGN SYS045,IGN
// DLBL SYS101,'ISY.TABELLE',,VSAM,CAT=ISYCAT
// EXEC ISYB0110,SIZE=AUTO
*$$PARBDR
DA TABELLA.......:IDS001
FINO A TABELLA...:IDS006
*$$PAREND
/*
/&
* $$EOJ
```



```
____ / ISYB0110 _____________________________________ PAG.   1
                                  ELABORAZIONE DEL 20/07/93

TABELLA IDS001 Tipo componente informativo
------------ ---------------------------------- -------- --------
CODICE       DESCRIZIONE                        INSERIM. DISABIL.
------------ ---------------------------------- -------- --------

ARE          Area funzionale                    05/05/93
CPY          Copy                               03/05/93
DOC          Documento                          03/05/93
JOB          Job Stream                         03/05/93
MAP          Mappa                              03/05/93
PGM          Programma                          03/05/93
PHA          Fase                               03/05/93
PRG          Progetto                           03/05/93
SCH          Schema ISY                         03/05/93
SIN          Sistema informativo                03/05/93
SPE          Specifica secondo ISY              03/05/93
SPG          Sottoprogetto                      03/05/93
TAB          Tabella SQL                        05/05/93
TBD          Tabella di decodifica              07/05/93
TRN          Transazione                        03/05/93

NUMERO ELEMENTI                                      15
```

```
____ / ISYB0110 _____________________________________ PAG.   2
                                  ELABORAZIONE DEL 20/07/93

TABELLA IDS002 Progetto
------------ ---------------------------------- -------- --------
CODICE       DESCRIZIONE                        INSERIM. DISABIL.
------------ ---------------------------------- -------- --------

IDS          Sistema di documentazione ISED     03/05/93
ISY          Sistema di sviluppo ISED           03/05/93
NPE          Sistema informativo area Personale 14/05/93
RPA          Patrimonio Immobiliare Comune Roma 03/05/93

NUMERO ELEMENTI                                       4
```

____ / ISYB0110 _____ PAG. 3
ELABORAZIONE DEL 20/07/93

TABELLA IDS003 Autore

CODICE	DESCRIZIONE	INSERIM.	DISABIL.
ANF	ANFORA Tobia	05/05/93	
CAF	CARLONI Fabrizio	10/05/93	
CAR	CARNIERI Riccardo	05/05/93	
FRA	FRATINI Marco	03/05/93	
GHI	GHIGI Gino	05/05/93	
MAC	MANCINI Maurizio	05/05/93	
MAG	MAGGI Onofrio Claudio	05/05/93	
MAN	MANFREDINI Marco	03/05/93	
MAO	MAGGINI Romeo	14/05/93	
MAS	MASTRANGELO Luigi	05/05/93	
MAZ	MAZZAGLIA Alejandro	05/05/93	
MIZ	MIZES Gustavo	05/05/93	
RAM	RAMOS Elisabeth Gharley	05/05/93	
REC	REA Carlo	10/05/93	

NUMERO ELEMENTI 14

____ / ISYB0110 _____ PAG. 4
ELABORAZIONE DEL 20/07/93

TABELLA IDS004 Ambiente operativo

CODICE	DESCRIZIONE	INSERIM.	DISABIL.
DOS	MS-DOS	03/05/93	
OS2	OS/2	03/05/93	
UNO	UNIX - ORACLE	05/05/93	
VMP	IBM - VM/SP - SQL/DS	05/05/93	
VSS	IBM - DOS/VSE - CICS - SQL/DS	05/05/93	
VSV	IBM - DOS/VSE - CICS - VSAM	05/05/93	
WIN	Microsoft Windows	03/05/93	

NUMERO ELEMENTI 7

____ / ISYB0110 _____ PAG. 5
ELABORAZIONE DEL 20/07/93

TABELLA IDS005 Metodologia o Strumento impiegato

CODICE	DESCRIZIONE	INSERIM.	DISABIL.
ADW	CASE Applicat.Development Workbench	03/05/93	
EXC	Microsoft Excel	03/05/93	
ISY	Sistema di sviluppo ISED	03/05/93	
WRD	Microsoft WORD	03/05/93	

NUMERO ELEMENTI 4

____ / ISYB0110 _____ PAG. 6
ELABORAZIONE DEL 20/07/93

TABELLA IDS006 Tipo di ubicazione

CODICE	DESCRIZIONE	INSERIM.	DISABIL.
CMS	Macchina virtuale CMS	03/05/93	
DSK	Disco magnetico	03/05/93	
FLO	Floppy o diskette	03/05/93	
LIB	Libreria VSE	03/05/93	
PER	Personal Computer	03/05/93	
TAP	Nastro magnetico	03/05/93	

NUMERO ELEMENTI 6

NUMERO TABELLE 6
MEDIA ELEMENTI PER TABELLA 8

Blocco di definizione di files non standard (in ambiente batch)

I due blocchi **&FILECON** e **&FILESEC** chiusi ciascuno da una **&END** permettono, se necessario, di definire files non previsti dagli standard adottati (ad es.: files su nastro, su disco, files VSAM, ecc.).

Il primo serve per definire le istruzioni Cobol SELECT (poste sotto la FILE-CONTROL della INPUT-OUTPUT SECTION) dei files necessari ed il secondo le corrispondenti File Definition FD (poste nella FILE SECTION della DATA DIVISION).

Blocco di definizione dei tasti funzionali/delle funzioni

Il blocco di righe riservato alla definizione dei tasti funzionali è individuato da una riga iniziale contenente la parola **&TASTI** e termina con una riga contenente la parola **&ENDTAS**.

Deve essere fornita una riga per ognuna delle funzioni particolari che si intendono gestire tramite tasti funzionali all'interno del programma. Non vanno definite in questa zona le funzioni standard relative alla tipologia di programma in quanto queste sono gestite all'interno dello scheletro.

Per ogni funzione viene definito:

- La sigla del tasto associato per default
- Un codice identificativo della funzione
- La modalità di attivazione dell'oggetto che realizza la funzione stessa

  R per routine attivata con PERFORM

  P per programma attivato con pseudo-LINK

  X per programma attivato con XCTL
- Il nome della routine o del programma a seconda che nel campo precedente sia stato fornito il carattere "R", o "P" o "X"
- La descrizione della funzione stessa.

Nel caso in cui debba essere attivato, tramite tasto funzionale applicativo, un programma con una XCTL o una pseudo-LINK, con passaggio di parametri che sono variabili di attivazione per il programma chiamato, deve essere richiesta l'attivazione di una routine. In questa routine deve essere preparata un'area di trasmissione contenente i valori dei parametri da comunicare (separati dal carattere !) ed inserite le seguenti istruzioni:

```
MOVE nome-programma-chiamato TO WS-PGM-XCTL
MOVE area-parametri TO CA-AREA-VARIABILE
MOVE lunghezza-area-parametri TO CA-AREA-VARINT
MOVE SW-XCTL-K TO WS-SW-FINE (per pseudo-LINK)
MOVE SW-XCTL-NORET-K TO WS-SW-FINE (per XCTL)
```

La zona dedicata alle funzioni è anche utilizzata nei programmi di visualizzazione a lista per selezionare la riga e, nello stesso tempo, fornire l'indicazione della funzione attivabile sulla riga selezionata della lista. In questo caso il significato dei campi di una riga di specifiche diviene il seguente:

- Campo destinato al tasto (non utilizzato)
- Il carattere che, digitato di lato alla riga, attiva su di essa la funzione
- La modalità di attivazione dell'oggetto che realizza la funzione stessa

  P per programma attivato con pseudo-LINK

  X per programma attivato con XCTL

  T per attivare direttamente un'altra transazione
- Il nome del programma che deve essere attivato per eseguire la funzione
- La descrizione della funzione stessa
- L'indicazione dei campi chiave che devono essere passati al programma che esegue la funzione.

Le chiavi vengono definite nella zona dedicata alla definizione della lista. Vengono qui richiamate con numeri progressivi adiacenti (da 1 a 9) che rappresentano l'ordine con cui sono state definite. È possibile indicare "ALL" se al programma devono essere fornite tutte le chiavi.

```
&TASTI
/*                        DEFINIZIONE DI TASTI FUNZIONALI
/*                  +---"R"outine,"P"rogramma,"X"ctl programma
/*                  |
/*       Fnn FUN V     PROGRAMMA/ROUTINE    DESCRIZIONE
/*       --- --- ----- -------------------- --------------------
         F10 ABI R     700-ABILITAZ-UTE     Abilitazione utente
&ENDTAS
```

Esempio di definizione tasti funzionali

```
&TASTI
/*                        DEFINIZIONE DELLE FUNZIONI
/*                  +---"P"rogramma,"X"ctl programma,"T"ran.(ret),"Y"tran.(noret
/*                  |
/*            FUN V     PROGRAMMA            DESCRIZIONE          KEY
/*         --- --- ----- -------------------- -------------------- --------
            S   X     ISYOPS00             LISTA COMPONENTI     ALL
&ENDTAS
```

Esempio di definizione funzioni per liste

ZONA DELLE SPECIFICHE DI ROUTINE

La zona riservata alle specifiche di routine contiene un insieme di blocchi di istruzioni o commenti che vengono inseriti in appositi punti all'interno dello scheletro.

Ogni blocco di istruzioni viene aperto da una scheda **&xxxxxx** che ne individua il nome (max 7 caratteri preceduti da &) e terminata da una scheda **&END**.

Il generatore non effettua nessuna operazione sulle istruzioni di questa zona limitandosi ad inserire nel punto previsto il blocco cui corrisponde un particolare nome.

Blocco di documentazione

All'interno del file specifiche dovrà essere inserita anche la documentazione del programma. Questa sarà suddivisa in blocchi ognuno dei quali riservato ad aspetti specifici. Ognuna delle righe dovrà iniziare con un asterisco a colonna 7 come è obbligatorio per i commenti.

I blocchi e le relative schede di apertura sono i seguenti:

&DOCGEN Documentazione generale

Contiene una descrizione dettagliata delle funzionalità del programma.

&DOCSTR Documentazione struttura

Contiene lo schema di struttura delle routine particolari definite nel file specifiche e del loro aggancio con la struttura base. Si ricordi che la struttura base, cioè quella associata alla tipologia del programma è descritta nel file scheletro.

&DOCINP Documentazione dell'input

Elenco di tutti gli archivi o tabelle accedute dal programma. Nel caso di programmi di utilità: elenco dei dati di ingresso al programma.

&DOCOUT Documentazione dell'output

Elenco di tutti gli archivi o tabelle aggiornate (in inserimento, modifica o cancellazione) dal programma. Nel caso di programmi di utilità: elenco dei dati in uscita dal programma.

&DOCMOD Documentazione delle modifiche

E' una tabella che contiene per ogni intervento di modifica effettuato sul programma, la data in cui è stato effettuato, la persona che lo ha effettuato ed una descrizione sintetica dell'intervento. La prima riga dovrà essere relativa al rilascio operativo del programma stesso.

&DOCCTL Documentazione controlli logici

E' una tabella che contiene il nome del campo su cui si è indicato di voler effettuare un controllo logico e la descrizione informale del controllo da fare, nonché il codice del messaggio di errore che va emesso in caso si verifichi la condizione di errore.

```
&DOCGEN
      ************************************************************
      *    DESCRIZIONE:                                          *
      *    descrizione generale del funzionamento del programma  *
      *                                                          *
      *                                                          *
&END
&DOCINP
      *    INPUT:                                                *
      *    descrizione dei file e dei dati di input al programma *
      *                                                          *
      *                                                          *
&END
&DOCOUT
      *    OUTPUT:                                               *
      *    descrizione dei file o dati di output                 *
      *                                                          *
      *                                                          *
&END
&DOCSTR
      *    STRUTTURA:                                            *
      *    descrizione sintetica dei passi di elaborazione       *
      *                                                          *
      *                                                          *
&END
&DOCMOD
      ************************************************************
      *   DATA    | NOMINATIVO    |  INTERVENTO                 *
      *-----------+---------------+-----------------------------*
      * GG/MM/AA  |  xxxxxxxxxxx  | RILASCIO OPERATIVO          *
      *           |               |                             *
      *           |               |                             *
      *           |               |                             *
&END
```

Esempio di documentazione

Completata la scrittura di tutte le parti del file specifiche fin qui descritte si può considerare portata a termine la fase iniziale del progetto realizzativo delle procedure. I documenti così prodotti vanno a costituire da una parte le specifiche di programmazione quale indispensabile strumento di riferimento per il gruppo di sviluppo, dall'altra la documentazione tecnica di progetto.

Proprio per questo motivo, onde facilitare l'operazione di scrittura della sezione documentativa delle specifiche, è possibile anziché utilizzare i blocchi **&DOCxxx**, che obbligano ad iniziare il testo da colonna sette con il carattere *, sostituirli con analoghi blocchi **&DESGEN**, **&DESINP**, **&DESOUT**, **&DESSTR**, **&DESCTL** (tutti terminati dalla label **&END**). *All'interno di questi blocchi le righe di testo possono iniziare da colonna uno ma non andare oltre colonna 61.*

Messaggi di errore

In questa zona, identificata dal nome **&WRKMSG**, dovranno essere inserite le istruzioni di definizione dei codici dei messaggi di errore emessi dalle routine. Il formato dei codici di errore deve essere quello previsto dallo standard.

