

**NSIP
Manutenzione MAC e MEV**

Sistema di sviluppo

ISY

Sistema di sviluppo software multiambiente

MANUALE DI SISTEMA

Documento : ISYDSYS0
Aggiornamento : 0
Data : 13 febbraio 1995

**Centro Elaborazione Dati
e Reti di Comunicazione
CONSIGLIO NAZIONALE RICERCHE**

Sommario

INTRODUZIONE.....	1
L'AMBIENTE ON-LINE CICS.....	2
L'ARCHITETTURA DELLE TRANSAZIONI (ISYOTRN).....	3
LA STRUTTURA DELLA COMMUNICATION AREA.....	5
LA PSEUDO-LINK FRA PROGRAMMI.....	6
I TASTI FUNZIONALI (ISYRFUN, ISYRKEY).....	8
LE USCITE STANDARD DAL PROGRAMMA.....	10
L'INTERFACCIA PER LA GESTIONE DEL PANNELLO.....	13
IL SIGN-ON (ISYOSON).....	16
L'ACQUISIZIONE DEI PARAMETRI NELLE TRANSAZIONI (ISYOPAR).....	17
L'HELP ON-LINE.....	20
LA GESTIONE DELLE LISTE (ISYOVIS).....	21
<i>La stampa on-line delle liste (ISYOPRT)</i>	22
LE STAMPE ON-LINE.....	23
L'ARCHITETTURA DEI PROGRAMMI BATCH.....	24
ACQUISIZIONE DEI PARAMETRI NEI PROGRAMMI BATCH (ISYBIPM, ISYRIPM).....	25
CONTROLLO E CONVERSIONE DATI (ISYXCCD).....	30
CONTROLLO ED EDIT DEI DATI ON-LINE (ISYOCDD).....	31
<i>Modalità di controllo dei campi</i>	31
<i>L'HELP di campo (ISYOHCD)</i>	36
<i>I messaggi di errore (ISYOHER)</i>	38
<i>La gestione dei campi obbligatori</i>	38
CONTROLLO ED EDIT DEI DATI IN BATCH (ISYBCDD).....	39
GESTIONE DEGLI ERRORI.....	40
INTERFACCIA PER LE FUNZIONI DI ISYXCCD0.....	41
<i>Funzione di acquisizione per campo alfanumerico</i>	42
<i>Funzioni di gestione per campi numerici</i>	42
<i>Funzioni di gestione date</i>	43
<i>Funzioni di gestione ore</i>	45
<i>Funzioni di gestione tabelle</i>	45
<i>Abilitazione di valori all'utente</i>	46
<i>Funzioni di gestione campi per singolo bit</i>	46
<i>Funzione di gestione del messaggio di errore</i>	47
MODALITÀ DI ATTIVAZIONE ON-LINE.....	49
<i>Controllo dati da pannello</i>	49
<i>Solo controllo on-line</i>	50
<i>Edit on-line</i>	51
MODALITÀ DI ATTIVAZIONE BATCH.....	52
<i>Solo controllo batch</i>	52
<i>Edit batch</i>	52
<i>Attivazione semplificata</i>	53
NORME GENERALI.....	54
DOCUMENTAZIONE DEI PROGRAMMI.....	55
TRACCIATI RECORD.....	57
USO DELLE SYSXXX NEI PROGRAMMI BATCH.....	58
USO DELLE PF NEI PROGRAMMI ON-LINE.....	59
STRUTTURA DEI PANNELLI.....	60
FUNZIONI DI UTILITÀ.....	61

TRASFORMAZIONE SOURCE BATCH IN ON-LINE E VICEVERSA	61
GESTIONE DELLE TABELLE DI CONTROLLO	64
TABELLE PER GESTIONE DEI SINGOLI BIT DI UN BYTE.....	65
CONTROLLO DELL'ABILITAZIONE	66
CONTENUTO DEL FILE SCHELETRO	67
ISTRUZIONI FISSE.....	68
ISTRUZIONI CON DATI VARIABILI	68
<i>IMPOSTAZIONE DI UNA VARIABILE (&SET).....</i>	<i>68</i>
<i>CALCOLO DI UNA VARIABILE (&COMP).....</i>	<i>69</i>
<i>INSERIMENTO DI UNA ROUTINE DA FILE SPECIFICHE (&INCLUDE).....</i>	<i>69</i>
<i>SALTO INCONDIZIONATO (&GOTO).....</i>	<i>69</i>
<i>CONFRONTO (&IF).....</i>	<i>70</i>
<i>CICLO (&LOOP).....</i>	<i>70</i>
<i>RIPETIZIONE DI UN BLOCCO DI ISTRUZIONI (&REPEAT).....</i>	<i>70</i>

Introduzione

Il presente sistema di sviluppo è stato progettato e realizzato per fornire un metodo comune di sviluppo a tutte le applicazioni realizzate in COBOL nei differenti ambienti operativi nei quali è utilizzabile il CICS o un suo emulatore.

In questo documento si vogliono descrivere le regole e gli strumenti da utilizzare per realizzare, con una notevole produttività, applicazioni standard, omogenee, e con un elevato grado di portabilità.

Il sistema consiste in un'architettura di sviluppo delle applicazioni ed in un certo numero di programmi che realizzano in maniera standard alcune funzionalità definite nell'architettura stessa. I programmi applicativi dovranno essere scritti secondo regole che garantiscono la disponibilità delle funzionalità previste nell'architettura.

I programmi sviluppati devono rispondere tutti a precisi standards che garantiscono l'uniformità della realizzazione.

Le funzionalità e l'architettura del sistema sono descritte separatamente in ambiente batch ed online.

Le varie utility di sistema sono descritte una per una e di esse è mostrato il diagramma di flusso.