```
&WRKMSG
/*    MESSAGGI DI ERRORE SPECIFICI DEL PROGRAMMA
/*       INSERIRE I MESSAGGI IN FORMATO:
/*     01  MSG-xxxxxxxx                PIC X(07) VALUE 'pppxxxt'.
       01  MSG-COMP-NOTF                PIC X(07) VALUE 'IsY039C'.
&END
```

Esempi di definizione dei codici per messaggi

Campi di lavoro delle routine

In questa zona, identificata dal nome **&WRKRTN**, dovranno essere inserite le istruzioni di definizione dei campi delle routine specifiche di programma. I campi relativi ad ogni routine dovranno essere separati gli uni dagli altri ed individuati da una apposita testata.

La definizione dei campi avverrà con normali istruzioni COBOL.

I campi di lavoro utilizzati in istruzioni SQL dovranno analogamente essere raggruppati ed preceduti dall'istruzione **&SQLRTN**. Questi campi saranno inseriti nella Declare Section nei programmi che utilizzano SQL/DS.

```
&WRKRTN
/*        CAMPI DI LAVORO SPECIFICI DEL PROGRAMMA

      ****************************************************************
      *     CAMPI DI LAVORO DELLA ROUTINE nnn-xxxxxxxxxx             *
      ****************************************************************

       01  CAMPO-1.
           02  S-CAMPO-1-1              PIC XX.
           02  .....

       01  ......

&END
&SQLRTN
/*        CAMPI DI LAVORO SQL SPECIFICI DEL PROGRAMMA

      ****************************************************************
      *     CAMPI DI LAVORO SQL DELLA ROUTINE nnn-xxxxxxxxxx         *
      ****************************************************************

       01  CAMPO-SQL-1                  PIC XX.
       01  ......

&END
```

Esempio di definizione di campi di routine

Preparazione tasti funzionali

In questa parte delle specifiche dovranno essere inserite le istruzioni di inibizione delle funzioni particolari, gestite dai tasti funzionali non standard, non attive in determinate condizioni.

Le istruzioni dovranno essere poste dopo la scheda identificativa **&PRETAS**, e saranno opportunamente agganciate in fase di generazione del programma COBOL alla routine che gestisce i tasti funzionali.

```
&PRETAS
           IF   CA-INSERIMENTO
                MOVE SPACES TO WS-FUNZ-COD-F09
                MOVE SPACES TO WS-FUNZ-COD-F10.
           IF   UTE-DATDISA = 0
                MOVE SPACES TO WS-FUNZ-COD-F10
           ELSE
                MOVE SPACES TO WS-FUNZ-COD-F09.
&END
```

Esempio di routine di inibizione tasti funzionali non standard.

Utilizzo dei dati del COPY di progetto

Gli *agganci* che seguono sono necessari qualora venga utilizzato il COPY di progetto che permette di passare dati tra programmi diversi di una stessa transazione.

Tali dati devono poter essere inizializzati, scritti da chi li deve passare, letti da chi li deve ricevere. Le istruzioni per le suddette operazioni devono poter essere opportunamente agganciate in punti particolari del programma generato. A questo scopo sono state previste le seguenti schede identificative:

&INIMAP per l'inizializzazione dei dati o per la modifica degli attributi dei campi in mappa

e per il *solo* schema MPNISY

&WRTQUE per la scrittura dei dati

&REDQUE per la lettura dei dati

Routines per le operazioni di I/O

Tutte le routine che seguono dovranno essere poste dopo la scheda identificativa **&ROUTINE**, avere una label come di seguito specificato ed essere poste in sequenza crescente in base alla parte numerica della label stessa.

600-I-O-READ-INIZ (Lettura iniziale)

Comprende tutte le operazioni che devono essere effettuate all'inizio del ciclo di lettura del/degli archivi interessati o comunque per la presentazione del primo pannello. Tra queste, a mò di esempio, la predisposizione delle testate e il posizionamento sul database.

610-I-O-NEXT (Lettura successiva)

In funzione dello *schema* utilizzato, comprende tutte le operazioni che devono essere effettuate ad ogni ciclo di lettura del/degli archivi interessati o di scorrimento in avanti del pannello.

Nel caso dello scorrimento degli archivi verrà attivata in loop fino al verificarsi della condizione di EOF che dovrà essere segnalata con l'indicatore WS-SW-FINE-STAMPA impostato al valore "1".

La preparazione delle righe di stampa non deve essere effettuata in quanto contenuta nel file scheletro.

620-I-O-READ-UPD (Lettura per update)

Comprende tutte le operazioni di lettura che devono essere effettuate preventivamente ad un aggiornamento degli archivi.

630-I-O-UPDATE (Aggiornamento)

Comprende tutte le istruzioni per l'aggiornamento del/dei record precedentemente letti per update. Deve contenere anche le istruzioni di consolidamento dell'operazione (ad es. COMMIT in caso di utilizzo del SQL).

640-I-O-UNLOCK (Operazione di ripristino)

Comprende le istruzioni per l'annullamento degli inserimenti, aggiornamenti, cancellazioni effettuate dopo l'ultima operazione di consolidamento (ad es. ROLLBACK in caso di utilizzo del SQL).

650-I-O-INSERT (Inserimento)

Comprende tutte le istruzioni necessarie per l'inserimento di uno o più nuovi record. Deve contenere anche le istruzioni di consolidamento dell'operazione (ad es. COMMIT in caso di utilizzo del SQL).

660-I-O-DELETE (Cancellazione)

Comprende tutte le istruzioni necessarie per la cancellazione del/dei record precedentemente letti per update. Deve contenere anche le istruzioni di consolidamento dell'operazione (ad es. COMMIT in caso di utilizzo del SQL).

Routines per eventuali controlli particolari dei dati in input

Tutte le routine che seguono dovranno essere poste dopo la scheda identificativa **&ROUTINE** ed avere una label come di seguito specificato.

800-CTRL-nome-del-campo (Controlli logici sui dati)

In questa parte delle specifiche dovranno essere inserite le eventuali routine che effettuano controlli speciali sui dati quali esistenza su un archivio, congruenza con altri dati, ecc.

In funzione dell'esito di tali controlli dovranno essere impostati gli opportuni messaggi di errore o segnalazione.

Routines di questo tipo possono essere utilizzate anche in altri casi.

Ad esempio, se in inserimento si vuole visualizzare una descrizione associata ad un campo di mappa, da prelevare in un archivio SQL, si utilizza un controllo logico sul campo di mappa. Nella routine associata si preleva dall'archivio la descrizione associata e si muove detta descrizione nel suo corrispondente campo di mappa con suffisso E.

Routines per inizializzazioni e riepiloghi nelle liste con rotture controllate

Seguono le routines che dovranno essere definite per il controllo delle rotture nelle liste e nei tabulati che le prevedono. Si ricorda che i livelli sono determinati dalla *parzialità* delle operazioni che si desidera effettuare su una lista ordinata e possono essere in numero massimo di 9.

Esse dovranno essere poste dopo la scheda identificativa **&ROUTINE**, avere una label come di seguito specificato ed essere poste in sequenza crescente in base alla parte numerica della label stessa.

nnn-INILIV-nome-gruppo (Operazioni iniziali di gruppo)

Questa routine dovrà contenere le istruzioni necessarie per l'inizializzazione eventuale di dati utilizzati nei riepiloghi e/o nella testata, associati al livello definito da *nome-gruppo* presente nel parametro .EG. Essa sarà attivata ad ogni rottura del livello cui appartiene.

nnn deve essere uguale a 900+(5*i), dove i e' il numero del livello di rottura (i > 1).

nnn-ENDLIV-nome-gruppo (Operazioni finali di gruppo)

Questa routine dovrà contenere le istruzioni necessarie per il riepilogo dei dati del livello definito da *nome-gruppo* presente nel parametro .EG. Essa sarà attivata ad ogni rottura del livello cui appartiene.

nnn deve essere uguale a 900+(5*i), dove i e' il numero del livello di rottura (i > 1).

900-ENDLIV-gruppo-fine-lista (Operazioni finali di lista)

Questa routine sarà attivata alla fine della lista e dovrà gestire le operazioni relative al riepilogo finale della lista stessa.

Ulteriori routines

Le eventuali ulteriori routines, non ricadenti in una delle tipologie precedenti, dovranno essere poste anch'esse dopo la scheda identificativa **&ROUTINE**, essere codificate in modo tale che la label di ognuna avrà un prefisso numerico superiore a 1000 ed essere poste in sequenza crescente in base alla parte numerica della label stessa.

Ognuna di esse dovrà essere preceduta da una breve descrizione. Le routine inserite in questa zona saranno richiamate da un istruzione PERFORM presente in uno qualsiasi dei blocchi descritti in precedenza.

Tutto il blocco contenente le routine sarà terminato dalla scheda **&END**.

Schemi scheletro disponibili

In questo capitolo sono riportate le descrizioni degli schemi scheletro disponibili, affinche possa essere valutata l'opportunità della loro utilizzazione.

L'utilizzazione di tali schemi deve servire non solo a rendere estremamente omogeneo il software realizzato, facilitandone quindi la gestione specialmente in fase di manutenzione, ma anche a predeterminare tipologie di elaborazione delle informazioni che sono particolarmente utili nella fase dell'analisi tecnica, allorché bisogna tradurre le specifiche funzionali di una procedura nelle singole transazioni che le realizzeranno.

Al fine di valutare e scegliere meglio lo schema da applicare nei casi che si presentano, si tengano anche presenti, oltre a quanto descritto negli schemi seguenti, tutte le altre funzionalità rese disponibili dagli standard utilizzati, la cui applicazione è garantita dalle istruzioni presenti negli schemi stessi.

Di tali funzionalità vengono ricordate di seguito le più significative in relazione all'architettura che la singola transazione deve realizzare:

- *ricircolo*, consistente nella possibilità (tasto funzionale RIC) di completare le operazioni in corso (es.: effettuare gli aggiornamenti) e di riattivare la stessa transazione senza uscire dalla vecchia ma specificando i nuovi dati nella zona predisposta allo scopo e identificata con

  ===>_____

- *sospensione*, consistente nella possibilità di interrompere (tasto funzionale SOS) in qualunque momento una transazione per attivarne un'altra e poi ritornare esattamente nelle condizioni di partenza
- *tasti funzionali applicativi* per il collegamento con altri programmi o routine particolari

Uno schema serve normalmente per realizzare un programma, ma più programmi provenienti anche da schemi differenti possono essere collegati tra loro per realizzare le funzionalità richieste alla transazione.

Tutti i pannelli, che gli schemi permettono di preparare, utilizzano gli standards del sistema di sviluppo per quanto riguarda la definizione delle prime righe (nome pannello, nome transazione, modalità di funzionamento, ...) e delle ultime (tasti funzionali, segnalazioni di errore, campo standard di input, ...), e quindi si limitano alla definizione delle righe intermedie (si ricorda che le righe disponibili per il corpo della mappa sono al massimo 20).

A prescindere dallo schema utilizzato, tutte le specifiche devono inserire i moduli di documentazione così come già specificato.

Maintenance con un solo programma (MNTISY)

Questo schema può essere utilizzato per i programmi associati a transazioni che prevedono di effettuare inserimenti, aggiornamenti, cancellazioni tramite un unico programma e un unico pannello.

Suddette operazioni ovviamente possono interessare più archivi nello stesso tempo, ma tutte le informazioni da gestire, cioè l'intero record logico trattato, devono essere simultaneamente presenti su un solo pannello.

I tasti funzionali sempre disponibili, se non esplicitamente esclusi, sono:

F1:HLP

per l'help in funzione della posizione del cursore

F2:SOS

viene sospesa la transazione che poi potrà essere riattivata a partire dallo stesso punto

F3:END

per terminare la transazione con l'esecuzione dell'operazione di maintenance richiesta

F5:CAN

per attivare la routine prevista nelle specifiche per le operazioni di cancellazione, da confermare successivamente con la pressione dello stesso tasto

F8:NXT

per eseguire le istruzioni previste nelle specifiche alla richiesta di scroll del pannello

F10:RIC

non viene attivata nessuna routine delle specifiche ma gestito il *ricircolo* con i dati presenti sull'ultima riga del pannello

ANNL:ANN

per annullare l'operazione in corso;

Possono essere utilizzati altri tasti funzionali *applicativi,* specifici della transazione realizzata. Nel caso che vengano attivati altri programmi, per evitare che venga mostrato il pannello che richiede i dati di attivazione propri del programma attivato, suddetti dati di attivazione possono essere predisposti, nell'area di comunicazione, dal programma chiamante, simulando quello che avviene quando gli stessi dati sono forniti insieme al codice della transazione (separati uno dall'altro dal carattere !).

Lo stesso programma, generato con uno schema di questo tipo a partire da specifiche complete di tutte le funzionalità di I/O, può, dipendendemente dal livello di abilitazione della transazione al singolo utente, presentare l'inibizione di talune di esse, in modo selettivo, fino alla sola possibilità di visualizzazione dei dati.

In altri termini, la stessa transazione, potrà essere utilizzata da alcuni utenti abilitati per effettuare inserimenti, aggiornamenti e cancellazioni, mentre altri potranno con essa effettuare solo alcune di queste operazioni.

I livelli di abilitazione gestiti sono:

- abilitazione completa
- solo visualizzazione
- solo modifica
- solo inserimento
- solo cancellazione
- tutte le funzionalità escluso l'inserimento
- tutte le funzionalità esclusa modifica
- tutte le funzionalità esclusa cancellazione

Le specifiche che utilizzano questo schema dovranno quindi contenere le seguenti parti:

- definizione della mappa completa di tutti gli attributi e i caratteri di continuazione (tra le parole separate dei campi fissi), che saranno stati precedentemente definiti;
- definizione di tutti i campi utilizzati, secondo le regole e il formato precedentemente descritti;
- definizione dei dati di attivazione della transazione;
- definizione dei tasti funzionali applicativi e dei tipi e nomi dei programmi ad essi associati;
- definizione dei messaggi di errore e dei campi di lavoro in formato COBOL;
- routine per letture iniziali degli archivi;
- istruzioni da eseguire allo scroll del pannello;
- routine con le letture da effettuare prima di un aggiornamento;
- routine con le istruzioni di aggiornamento degli archivi;
- routine con le istruzioni di inserimento del/dei record negli archivi;
- routine con le istruzioni di cancellazione del/dei record negli archivi;
- routine per il consolidamento delle operazioni effettuate;
- routine per l'annullamento delle operazioni effettuate;
- routines per gli eventuali controlli logici dei campi definiti in input.

Lo schema MNTISY, tra l'altro, definisce dinamicamente e in modo standard le seguenti parti di programma che saranno inserite nel programma COBOL generato:

- copy dell'unica mappa associata al programma e copy di sistema
- campi di attivazione della transazione
- campi della mappa in formato esterno e interno
- tutti gli attributi dei campi di attivazione e della mappa che servono alle routines del sistema per il controllo e la conversione dei dati relativi
- aree di lavoro e definizioni standard (aree di salvataggio, attributi, switches, messaggi, ...)
- istruzioni per la gestione della conversazionalità della transazione

- istruzioni per il controllo formale dei dati in input
- istruzioni per il collegamento con le routines di controllo logico definite nelle specifiche
- istruzioni per l'editing dei dati in output
- istruzioni per la gestione dei tasti funzionali standard prima definiti e per il collegamento con le routines definite nelle specifiche

Di seguito si riporta la schematizzazione essenziale di un file di specifiche finalizzato alla generazione di un programma con lo schema MNTISY.

```
&LANG        Linguaggio utilizzato
&PGMNAME     Nome del programma
&PGMDES      Descrizione del programma
&PROGET      Sigla del progetto
/*        -------------------- disegno della mappa  -------------------
&MAPFMT
)DA
/*                  DEFINIZIONE ATTRIBUTI DEI CAMPI DI MAPPA
/*
/*        +-----  CARATTERE DI DEFINIZIONE
/*        ]
/*        ] +---  "U"NPROT,"A"SKIP,"P"ROT
/*        ] ]+- "N"ORMAL,"B"RIGHT,"D"ARK
/*        ] ]]
/*        V VV
/*        - -----
          # AB
          $ AN
          @ AD
          % UN
          \ UD
)DA
)FM
.CO &
.RG nnnn



)FM
)DF
/*                              CARATTERISTICHE DEI CAMPI SU MAPPA
/*                              +--- "O"bbligatorio,"W"orking St.con salvat.
/*                              |+-- "S"ql,"W"orking,"R"ecord
/*                              ||+- "F"unz.,"D"izion,"M"ove,"Z"descr.tabell/
/*                              |||+ "L"ogic check
/*                              ||||
/*                              ||||  FUNZIONE DI CONTROLLO
/*                              ||||  ++---------- controllo formale
/*                              ||||  ||++-------- lunghezza
/*                              ||||  ||||+------- decimali
/*                              ||||  ||||+++++++- nome tabella
/*                              ||||  ||||||||||
/*        NOME CAMPO RECORD     VVVV  VVVVVVVVVV
/*        ------------------- ----- ------------------
)DF
&ENDFMT
&INPUT
/* ------------------- PANNELLO DI ACQUISIZIONE DATI ----------
/*
/*                    DEFINIZIONE DEI CAMPI DI ATTIVAZIONE
/*                              +---- "O"BBLIGATORIO
/*                              ]+--- "S"QL,"W"ORKING
/*                              ]]+-- "F"UNZIONE,"D"IZIONARIO
/*                              ]]]+- "L"OGIC CHECK
/*                              ]]]]
/*                              ]]]]
/*        NOME CAMPO            VVVV  FUNZIONE CONTROLLO IDENTIFICAZIONE
/*        ------------------- ----- ------------------ -----------------
.RG 0010
```

```
&ENDINP
&TASTI
/*                    DEFINIZIONE DELLE FUNZIONI
/*                +---"P"rogramma,"X"ctl programma,"T"ransazione
/*                |
/*           FUN V     PROGRAMMA           DESCRIZIONE           KEY
/*         --- --- ------ ------------------ -------------------- --------

&ENDTAS
&ENDVAR
&DOCGEN
      ***************************************************************
       *    DESCRIZIONE:                                              *
       *    descrizione generale del funzionamento del programma      *
       *                                                              *
       *                                                              *
&END
&DOCINP
       *    INPUT:                                                    *
       *    descrizione dei file e dei dati di input al programma     *
       *                                                              *
       *                                                              *
&END
&DOCOUT
       *    OUTPUT:                                                   *
       *    descrizione dei file o dati di output                     *
       *                                                              *
       *                                                              *
&END
&DOCSTR
       *    STRUTTURA:                                                *
       *    descrizione sintetica dei passi di elaborazione           *
       *                                                              *
       *                                                              *
&END
&DOCMOD
      ***************************************************************
       *   DATA    | NOMINATIVO   |  INTERVENTO                      *
       *-----------+--------------+----------------------------------*
       * GG/MM/AA |  xxxxxxxxxxx | RILASCIO OPERATIVO               *
       *          |              |                                  *
       *          |              |                                  *
       *          |              |                                  *
&END
&WRKRTN
       *  CAMPI DI LAVORO SPECIFICI DEL PROGRAMMA    *

/*       INSERIRE CAMPI IN FORMATO COBOL
&END
&SQLRTN
       *  CAMPI DI LAVORO SQL SPECIFICI DEL PROGRAMMA*

/*       INSERIRE CAMPI IN FORMATO COBOL
&END
&WRKMSG
       *  MESSAGGI DI ERRORE SPECIFICI DEL PROGRAMMA *

/*       INSERIRE I MESSAGGI IN FORMATO:
/*     01  MSG-xxxxxxxx                          PIC X(07)  VALUE 'PPPXXXT'.

&END
&PRETAS

&END
```

```
&ROUTINE
/*        INSERIRE LE ROUTINE SPECIFICHE.
/*        LA LABEL DI OGNI ROUTINE SARA' DEL TIPO:
/*       nnnn-label.
/*        CON nnnn crescente
/*        LA LABEL FINALE SARA':
/*       NNNN-LABEL-EX.
/*           EXIT.
/*
/*       ISTRUZIONI DI I-O DEI RECORD DA VISUALIZZARE/AGGIORNARE

       * - I-O-READ-INIZ                    (RICERCA RECORD RICHIESTO)      *

        600-I-O-READ-INIZ.

            MOVE CA-INSERIMENTO-K TO CA-FUNZIONE.

/* ISTRUZIONI PER LA LETTURA DEL RECORD

/* GESTIONE CONDIZIONE DI RECORD TROVATO
            MOVE CA-VARIAZIONE-K TO CA-FUNZIONE.

        600-I-O-READ-INIZ-EX.
            EXIT.


       * - I-O-NEXT                         (SCORRIMENTO ARCHIVIO)     *

        610-I-O-NEXT.

/* ISTRUZIONI PER LA LETTURA DEL RECORD SUCCESSIVO

        610-I-O-NEXT.
            EXIT.


       * - I-O-READ-UPD          (PREPARA RECORD PER AGGIORNAMENTO)  *

        620-I-O-READ-UPD.

            MOVE campi-mappa-I NELLE CHIAVI DI LETTURA

/* ISTRUZIONI PER LA RILETTURA DEL RECORD DA AGGIORNARE

        620-I-O-READ-UPD-EX.
            EXIT.


       * - I-O-UPDATE                          (AGGIORNAMENTO RECORD)       *

        630-I-O-UPDATE.
```

```
/* ISTRUZIONI PER AGGIORNAMENTO DEL RECORD

       630-I-O-UPDATE-EX.
           EXIT.


      * - I-O-UNLOCK                (RILASCIO RECORD NON AGGIORNATO)      *

       640-I-O-UNLOCK.


       640-I-O-UNLOCK-EX.
           EXIT.


      * - I-O-INSERT                          (INSERIMENTO RECORD)       *

       650-I-O-INSERT.

/* ISTRUZIONI PER INSERIMENTO DEL RECORD

       650-I-O-INSERT-EX.
           EXIT.


      * - I-O-DELETE                        (CANCELLAZIONE RECORD)       *

       660-I-O-DELETE.

           MOVE campi-mappa-I NELLE CHIAVI DI LETTURA

/* ISTRUZIONI PER LA CANCELLAZIONE DEL RECORD

       660-I-O-DELETE-EX.
           EXIT.
/*      CONTROLLI LOGICI EFFETTUATI SUI CAMPI IN INPUT
/*      LA LABEL DEVE ESSERE DEL TIPO 800-CTRL-(NOME-CAMPO)
       800-CTRL-nome-del-campo.

           MOVE campo-mappa-I NELLE CHIAVI DI LETTURA

/* ISTRUZIONI APPLICATIVE PER L'ESECUZIONE DEL CONTROLLO

           IF condizione-di errore
              MOVE MSG-xxxxxxxx TO WS-MSG-ERRORE

       800-CTRL-nome-del-campo-EX.
           EXIT.
&END
```

Maintenance conversazionale con un solo programma (MCOISY)

Questo schema è sostanzialmente uguale ad MNTISY, ma con l'importante unica differenza che esso è conversazionale invece di essere pseudo-conversazionale, cioè non rilascia il controllo dopo l'invio del pannello. Questa particolarità permette di introdurre in input testi in formato minuscolo (con la pseudo-conversazionalità tutti i dati vengono automaticamente sempre trasfomati in maiuscolo). Questo schema, quindi, deve essere utilizzato, invece di MNTISY, solamente se è indispensabile l'uso di testi contenenti lettere minuscole.

Visualizzazione a lista (VISISY)

Questo schema può essere utilizzato per i programmi che prevedono di visualizzare elenchi di dati da cui effettuare eventualmente selezioni.

A tali programmi è associata una mappa standard formata, per la parte *applicativa* sempre da 20 righe di 77 caratteri ciascuna più un campo fisso di selezione della riga. Per tale motivo, in questo caso non è necessario generare la mappa video.

Saranno presenti, normalmente, una testata e la parte di dettaglio che andrà ripetuta, ed i campi definiti non avranno necessità dell'attributo in quanto saranno costruiti dinamicamente sullo stesso campo di mappa (di 77 caratteri). In particolare la testata avrà sempre l'attributo *bright* e le righe di dettaglio l'attributo *normal*.

Una caratteristica importante di questo schema è costituita dalla possibilità di selezionare una riga dall'elenco visualizzato mediante uno o più caratteri che a loro volta selezionano uno o più programmi attivabili in relazione ai dati di quella riga, fornendo un'ampia flessibilità di architettura delle transazioni (ad es. navigazione tra le strutture e gli impieghi di dati organizzati gerarchicamente).

Questa funzionalità è resa possibile definendo nella parte riservata alla descrizione delle variabili, dopo i campi corrispondenti alle righe di testata e alla riga di dettaglio dell'elenco, i campi che devono essere trasmessi, come parametri di selezione, ai programmi che vengono attivati.

Come detto, questi campi prevedono di indicare come prima opzione il carattere K. Essi possono essere o meno campi visualizzati nell'elenco.

Nel blocco caratterizzato dalla label &TASTI, nel campo predisposto per l'indicazione di quali sono le chiavi trasmesse, esse possono essere identificate dalla posizione in cui sono ordinate.

Le specifiche che utilizzano questo schema dovranno contenere le seguenti parti:

- definizione della testata della visualizzazione (la testata del pannello è standard) che sarà presentata su ogni pannello durante lo scroll in avanti e all'indietro dell'elenco, comprendente campi fissi e campi variabili su una o più righe;
- definizione della parte di dettaglio, comprendente normalmente solo campi variabili che potrà essere costituita anch'essa di una o più righe;
- definizione di tutti i campi utilizzati, secondo le regole e il formato precedentemente descritti;
- definizione, oltre i campi di testata e di dettaglio, dei campi chiave da passare, relativamente alla riga selezionata, ai programmi attivati;
- definizione dei dati di attivazione della transazione;
- definizione dei caratteri di selezione ammissibili e dei tipi e nomi dei programmi ad essi associati;
- definizione dei messaggi di errore e dei campi di lavoro in formato COBOL;
- routine per letture iniziali degli archivi;
- routine per la preparazione di ogni successiva riga di dettaglio;

- routines per gli eventuali controlli logici dei campi definiti in input.

I tasti funzionali sempre disponibili sono:

F1:HLP

per l'help in funzione della posizione del cursore

F2:SOS

viene sospesa la transazione che poi potrà essere riattivata a partire dallo stesso punto

F3:END

per terminare la transazione;

F4:ATT

per attivare la funzione di riga richiesta tramite l'apposito carattere per l'occorrenza sulla lista

F6:PRT

per effettuare la stampa on-line, sulla stampantina associata preventivamente al record utente, dell'intero elenco visualizzato a terminale;

F7:BCK

per effettuare lo scroll indietro del pannello

F8:NXT

per effettuare lo scroll in avanti del pannello

ANNL:ANN

per annullare l'operazione in corso.

I tasti funzionali F7 e F8 compaiono solamente se esistono, rispettivamente, un pannello che precede e uno che segue quello attualmente visualizzato.

Lo schema VISISY, tra l'altro, definisce dinamicamente e in modo standard le seguenti parti di programma che saranno inserite nel programma COBOL generato:

- campi di attivazione della transazione
- campi della testata e della riga di dettaglio della mappa in formato esterno e interno
- tutti le informazioni dei campi di attivazione e della mappa che servono alle routines del sistema per il controllo e la conversione dei dati relativi
- aree di lavoro e definizioni standard (aree di salvataggio, attributi, switches, messaggi, ...)
- istruzioni per la gestione della conversazionalità della transazione
- istruzioni per il controllo formale dei dati di attivazione
- istruzioni per il collegamento con le routines di controllo logico definite nelle specifiche
- istruzioni per l'editing dei dati in output
- istruzioni per la gestione dei caratteri di selezione di una riga di dettaglio e l'attivazione dei programmi associati che sono stati definiti nelle specifiche

Di seguito si riporta la schematizzazione essenziale di un file di specifiche finalizzato alla generazione di un programma con lo schema VISISY.

```
&LANG          Linguaggio
&PGMNAME       Nome del programma
&PGMDES        Descrizione del programma
&PROGET        Sigla del progetto
&SQLDS
&DB2
&LSTFMT
/*       -------------------- DISEGNO DEL TABULATO --------------------
)FM
.HP 0001 HEAD

.HP
.DT 0001 DETT

.DT
)FM
)DF
/*                              CARATTERISTICHE DEI CAMPI SU TABULATO
/*                              +--- "G"ruppo
/*                              |+-- "S"ql,"W"orking,"R"ecord
/*                              ||+- "F"unz.,"D"izion,"M"ove,"Z"descr.tabell/
/*                              |||
/*                              |||
/*                              |||   FUNZIONE DI CONTROLLO
/*                              |||   ++---------- controllo formale
/*                              |||   ||++-------- lunghezza
/*                              |||   ||||+------- decimali
/*                              |||   |||||++++++- nome tabella
/*                              |||   ||||||||||
/*       NOME CAMPO RECORD      VVV   VVVVVVVVVV
/*       ------------------ ----- ------------------
)DF
&ENDLST
&INPUT
/* -------------------- PANNELLO DI ACQUISIZIONE DATI ----------
/*
/*                     DEFINIZIONE DEI CAMPI DI ATTIVAZIONE
/*                           +---- "O"BBLIGATORIO
/*                           ]+--- "S"QL,"W"ORKING
/*                           ]]+-- "F"UNZIONE,"D"IZIONARIO
/*                           ]]]+- "L"OGIC CHECK
/*                           ]]]]
/*                           ]]]]
/*       NOME CAMPO          VVVV  FUNZIONE CONTROLLO IDENTIFICAZIONE
/*       ------------------ ----- ------------------ ------------------
.RG 0010

&ENDINP
&TASTI
/*                     DEFINIZIONE DELLE FUNZIONI
/*              +---"P"rogramma,"X"ctl programma,"T"ransazione
/*              |
/*          FUN V     PROGRAMMA            DESCRIZIONE           KEY
/*       --- --- ----- ------------------ ------------------- --------

&ENDTAS
&ENDVAR
```

```
&DOCGEN
      ****************************************************************
      *    DESCRIZIONE:                                              *
      *    descrizione generale del funzionamento del programma      *
      *                                                              *
      *                                                              *
&END
&DOCINP
      *    INPUT:                                                    *
      *    descrizione dei file e dei dati di input al programma     *
      *                                                              *
      *                                                              *
&END
&DOCOUT
      *    OUTPUT:                                                   *
      *    descrizione dei file o dati di output                     *
      *                                                              *
      *                                                              *
&END
&DOCSTR
      *    STRUTTURA:                                                *
      *    descrizione sintetica dei passi di elaborazione           *
      *                                                              *
      *                                                              *
&END
&DOCMOD
      ****************************************************************
      *    DATA    | NOMINATIVO    |  INTERVENTO                    *
      *------------+---------------+--------------------------------*
      *  GG/MM/AA |  xxxxxxxxxxx  | RILASCIO OPERATIVO             *
      *           |               |                                *
      *           |               |                                *
      *           |               |                                *
&END
&WRKMSG
      *    MESSAGGI DI ERRORE SPECIFICI DEL PROGRAMMA      *

/*       INSERIRE I MESSAGGI IN FORMATO:
/*     01  MSG-xxxxxxxx                                    PIC X(07) VALUE
'PPPxxxt'.
&END
&WRKRTN
      *    CAMPI DI LAVORO SPECIFICI DEL PROGRAMMA  *

/*       INSERIRE CAMPI IN FORMATO COBOL
&END
&SQLRTN
      *  CAMPI DI LAVORO SQL SPECIFICI DEL PROGRAMMA*

/*       INSERIRE I CAMPI SQL NON DEFINITI NEL PANNELLO
&END
```

```
&ROUTINE
/*       INSERIRE LE ROUTINE SPECIFICHE.
/*       LA LABEL DI OGNI ROUTINE SARA' DEL TIPO:
/*     nnnn-label.
/*       CON nnnn >= 1000
/*       LA LABEL FINALE SARA':
/*     nnnn-label-EX.
/*          EXIT.
/*
/*
/*      ISTRUZIONI DI I-O DEI RECORD DA VISUALIZZARE
       * - I-O-READ-INIZ                          (LETTURE INIZIALI)      *

       600-I-O-READ-INIZ.

/* ISTRUZIONI DI LETTURA CAMPI DI TESTATA
/* ISTRUZIONI DI POSIZIONAMENTO PER IL SUCCESSIVO CICLO DI LETTURA

       600-I-O-READ-INIZ-EX.
            EXIT.


       * - I-O-NEXT                              (SCORRIMENTO ARCHIVIO)      *

       610-I-O-NEXT.

/* ISTRUZIONI DI CICLO LETTURA (SCORRIMENTO ARCHIVIO)
/* VERRANNO ATTIVATE IN LOOP FINO A FINE-STAMPA.

/* TEST DI FINE FILE
              IF fine-file
                   MOVE SW-FINE-STAMPA-K TO WS-SW-FINE-STAMPA

       610-I-O-NEXT-EX.
            EXIT.
/*      CONTROLLI LOGICI EFFETTUATI SUI CAMPI IN INPUT
/*      LA LABEL DEVE ESSERE DEL TIPO 800-CTRL-(NOME-CAMPO)
       800-CTRL-nome-del-campo.

            MOVE campo-mappa-I NELLE CHIAVI DI LETTURA

/* ISTRUZIONI APPLICATIVE PER L'ESECUZIONE DEL CONTROLLO

            IF condizione-di errore
                 MOVE MSG-xxxxxxxx TO WS-MSG-ERRORE

       800-CTRL-nome-del-campo-EX.
            EXIT.
&END
```

Inserimento multipannello (MPNISY)

Quando in una transazione di maintenance è indispensabile l'uso di più pannelli simultaneamente (perchè ad esempio devono essere inseriti e/o aggiornati *simultaneamente* una grande quantità di dati), possono essere collegati in cascata più programmi, con i relativi pannelli, se gli stessi sono stati generati con lo schema MNPISY.

Solamente l'ultimo programma della sequenza definita, può fare operazioni di I/O sugli archivi.

Tale schema si differenzia da uno schema MNTISY per i seguenti particolari:

- devono essere definite le seguenti variabili:

&TRAN	che definisce la transazione cui il programma *appartiene*
&TYPE	che definisce la sequenza dei programmi che fanno parte dello stesso gruppo, e può valere FIRST, SECOND,..., LAST
&NXTP GM	che definisce il programma che deve essere attivato successivamente in sequenza

- deve essere preventivamente definito e catalogato il COPY di progetto **xxxCWCA** (xxx è il codice del progetto) che contiene per ogni transazione del tipo *multipannello* la definizione dei campi che servono per il passaggio dei dati tra un programma e l'altro della catena;
- devono essere incluse le routines per:

  inizializzazione dei dati da passare (&INIMAP)

  scrittura dei dati da passare (&WRTMAP)

  lettura dei dati passati (&REDQUE)

I tasti funzionali sempre disponibili, se non esplicitamente esclusi, sono:

F1:HLP

per l'help in funzione della posizione del cursore

F2:SOS

viene sospesa la transazione che poi potrà essere riattivata a partire dallo stesso punto

F3:END

per terminare la transazione con l'esecuzione dell'operazione di maintenance richiesta

F8:NXT

per eseguire le istruzioni previste nelle specifiche alla richiesta di scroll del pannello

F10:RIC

non viene attivata nessuna routine delle specifiche ma gestito il *ricircolo* con i dati presenti sull'ultima riga del pannello

ANNL:ANN

per annullare l'operazione in corso;

Possono essere utilizzati altri tasti funzionali *applicativi*, specifici della transazione realizzata. In questo caso, per evitare che venga mostrato il pannello che richiede i dati di attivazione propri del programma attivato, suddetti dati di attivazione possono essere predisposti, nell'area di comunicazione, dal programma chiamante, simulando quello che avviene quando gli stessi dati sono forniti insieme al codice della transazione (separati uno dall'altro dal carattere !).

Le specifiche che utilizzano questo schema dovranno quindi contenere le seguenti parti:

- definizione della mappa completa di tutti gli attributi e i caratteri di continuazione (tra le parole separate dei campi fissi), che saranno stati precedentemente definiti;
- definizione di tutti i campi utilizzati, secondo le regole e il formato precedentemente descritti;
- definizione dei dati di attivazione della transazione;
- definizione dei tasti funzionali applicativi e dei tipi e nomi dei programmi ad essi associati;
- definizione dei messaggi di errore e dei campi di lavoro in formato COBOL;
- routine per letture iniziali degli archivi;
- istruzioni da eseguire allo scroll del pannello;
- routine con le letture da effettuare prima di un aggiornamento;
- routine con le istruzioni di aggiornamento degli archivi;
- routine con le istruzioni di inserimento del/dei record negli archivi;
- routine con le istruzioni di cancellazione del/dei record negli archivi;
- routine per il consolidamento delle operazioni effettuate;
- routine per l'annullamento delle operazioni effettuate;
- routines per gli eventuali controlli logici dei campi definiti in input.

Lo schema MPNISY, tra l'altro, definisce dinamicamente e in modo standard le seguenti parti di programma che saranno inserite nel programma COBOL generato:

- copy dell'unica mappa associata al programma e copy di sistema
- campi di attivazione della transazione
- campi della mappa in formato esterno e interno

- tutti gli attributi dei campi di attivazione e della mappa che servono alle routines del sistema per il controllo e la conversione dei dati relativi
- aree di lavoro e definizioni standard (aree di salvataggio, attributi, switches, messaggi, ...)
- istruzioni per la gestione della conversazionalità della transazione
- istruzioni per il controllo formale dei dati in input
- istruzioni per il collegamento con le routines di controllo logico definite nelle specifiche
- istruzioni per l'editing dei dati in output
- istruzioni per la gestione dei tasti funzionali standard prima definiti e per il collegamento con le routines definite nelle specifiche

Maintenance a lista immediato (MLSISY)

Questo schema può essere utilizzato per i programmi associati a transazioni che prevedono di effettuare aggiornamenti e cancellazioni su una lista di elementi, costituiti ciascuno da dati provenienti da uno o più records, e inserimenti *in massa* degli stessi elementi.

Tali operazioni sono *immediate*, nel senso che le modifiche richieste vengono effettuate prima di un eventuale scroll ad un pannello successivo.

Una caratteristica di questo schema, analoga a quella dello schema di visualizzazione a lista, è costituita dalla possibilità di selezionare una riga dall'elenco mediante un carattere che a sua volta determina l'attivazione di un programma o di una routine presente nelle specifiche, in relazione ai dati di quella riga. In particolare con il carattere C, che non deve mai essere utilizzato nelle specifiche viene attivata la routine 660-I-O-DELETE per la cancellazione, senza conferma, dell'elemento selezionato.

Da notare che tanto le funzioni di riga, quanto i tasti funzionali applicativi, vengono definiti nel blocco contraddistinto dalla label &TASTI; la differenza consiste nel fatto che, per definire una funzione di riga, nel campo riservato al nome del tasto funzione, va riportata la stringa **FRG**.

Con questo schema l'*inserimento* di uno o più elementi nuovi viene richiesto con la pressione di un tasto funzionale (PF9). In questo caso viene presentato un pannello con il numero di elementi previsti completamente vuoti e predisposti a ricevere i dati. Da notare che i dati che erano stati definiti *protetti* e *obbligatori* (ad es. chiavi non modificabili) vengono sprotetti per permettere l'introduzione dei dati. E' opportuno a tale proposito che suddetti campi siano terminati da un attributo ASKIP (vedi esempio su descrizione specifiche).

I tasti funzionali sempre disponibili, se non esplicitamente esclusi, ed oltre quelli applicativi eventualmente specificati, sono:

F1:HLP

per l'help in funzione della posizione del cursore

F2:SOS

viene sospesa la transazione che poi potrà essere riattivata a partire dallo stesso punto

F3:END

per terminare la transazione con l'esecuzione dell'operazione di maintenance richiesta

F4:ATT

per attivare la funzione di riga richiesta tramite l'apposito carattere per l'occorrenza sulla lista

F8:NXT

per eseguire le istruzioni previste nelle specifiche alla richiesta di scroll del pannello

F10:RIC

non viene attivata nessuna routine delle specifiche ma gestito il *ricircolo* con i dati presenti sull'ultima riga del pannello

ANNL:ANN

per annullare l'operazione in corso;

Le specifiche che utilizzano questo schema dovranno contenere le seguenti parti:

- definizione della mappa completa di tutti gli attributi e i caratteri di continuazione (tra le parole separate dei campi fissi), che saranno stati precedentemente definiti; tale mappa dovrà contenere la definizione dell'elemento di dettaglio con la sua molteplicità e con tutti i campi contenuti accompagnati dallo specifico attributo (vedi l'istruzione speciale .MP nnnn);
- definizione di tutti i campi utilizzati, secondo le regole e il formato precedentemente descritti;
- definizione dei dati di attivazione della transazione;
- definizione dei tasti funzionali applicativi e dei tipi e nomi dei programmi ad essi associati;
- definizione dei messaggi di errore e dei campi di lavoro in formato COBOL;
- routine per letture iniziali degli archivi;
- istruzioni da eseguire allo scroll del pannello;
- routine con le letture da effettuare prima di un aggiornamento;
- routine con le istruzioni di aggiornamento degli archivi;
- routine con le istruzioni di inserimento del/dei record negli archivi;
- routine con le istruzioni di cancellazione del/dei record negli archivi;
- routine per il consolidamento delle operazioni effettuate;
- routine per l'annullamento delle operazioni effettuate;
- routines per gli eventuali controlli logici dei campi definiti in input.

Lo schema MLSISY, tra l'altro, definisce dinamicamente e in modo standard le seguenti parti di programma che saranno inserite nel programma COBOL generato:

- copy dell'unica mappa associata al programma e copy di sistema
- campi di attivazione della transazione
- campi della mappa in formato esterno e interno
- tutti gli attributi dei campi di attivazione e della mappa che servono alle routines del sistema per il controllo e la conversione dei dati relativi
- aree di lavoro e definizioni standard (aree di salvataggio, attributi, switches, messaggi, ...)
- istruzioni per la gestione della conversazionalità della transazione
- istruzioni per il controllo formale dei dati in input

- istruzioni per il collegamento con le routines di controllo logico definite nelle specifiche
- istruzioni per l'editing dei dati in output
- istruzioni per la gestione dei tasti funzionali standard prima definiti e per il collegamento con le routines definite nelle specifiche

Di seguito si riporta la schematizzazione essenziale di un file di specifiche finalizzato alla generazione di un programma con lo schema MLSISY.

```
&LANG          Linguaggio
&PGMNAME       Nome del programma
&PGMDES        Descrizione del programma
&PROGET        Sigla del progetto
&ONLYVIS
&SQLDS
&DB2
&MAPFMT
/* ---------------- DISEGNO E DEFINIZIONE DELLA MAPPA ---------------- *
)DA
/*                DEFINIZIONE ATTRIBUTI DEI CAMPI DI MAPPA
/*
/*         +-----  CARATTERE DI DEFINIZIONE
/*         ]
/*         ] +---  "U"NPROT,"A"SKIP,"P"ROT
/*         ] ]+-- "N"ORMAL,"B"RIGHT,"D"ARK
/*         ] ]]
/*         V VV
/*         - -----
           # AB
           $ AN
           @ AD
           % UN
           \ UD
)DA
)FM
/*                      DISEGNO DELLA MAPPA
.CO &
.RG nnnn
/* RIGHE DI TESTATA
       ...
       ...
.MP 0015
/* RIGHE DI DETTAGLIO
       ...
       ...
.MP
)FM
)DF
/*                              CARATTERISTICHE DEI CAMPI SU MAPPA
/*                              +--- "O"bbligatorio,"W"orking St.con salvat.
/*                              |+-- "S"ql,"W"orking,"R"ecord
/*                              ||+- "F"unz.,"D"izion,"M"ove,"Z"descr.tabell/
/*                              |||+ "L"ogic check
/*                              ||||
/*                             T||||  FUNZIONE DI CONTROLLO
/*                              ||||  ++---------- controllo formale
/*                              ||||  ||++-------- lunghezza
/*                              ||||  ||||+------ decimali
/*                              ||||  ||||++++++- nome tabella
/*                              ||||  ||||||||||
/*         NOME CAMPO RECORD    VVVV  VVVVVVVVVV
/*         ------------------ ----- ----------------
       ...
       ...
)DF
&ENDFMT
```

```
&INPUT
/* ------------------- PANNELLO DI ACQUISIZIONE DATI ----------
/*
/*                    DEFINIZIONE DEI CAMPI DI ATTIVAZIONE
/*                          +---- "O"BBLIGATORIO
/*                          ]+--- "$"QL,"W"ORKING
/*                          ]]+-- "F"UNZIONE,"D"IZIONARIO
/*                          ]]]+- "L"OGIC CHECK
/*                          ]]]]
/*                          ]]]]
/*       NOME CAMPO         VVVV  FUNZIONE CONTROLLO IDENTIFICAZIONE
/*       ------------------ ----- ---------------- ------------------
.RG NNNN
      ...
      ...
&ENDINP
&TASTI
/*                    DEFINIZIONE DELLE FUNZIONI
/*              +---"P"rogramma,"X"ctl programma,"T"ransazione
/*              |
/*          FUN V     PROGRAMMA            DESCRIZIONE          KEY
/*       --- --- ----- ------------------ ------------------- --------
         FRG C   R     660-I-O-DELETE       CAN.NR POSIZIONE
&ENDTAS
&ENDVAR
&DOCGEN
      ***************************************************************
       *    DESCRIZIONE:                                             *
       *    descrizione generale del funzionamento del programma     *
       *                                                             *
       *                                                             *
&END
&DOCINP
       *    INPUT:                                                   *
       *    descrizione dei file e dei dati di input al programma    *
       *                                                             *
       *                                                             *
&END
&DOCOUT
       *    OUTPUT:                                                  *
       *    descrizione dei file o dati di output                    *
       *                                                             *
       *                                                             *
&END
&DOCSTR
       *    &TRUTTURA:                                               *
       *    descrizione sintetica dei passi di elaborazione          *
       *                                                             *
       *                                                             *
&END
&DOCMOD
      ***************************************************************
       *   DATA    | NOMINATIVO   |  INTERVENTO                      *
       *-----------+--------------+----------------------------------*
       * GG/MM/AA  |  xxxxxxxxxxx | RILASCIO OPERATIVO               *
       *           |              |                                  *
       *           |              |                                  *
       *           |              |                                  *
&END

&MESSMSG
       *  MESSAGGI DI ERRORE SPECIFICI DEL PROGRAMMA

/*       INSERIRE I MESSAGGI IN FORMATO:
```

```
/*     01  MSG-XXXXXXXX                                  PIC X(07) VALUE 'PPPXXXT'.
&END
&WRKRTN
      *  CAMPI DI LAVORO SPECIFICI DEL PROGRAMMA
/*       INSERIRE CAMPI IN FORMATO COBOL
&END
&SQLRTN
      *  CAMPI DI LAVORO SQL SPECIFICI DEL PROGRAMMA
/*       INSERIRE I CAMPI SQL NON DEFINITI NEL PANNELLO
&END
&PRETAS
/*      ISTRUZIONI FORMATO COBOL PER LE ROUTINE ASSOCIATE AI TASTI
/*      FUNZIONE NON STANDARD
&END
&ROUTINE
/*       INSERIRE LE ROUTINE APPLICATIVE.
/*       LA  LABEL DI OGNI ROUTINE SARA' DEL TIPO:
/*     MMMM-LABEL.
/*       CON MMMM IN ORDINE CRESCENTE
/*       LA LABEL FINALE SARA':
/*     MMMM-LABEL-EX.
/*           EXIT.
/* ------------------------------------------------------------------ *
/*      ISTRUZIONI DI I-O DEI RECORD DA VISUALIZZARE/AGGIORNARE
/* ------------------------------------------------------------------ *
      *
      * - I-O-READ-INIZ (IMPOSTAZIONE TESTATA E POSIZION.INIZIALE)
      *
     600-I-O-READ-INIZ.
           MOVE SPACE                TO WS-SW-FINEFILE.
/* PRIMA LETTURA
           IF W-CAMPI-RIGA-I = LOW-VALUE
              MOVE campo-di-attivazione-I NELLE CHIAVI DI LETTURA
/* LETTURE SUCCESSIVE
           ELSE
              MOVE campo-chiave-I NELLE CHIAVI DI LETTURA
/* CONDIZIONE PER LA PAGINAZIONE AVANTI E INDIETRO
              MOVE W-campo-di-mappa-IK NELLE CHIAVI DI LETTURA
/* ISTRUZIONI PER LA LETTURA INIZIALE CON CONDIZIONE DI AVANZAMENTO
      ...
      ...
     600-I-O-READ-INIZ-EX.
           EXIT.
      *
      * - I-O-NEXT                              (SCORRIMENTO ARCHIVIO)
      *
     610-I-O-NEXT.
/* ISTRUZIONI PER LE LETTURE DI SCORRIMENTO
      ...
      ...
/* TEST DI FINEFILE
           IF fine-file
              MOVE SW-FINE-STAMPA-K TO WS-SW-FINE-STAMPA
     610-I-O-NEXT-EX.
           EXIT.
      *
      * - I-O-READ-UPD          (PREPARA RECORD PER AGGIORNAMENTO) *
      *
     620-I-O-READ-UPD.
           MOVE campi-chiave-I NELLE CHIAVI DI LETTURA
           MOVE W-campo-di-mappa-I (RICORR.DELLA RIGA) NELLE CHIAVI DI LETT.
```

```
/* ISTRUZIONI PER LA RILETTURA DEL RECORD
      ...
      ...
       620-I-O-READ-UPD-EX.
           EXIT.
      *
      * - I-O-UPDATE                          (AGGIORNAMENTO RECORD)
      *
       630-I-O-UPDATE.
/* ISTRUZIONI PER L'AGGIORNAMENTO DEL RECORD
      ...
      ...
       630-I-O-UPDATE-EX.
           EXIT.
      *
      * - I-O-UNLOCK                 (RILASCIO RECORD NON AGGIORNATO)
      *
       640-I-O-UNLOCK.
           PERFORM 350-ANNULLA-AGGIORNAMENTI THRU
                   350-ANNULLA-AGGIORNAMENTI-EX.
       640-I-O-UNLOCK-EX.
           EXIT.
      *
      * - I-O-INSERT                            (INSERIMENTO RECORD)
      *
       650-I-O-INSERT.
/* ISTRUZIONI PER L'INSERIMENTO DEL RECORD
      ...
      ...
       650-I-O-INSERT-EX.
            EXIT.
      *
      * - I-O-DELETE                           (CANCELLAZIONE RECORD)
      *
       660-I-O-DELETE.
           MOVE campi-chiave-I NELLE CHIAVI DI LETTURA
           MOVE W-campo-di-mappa-I (RICORR.DELLA RIGA) NELLE CHIAVI DI LETT.
/* ISTRUZIONI PER LA CANCELLAZIONE DEL RECORD
      ...
      ...
       660-I-O-DELETE-EX.
           EXIT.
      *
      * - I-O-CONSOLIDA              (CONSOLIDAMENTO AGGIORNAMENTI)
      *
       670-I-O-CONSOLIDA.
           PERFORM 340-CONSOLIDA-AGGIORNAMENTI THRU
                   340-CONSOLIDA-AGGIORNAMENTI-EX.
       670-I-O-CONSOLIDA-EX.
           EXIT.
/* ------------------------------------------------------------ *
/*      CONTROLLI LOGICI EFFETTUATI SUI CAMPI IN INPUT
/*      LA LABEL DEVE ESSERE DEL TIPO 800-CTRL-(NOME-CAMPO)
/* ------------------------------------------------------------ *
       800-CTRL-nome-campo.
           MOVE W-nome-campo-I NELLE CHIAVI DI LETTURA
/* ISTRUZIONI PER LA VERIFICA DI ERRORI LOGICI
           IF condizione-di-errore
              MOVE WS-ATTR-ERROR TO W-nome-campo-A
              MOVE MSG-xxxxxxxx TO WS-MSG-ERRORE.
       800-CTRL-nome-campo-EX.
           EXIT.
&END
```

Stampa tabulati on-line (PRTISY)

Questo schema permette la preparazione di programmi di stampa on-line.

Lo stesso programma generato viene eseguito, una prima volta per l'acquisizione dei parametri di attivazione (con una transazione associata al terminale dell'utente) ed una seconda volta per l'esecuzione della stampa (con una transazione associata alla stampante il cui codice è stato preventivamente impostato dall'utente).

Con questo schema possono essere utilizzati fino a 9 livelli (gruppi) di dati di riepilogo parziale ed un livello di riepilogo finale così come mostrato nell'esempio riportato nel paragrafo *Descrizione dei campi.* In tal caso dovranno essere previste le routine 900-... come in precedenza specificato.

Si osservi che, in una stampa che prevede una testata e una lista di elementi, può essere forzato il *salto pagina* (ad. esempio alla rottura di uno o più campi di testata) forzando zero nel campo WS-CTR-RIGHE-RESTANTI.

Le specifiche che utilizzano questo schema dovranno contenere le seguenti parti:

- definizione della testata della stampa che sarà presentata su ogni pagina, comprendente campi fissi e campi variabili su una o più righe;
- definizione della parte di dettaglio, comprendente normalmente solo campi variabili che potrà essere costituita anch'essa di una o più righe;
- definizione di tutti i campi utilizzati, secondo le regole e il formato precedentemente descritti;
- definizione dei dati di attivazione della transazione di stampa;
- definizione dei messaggi di errore e dei campi di lavoro in formato COBOL;
- routine per letture iniziali degli archivi;
- routine per la preparazione di ogni successiva riga di dettaglio;
- routines per gli eventuali controlli logici dei campi definiti in input.

I tasti funzionali presenti sul pannello di acquisizione dei parametri di attivazione sono:

F1:HLP

per l'help in funzione della posizione del cursore

F3:END

per terminare la transazione;

ANNL:ANN

per annullare l'operazione in corso.

Lo schema PRTISY, tra l'altro, definisce dinamicamente e in modo standard le seguenti parti di programma che saranno inserite nel programma COBOL generato:

- campi di attivazione della transazione
- campi della testata e della riga di dettaglio della stampa in formato esterno e interno

- tutti le informazioni dei campi di attivazione e della stampa che servono alle routines del sistema per il controllo e la conversione dei dati relativi
- aree di lavoro e definizioni standard (aree di salvataggio, attributi, switches, messaggi, ...)
- istruzioni per la gestione della conversazionalità della parte di transazione relativa all'acquisizione dei dati
- istruzioni per il controllo formale dei dati di attivazione
- istruzioni per il collegamento con le routines di controllo logico definite nelle specifiche
- istruzioni per l'editing dei dati in output

Di seguito si riporta la schematizzazione essenziale di un file di specifiche finalizzato alla generazione di un programma con lo schema PRTISY.

```
&LANG         Linguaggio
&PGMNAME      Nome programma
&PGMDES       Descrizione del programma
&PROGET       Sigla del progetto
&PAGELNT      nn
&TESTATA      nn
&LSTFMT
/*      ------------------- DISEGNO DEL TABULATO --------------------
)FM
.HP 0001 HEAD
/* RIGHE DI TESTATA
      ...
      ...
.HP
.DT 0001 DETT
/* RIGHE DI DETTAGLIO
      ...
      ...
.DT
)FM
)DF
/*                              CARATTERISTICHE DEI CAMPI SU TABULATO
/*                              +--- "G"ruppo
/*                              |+-- "S"ql,"W"orking,"R"ecord
/*                              ||+- "F"unz.,"D"izion,"M"ove,"Z"descr.tabell/
/*                              |||
/*                              |||
/*                              |||   FUNZIONE DI CONTROLLO
/*                              |||   ++---------- controllo formale
/*                              |||   ||++------- lunghezza
/*                              |||   ||||+------ decimali
/*                              |||   ||||++++++- nome tabella
/*                              |||   ||||||||||
/*        NOME CAMPO RECORD     VVV   VVVVVVVVVV
/*        ----------------- ----- ----------------
      ...
      ...
)DF
&ENDLST
&INPUT
/* ------------------- PANNELLO DI ACQUISIZIONE DATI ----------
/*
/*                      DEFINIZIONE DEI CAMPI DI ATTIVAZIONE
/*                            +---- "O"BBLIGATORIO
/*                            ]+--- "S"QL,"W"ORKING
/*                            ]]+-- "F"UNZIONE,"D"IZIONARIO
/*                            ]]]+- "L"OGIC CHECK
/*                            ]]]]
/*                            ]]]]
/*        NOME CAMPO          VVVV  FUNZIONE CONTROLLO IDENTIFICAZIONE
/*        ----------------- ----- ---------------- ----------------
.RG NNNN
      ...
      ...
&ENDINP
&ENDVAR
```

```
&DOCGEN
      ****************************************************************
      *    DESCRIZIONE:                                              *
      *    descrizione generale del funzionamento del programma      *
      *                                                              *
      *                                                              *
&END
&DOCINP
      *    INPUT:                                                    *
      *    descrizione dei file e dei dati di input al programma     *
      *                                                              *
      *                                                              *
&END
&DOCOUT
      *    OUTPUT:                                                   *
      *    descrizione dei file o dati di output                     *
      *                                                              *
      *                                                              *
&END
&DOCSTR
      *    STRUTTURA:                                                *
      *    descrizione sintetica dei passi di elaborazione           *
      *                                                              *
      *                                                              *
&END
&DOCMOD
      ****************************************************************
      *    DATA    | NOMINATIVO    |  INTERVENTO                    *
      *------------+---------------+--------------------------------*
      *  GG/MM/AA  |  xxxxxxxxxxxx | RILASCIO OPERATIVO             *
      *            |               |                                *
      *            |               |                                *
      *            |               |                                *
&END

&WRKMSG
      *    MESSAGGI DI ERRORE SPECIFICI DEL PROGRAMMA

/*       INSERIRE I MESSAGGI IN FORMATO:
/*     01  MSG-xxxxxxxx                          PIC X(07) VALUE 'PPPXXXT'.
&END
&WRKRTN
      *    CAMPI DI LAVORO SPECIFICI DEL PROGRAMMA
/*       INSERIRE CAMPI IN FORMATO COBOL
&END
&SQLRTN
      *  CAMPI DI LAVORO SQL SPECIFICI DEL PROGRAMMA*
/*       INSERIRE I CAMPI SQL NON DEFINITI NEL PANNELLO
&END
&ROUTINE
/*       INSERIRE LE ROUTINE SPECIFICHE.
/*       LA  LABEL DI OGNI ROUTINE SARA' DEL TIPO:
/*     NNNN-LABEL.
/*       CON NNNN IN ORDINE CRESCENTE
/*       LA LABEL FINALE SARA':
/*     NNNN-LABEL-EX.
/*          EXIT.
/* ------------------------------------------------------------------ *
/*      ISTRUZIONI DI I-O DEI RECORD DA VISUALIZZARE
/* ------------------------------------------------------------------ *
      *
      * - I-O-READ-INIZ                          (LETTURE INIZIALI)
      *
```

```
/*      ISTRUZIONI DI INIZIALIZZAZIONE STAMPA (LETTURE INIZIALI)
       600-I-O-READ-INIZ.
            MOVE SPACE              TO WS-SW-FINE-STAMPA.
            MOVE campi-di-attivazione-I NELLE CHIAVI DI LETTURA
/* ISTRUZIONI DI LETTURA CAMPI DI TESTATA
       ...
       ...
/* ISTRUZIONI DI POSIZIONAMENTO PER LE LETTURE SUCCESSIVE
       ...
       ...
       600-I-O-READ-INIZ-EX.
            EXIT.
       *
       * - I-O-NEXT                          (SCORRIMENTO ARCHIVIO)
       *
/*      ISTRUZIONI DI CICLO STAMPA (SCORRIMENTO ARCHIVIO)
/*      VERRANNO ATTIVATE IN LOOP FINO A FINE-STAMPA.
       610-I-O-NEXT.
/* ISTRUZIONI PER LETTURE DI SCORRIMENTO FILE
       ...
       ...
            IF record-da-non-stampare
                GO TO 610-I-O-NEXT.
/* TEST DI FINEFILE
            IF fine-file
                MOVE SW-FINE-STAMPA-K TO WS-SW-FINE-STAMPA.
       610-I-O-NEXT-EX.
            EXIT.
/* ------------------------------------------------------------------ *
/*    CONTROLLI LOGICI EFFETTUATI SUI CAMPI IN INPUT
/*    LA LABEL DEVE ESSERE DEL TIPO 800-CTRL-(NOME-CAMPO)
/* ------------------------------------------------------------------ *
       800-CTRL-nome-campo.
            MOVE nome-campo-I NELLE CHIAVI DI LETTURA
/* ISTRUZIONI PER L'ESECUZIONE DEL CONTROLLO
            IF condizione-di-errore
                MOVE MSG-xxxxxxxx TO WS-MSG-ERRORE.
       800-CTRL-nome-campo-EX.
            EXIT.
&END
```

Stampa tabulati batch (PRBISY)

Questo schema permette la preparazione di programmi di stampa batch. Da notare che praticamente qualunque tipo di elaborazione batch può essere realizzata mediante l'uso di questo schema il quale si occupa sostanzialmente solo della parte relativa all'acquisizione e controllo dei parametri di attivazione dell'elaborazione e all'editing in output dei dati di stampa. Si rimanda al documento *Specifiche di struttura dei programmi in ambiente batch* per una guida nella definizione dell'architettura dell'algoritmo di elaborazione.

Se le istruzioni di I/O non sono diverse tra l'ambiente on-line e quello batch, come nel caso del SQL, le specifiche preparate per produrre una stampa on-line (PRTISY) possono essere utilizzate anche per produrre la analoga stampa batch (PRBISY). Da osservare che anche specifiche preparate per una transazione di visualizzazione on-line (VISISY), con le quali peraltro può essere automaticamente prodotta la corrispondente stampa on-line, possono essere utilizzate per produrre anche l'analogo tabulato a lista batch (PRBISY). In sostanza con una specifica scritta una sola volta possono essere generati tre diversi programmi COBOL (una visualizzazione a lista al terminale, una stampa on-line della stessa lista e l'identica stampa batch).

Analogamente al pannello per l'acquisizione on-line dei dati, una analoga struttura, come mostrato nell'esempio riportato in precedenza, deve essere inserita nella Job Stream per l'esecuzione del job batch.

Anche con questo schema possono essere utilizzati fino a 9 livelli (gruppi) di dati di riepilogo parziale ed un livello di riepilogo finale così come mostrato nell'esempio riportato nel paragrafo *Descrizione dei campi*. In tal caso dovranno essere previste le routine 900-... come in precedenza specificato.

Si osservi che, in una stampa che prevede una testata e una lista di elementi, può essere forzato il *salto pagina* (ad. esempio alla rottura di uno o più campi di testata) forzando zero nel campo WS-CTR-RIGHE-RESTANTI.

Le specifiche che utilizzano questo schema dovranno contenere le seguenti parti:

- definizione della testata della stampa che sarà presentata su ogni pagina, comprendente campi fissi e campi variabili su una o più righe;
- definizione della parte di dettaglio, comprendente normalmente solo campi variabili che potrà essere costituita anch'essa di una o più righe;
- definizione di tutti i campi utilizzati, secondo le regole e il formato precedentemente descritti;
- definizione dei dati di attivazione della elaborazione batch;
- definizione dei messaggi di errore e dei campi di lavoro in formato COBOL;
- routine per letture iniziali degli archivi;
- routine per la preparazione di ogni successiva riga di dettaglio;
- routines per gli eventuali controlli logici dei campi definiti in input.

Lo schema PRBISY, tra l'altro, definisce dinamicamente e in modo standard le seguenti parti di programma che saranno inserite nel programma COBOL generato:

- campi di attivazione dell'elaborazione batch
- campi della testata e della riga di dettaglio della stampa in formato esterno e interno
- tutti le informazioni dei campi di attivazione e della stampa che servono alle routines del sistema per il controllo e la conversione dei dati relativi
- aree di lavoro e definizioni standard (aree di salvataggio, attributi, switches, messaggi, ...)
- istruzioni per il controllo formale dei dati di attivazione
- istruzioni per il collegamento con le routines di controllo logico definite nelle specifiche
- istruzioni per l'editing dei dati in output

Generazione della mappa CICS / BMS (MAPPA)

Da tutte le specifiche che generano programmi che utilizzano mappe (non standard come per le specifiche VISISY), con questo schema può essere ottenuto il file che definisce tutti i campi di mappa con relative posizioni e attributi. Tale file è costituito da un programma assembler nel caso della generazione delle mappe CICS / BMS. Si ricorda a tale proposito che questa mappa deve essere compilata, una volta per generare e catalogare il COPY che è inserito nel programma che utilizza la mappa e una volta per generare il programma la definizione fisica della mappa.

Campi di Common area e working storage standard

Tutti gli schemi disponibili fanno uso di alcuni copy. Tra questi ve ne sono due contenenti la definizione di variabili da utilizzarsi anche nella stesura dei files di specifiche:

- ISYCCA
- ISYCWSS

La prima rappresenta la common area, a sua volta suddivisa in una parte di sistema ed in una parte applicativa.

La seconda è la porzione standard di working storage.

I campi della common area sono caratterizzati dal prefisso "ca-". Quelli della working storage standard dal prefisso "ws-".

Di seguito sono descritti i più significativi per la programmazione.

ca-funzione

Questo campo contiene la sigla dell'operazione di I/O che verrà eseguita nel programma. In particolare, le specifiche del tipo MNTISY, devono inizializzare questa variabile tramite la costante CA-INSERIMENTO-K, nel blocco di lettura iniziale.

In questo modo, se leggendo il record richiesto, questo non viene trovato, la funzione sarà già predisposta alla modalità di inserimento; viceversa, in caso di esistenza del record stesso, la variabile deve essere rivalorizzata tramite la costante CA-VARIAZIONE-K, in modo da predisporre la funzione in modalità aggiornamento.

ca-ute-sigla

In questo campo viene memorizzato il codice identificativo dell'utente che ha effettuato il sign-on. Esso può pertanto essere utilizzato per scrivere sui records che lo prevedono l'autore della modifica.

ca-ute-unorg

In questo campo è contenuto l'identificativo dell'unità organizzativa di eventuale appartenenza dell'utente. Esso (o una sua opportuna sottodefinizione) può essere utilizzato ai fini di controlli interni di abilitazione per l'accesso ai dati.

ca-curdata-iso

In questo campo è resa sempre disponibile la data corrente in formato standard ISO, vale a dire AAAA-MM-GG.

Questo formato è uno di quelli accettati in input per i campi data di archivi SQL, e può pertanto essere utilizzato, fra le altre cose, per scrivere la data di inserimento o di aggiornamento sui records di archivi che lo prevedono.

ca-curtime-iso

Analogamente al campo precedente, esso rende disponobile l'ora di inizio della transazione nel formato standard ISO, vale a dire HH.MM.SS.

ws-msg-errore

E' questo il campo di working storage che gli schemi utilizzano per operare il test sull'esito dei controlli logici. Pertanto, quando nella routine apposita viene riscontrato un errore logico, deve essere mosso l'opportuno messaggio (definito nel blocco contraddistinto dalla label &WRKMSG) in questo campo. Ad esempio:

```
IF condizione-di-errore
    MOVE MSG-xxxxxxxx TO WS-MSG-ERRORE
```

ws-sw-fine-stampa

Si tratta dello switch di controllo dei cicli di lettura presenti in specifiche del tipo VISISY, MLSISY, PRTISY, PRBISY. Giacchè ad ogni lettura viene controllato il valore di questo switch, per chiudere il ciclo esso deve essere modificato tramite la costante SW-FINE-STAMPA-K.

ws-sw-fine

Questo campo è lo switch che determina la modalità di uscita da un programma. Esso deve essere utilizzato qualora debba essere attivato un sottoprogramma, con passaggio di parametri di attivazione e si intenda effettuare ciò con una XCTL o con una pseudo-LINK.

ws-attr-xx e ***ws-map-cursor***

Per ciascuna combinazione di attributi di mappa è definito in working storage un campo il cui nome è dato da *ws-attr* più un suffisso di due caratteri che sono le iniziali corrispondenti alla combinazione di attributi in oggetto (ad esempio, la combinazione di attributi unprot e bright sarà nel campo *ws-attr-ub*). Il campo *ws-map-cursor* contiene la posizione corrente del cursore sulla mappa.

Pertanto, se al verificarsi di certe condizioni, come è il caso in cui si è riscontrato un errore logico, devono essere modificati gli attributi di un campo e posizionato opportunamente il cursore, occorre fornire le istruzioni seguenti:

```
MOVE WS-ATTR-xx TO nome-campo-ATT
MOVE nome-campo-POS TO WS-MAP-CURSOR
```

ws-ctr-righe-restanti

Questo campo è il contatore di controllo delle righe da stampare su una pagina secondo la lunghezza fornita nella variabile di sistema &PAGELNT. Nei programmi di stampa infatti, esso è inizializzato a tale valore e decrementato ad ogni riga stampata. Il salto pagina viene effettuato ogniqualvolta questo contatore diviene zero.

Pertanto, se per necessità contingenti deve essere forzato un salto pagina, deve essere nel punto opportuno azzerato questo contatore.

Nel manuale ISYDDAT sono riportati i tracciati interi delle copy ISYCCA ed ISYCWSS.

Messaggi di errore della generazione

Quando il programma generatore incontra durante la fase di generazione delle situazioni non previste emette un messaggio di errore sul terminale per segnalare l'evento. Il messaggio viene anche duplicato su un file degli errori per permetterne una più comoda consultazione.

Il formato di un messaggio è il seguente:

```
fffpppppp vvvvvvvvvvvvvvvv ** nnnt mmm...mmm
```

dove:

fff	è una stringa identificativa del file su cui il programma generatore ha trovato l'errore, e cioè. • SCH per il file scheletro • SPE per il file specifiche
pppppp	è il progressivo del record sul file su cui è stato trovato l'errore
vv..vv	può essere presente in alcuni casi per una migliore specificazione dell'errore. Ad esempio in caso di variabile inesistente, è il nome della variabile.
nnn	è il codice identificativo dell'errore
t	è un codice di un carattere che indica il grado di gravità dell'errore, e cioè: • I per messaggi informativi • W per errori senza influenza sulla generazione • C per errori corretti automaticamente • E per errori bloccanti Solo gli errori dell'ultimo tipo portano ad una generazione sicuramente errata.
mm..mm	è il messaggio di errore

Al termine della fase di generazione viene inoltre inviato al terminale un messaggio con l'indicazione del numero complessivo di errori riscontrati suddiviso per tipologia.

I messaggi di errore sono i seguenti:

001W SCHEDA &ENDVAR ASSENTE

Nel file specifiche non è presente la scheda &ENDVAR che dovrebbe dividere la zona di definizione delle variabili da quella riservata alle routine. Non si ha comunque nessuna influenza nella generazione.

002C SCHEDA &ENDFMT NON TROVATA. ASSUNTA PRESENTE

Mentre il programma generatore stava esaminando la zona del file specifiche riservata alla descrizione del pannello, ha trovato una scheda iniziante con il carattere & senza aver prima trovato la scheda di fine zona &ENDFMT. La generazione prosegue assumendo la presenza della scheda &ENDFMT immediatamente prima dell'altra.

003C SCHEDA &ENDLST NON TROVATA. ASSUNTA PRESENTE

Mentre il programma generatore stava esaminando la zona del file specifiche riservata alla descrizione del tabulato, ha trovato una scheda iniziante con il carattere & senza aver prima trovato la scheda di fine zona &ENDLST. La generazione prosegue assumendo la presenza della scheda &ENDLST immediatamente prima dell'altra.

004C SCHEDA &ENDINP NON TROVATA. ASSUNTA PRESENTE

Mentre il programma generatore stava esaminando la zona del file specifiche riservata alla descrizione dei parametri di attivazione, ha trovato una scheda iniziante con il carattere & senza aver prima trovato la scheda di fine zona &ENDINP. La generazione prosegue assumendo la presenza della scheda &ENDINP immediatamente prima dell'altra.

005C SCHEDA &ENDTAS NON TROVATA. ASSUNTA PRESENTE

Mentre il programma generatore stava esaminando la zona del file specifiche riservata alla descrizione dei tasti funzionali, ha trovato una scheda iniziante con il carattere & senza aver prima trovato la scheda di fine zona &ENDTAS. La generazione prosegue assumendo la presenza della scheda &ENDTAS immediatamente prima dell'altra.

006E NUMERO RIGA IN .RG ERRATO O ZERO. SCARTATO.

All'interno della zona riservata alla descrizione del pannello o dei parametri di attivazioni è stata una scheda .RG di impostazione del numero di riga riportante il valore zero o non numerica. La scheda è stata scartata.

007C VARIABILE DUPLICATA. SOSTITUITA

All'interno della zona di definizione delle variabili nel file specifiche è stata trovata una seconda volta una variabile già definita. Il secondo valore viene sostituito al primo.

008E MOLTIPLICATORE IN .DT ERRATO O ZERO

Nel blocco di definizione del tabulato la scheda .DT ha un valore di moltiplicatore uguale a zero o non numerico.

009E ERRORE IN DEFINIZIONE STAMPA

Nel blocco di definizione di un pannello o di una stampa è stata trovata una riga non compresa nè nel sottoblocco)FM nè in quello)DF. La riga viene scartata.

010E SUPERATO N.MAX VARIABILI

Sono state definite un numero di variabili superiore al limite previsto (100).

011E SUPERATO N.MAX CAMPI MAPPA

Il numero di campi presente su una mappa è superiore al limite massimo previsto (100).

012E SUPERATO N.MAX CAMPI INPUT

Il numero di parametri in ingresso è superiore al massimo previsto (50).

013E SUPERATO N.MAX TASTI FUNZ.

Il numero di tasti funzionali definiti ha superato il limete massimo previsto (50).

014E SUPERATO N.MAX LINEE SALVATE IN &REPEAT

Il numero di righe relative ad una funzione &REPEAT, cioè quelle comprese fra la scheda &REPEAT e la &ENDREP è superiore al limite massimo previsto (100).

015E SUPERATO N.MAX CAMPI STAMPA

Il numero di campi presente nella definizione del tabulato è superiore al limite massimo previsto (100).

015E SUPERATO N.MAX CAMPI STAMPA

Il numero di campi presente nella definizione del tabulato è superiore al limite massimo previsto (100).

016E SUPERATO N.MAX ATTRIBUTI

Il numero di attributi presente nella definizione del tabulato è superiore al limite massimo previsto (30).

017E MANCA ATTRIBUTO SU UNO O PIU' CAMPI

Uno o più attributi non sono stati definiti.

018W CARATTERE DI CONTINUAZIONE NON SPECIFICATO

Manca la specifica del carattere di continuazione.

019E SPAZIO DOPO CAMPO DIGITABILE, NON RISPETTATO

Manca lo spazio per poter inserire lo Stopper field nelle mappe BMS.

023C FUNZIONE CONTROLLO SCONOSCIUTA. ASSUNTO "AL"

Nella definizione di un campo tramite una funzione è stato indicato un tipo di funzione non previsto. Il generatore assume il tipo alfanumerico (ALnn).

024E FUNZIONE CONTROLLO ERRATA. LUNGHEZZA NON NUMERICA

In una funzione di controllo è stata rilevata la presenza del campo lunghezza non numerico. Il campo viene scartato.

050E VARIABILE NON TROVATA

In una istruzione è stato trovato il riferimento ad una variabile non definita. La traduzione della variabile non viene effettuata.

052E LABEL DI GOTO NON TROVATA

In una istruzione &GOTO la Label non esiste o non è successiva all'istruzione stessa. L'istruzione ha l'effetto di portare alla fine del file.

053E FUNZIONE NON CONSENTITA IN &IF

Nell'istruzione &IF è stata utilizzata un'istruzione non prevista. L'azione conseguente non viene effettuata.

054E FORMATO FUNZIONE ERRATO. MANCA "="

In una istruzione &SET o &COMP dopo il secondo elemento non è un simbolo "=". L'istruzione non viene eseguita.

055E IL PRIMO ELEMENTO DEVE ESSERE UNA VARIABILE

In una istruzione &SET o &COMP il primo elemento non è una variabile. L'istruzione non viene eseguita.

056E OPERANDO NON NUMERICO IN &COMP

In una istruzione &COMP uno dei due operandi non è una costante numerica o è una variabile il cui valore non è numerico. L'istruzione non viene eseguita.

057E OPERAZIONE NON PREVISTA

In una istruzione &COMP il carattere che indica l'operazione da effettuare non è uno di quelli previsti (+,-,+,/).

058E ERRORE DI LUNGHEZZA IN NOME VARIABILI

In una istruzione è stata trovata una variabile il cui nome ha una lunghezza superiore agli 8 caratteri. La traduzione non viene effettuata.

059E SOVRAPPOSIZIONE VARIABILE E TESTO

Nella sostituzione di una variabile con il suo valore corrente lo spazio intercorrente fino all'inizio della successiva parola non è sufficiente a contenere il valore. Il valore viene troncato.

060E LUNGHEZZA ISTRUZIONE SUPERIORE MAX

Nella sostituzione di una variabile con il suo valore viene superata la dimensione massima dell'istruzione (colonna 72). L'istruzione viene troncata.

061C SUPERATO NUMERO MASSIMO DI CHIAVI DI ATTIVAZIONE

Possono essere definiti un numero massimo di 8 campi come chiavi.

062C SUPERATO NUMERO MASSIMO DI FUNZIONI ATTIVABILI

Possono essere definiti un numero massimo di 8 programmi attivabili.

063E CHIAVE O GRUPPO IN CORRISP. CON CAMPO STAMPA

064E MOLTIPLICATORE ERRATO O ZERO IN &LOOP

065E MOLTIPLICATORE IN .EG ERRATO O ZERO

066E MOLTIPLICATORE IN .EL ERRATO O ZERO

067C SUPERATO NUMERO MASSIMO GRUPPI DEFINIBILI

Possono essere definiti un numero massimo di 9 gruppi.

068C NUMERO DI LIVELLI ERRATO

Possono essere definiti un numero massimo di 9 livelli.

069E CAMPO DI LAVORO IN CORRISP. CON CAMPO MAPPA

070E RIGA NON CONTENENTE VARIABILI IN FILE DEFAULT

Procedure operative in ambiente VM/CMS

Il presente capitolo è finalizzato a fornire le informazioni basilari riguardo le operazioni di:

- generazione programmi sorgente
- generazione mappe
- compilazione
- testing

con riferimento all'ambiente di sviluppo CMS.

Mentre le fasi di compilazione e testing rappresentano i tradizionali passi finali del processo di sviluppo del software, le operazioni che abbiamo denominato *generazione* sono legate all'utilizzo degli strumenti dell'ambiente di sviluppo.

Di seguito sono fornite le informazioni sull'ambiente operativo di sviluppo e sulle operazioni da compiere dopo il completamento della scrittura del file di Specifihe.

Configurazione macchine ed ambiente di sviluppo

L'ambiente di sviluppo standard prevede la presenza di:

- n macchine CMS di sviluppo denominate pppVMnn
- 1 macchina CMS di libreria denominta pppLIB

dove: ppp è la sigla del progetto mentre nn è un progressivo.

Ciascuna delle suddette macchine ha una configurazione standard in termini di minidischi come quella riportata nello specchietto seguente:

VDEV	M	STAT	
191	A	R/W	Files di lavoro personali
110	B	R/O	Scheletri ISY e procedure
102	E	R/W	Simbolici delle mappe BMS
103	F	R/W	Simbolici dei programmi e delle specifiche
106	J	R/W	Procedure JCL di riferimento
107	K	R/W	Copy e macro
203	L	R/O	Simbolici dei progr. e delle specif. in libreria
206	N	R/O	Job stream di progetto

207	O	R/O	Copy e macro di progetto
210	T	R/O	Default di progetto

La presenza dei componenti informatici negli opportuni minidischi CMS è vincolante in quanto la procedura di generazione ricerca ivi i suoi oggetti.

Una volta che la versione di un programma (specifica ISY) è stata opportunamente testata e consolidata essa va copiata nell'apposito disco della macchina di libreria pppLIB e cancellata dalla macchina di sviluppo.

Qualora fosse necessario intervenire nuovamente sulla specifica, questa va prelevata dalla macchina di libreria e copiata sulla macchina di sviluppo. Su questa copia vengono effettuate le modifiche ed i nuovi test al termine dei quali va ripetuto lo scarico su pppLIB.

Generazione programmi e mappe

In ambiente CMS la procedura per la generazione dei simbolici dei programmi e delle mappe BMS è la medesima. Il suo nome é **GENPGMN**.

Come detto sopra i files delle specifiche devono risiedere sul minidisco F. Essi devono avere il nome codificato come segue:

FNAME = pppsxxxv

FTYPE = SPEC

dove :

ppp = sigla del progetto

s = codice dello scheletro da utilizzare in generazione

xxx = identificativo del programma

v = versione del file specifiche

Questi quattro valori sono contenuti nel file specifiche stesso, rispettivamente nelle variabili:

&PROGET

&PROGRAM

&SCHEMA

&VERSPE

La procedura di generazione può essere attivata nei seguenti formati:

1) **GENPGMN pppsxxxv**

legge il file pppsxxxv SPEC F e produce pppsxxxv COBOL F in quanto non essendo specificati altri parametri assume come schema quello corrispondente alla sigla contenuta in s

2) **GENPGMN pppsxxxv y**

legge sempre il file pppsxxxv SPEC F e produce pppyxxxv COBOL F dove y è una qualsiasi delle sigle dei possibili schemi (questa possibilità è data perchè da uno stesso file di specifiche possono essere generati programmi sorgente diversi a partire da differenti schemi, anche se la specifica ha un nome che fa comunque riferimento allo schema target principale)

3) **GENPGMN pppsxxxv M**

per produrre automaticamente la mappa pppMxxx ASSEMBLE E

4) **GENPGMN pppsxxxv D**

per produrre automaticamente il documento pppDxxxv DOC F

5) **GENPGMN pppsxxxv y vv**

dove la coppia di parametri x e vv identifica lo schema con la sua versione. Quest'ultimo formato è necessario in presenza di più versioni dello stesso schema. Dal momento che vi potrebbero essere applicazioni sviluppate a partire da una certa versione dello schema scheletro e la cui funzionalità sarebbe compromessa da una rigenerazione con una versione nuova, ci si può trovare nella necessità di effettuare comunque la generazione con lo schema originario.

Se la versione dello schema da utilizzare non viene indicata come parametro per la procedura di generazione viene assunta quella obbligatoriamente presente come valore della variabile &VERSCH nel file dei defaults di progetto.

Qualora si stabilisca che un programma sorgente va sempre generato con una certa versione di uno schema, comunque diversa da quella di default, si può introdurre la variabile &VERSCH direttamente nel file delle specifiche anzichè fornirla come parametro di attivazione per la procedura di generazione.

Quando il programma generatore incontra, durante la fase di generazione, delle situazioni non previste emette un messaggio di errore sul terminale per segnalare l'evento. Il messaggio viene anche duplicato su un file degli errori per permetterne una più comoda consultazione. Il file degli errori ha filename uguale al nome programma e filetype uguale ad ERRORI.

Compilazione e testing

Una volta generati la mappa, se necessaria, e il programma sorgente, devono essere effettuate le compilazioni degli oggetti ottenuti.

La procedura di compilazione si chiama **COMPILA.**

Essa richiede in input l'indicazione di filename, filetape e filemode del sorgente da compilare e presenta una serie di maschere successive con eventuali menu per delle scelte.

La prima richiesta è quella della libreria su cui devono essere effettuate le catalogazioni. Per tutti i progetti applicativi la libreria è la TESTLIB.USER (indicata con 1 sul menu corrispondente).

La seconda richiesta è quella del tipo di programma.

Per la compilazione della mappa logica la voce di menu è la 5.

Per la compilazione della mappa fisica la voce di menu è la 6.

Una volta compilata con successo la mappa, può essre compilato il programma, la cui tipologia determina la voce di menu da richiedere (ad esempio 13 per programmi on-line che utilizzano il dbms SQL/DS, 14 per programmi batch sempre con SQL/DS e così via).

Successivamente viene richiesto il nome della fase. Questa indicazione è opzionale e se non espliciatata si ha l'assunzione per default del nome CMS del file.

Il risultato della compilazione viene reso disponibile per la consultazione sulla coda del reader della macchina CMS di sviluppo. Come noto si accede a questa coda tramite il comando **RL**.

Per quanto riguarda i programmi si avrà un listing per ciascuna delle fasi di compilazione a cui è sottoposto dipendentemente dai linguaggi di cui fa uso (ad esempio per un programma COBOL/CICS/SQL vi sarà un listing di ritorno del precompiler SQL, uno del translator CICS ed uno del compiler COBOL).

Come ausilio al debugging dei programmi può essere utilizzato un apposito prodotto. Per informazioni circa il suo uso e funzionamento rimandiamo ai manuali relativi. Ci interessa qui dire che per poterne far uso è necessario compilare il programma con un'altra procedura il cui nome è **COMPINT** ma il cui modo di attivazione e funzionamento è analogo a quella descritta precedentemente.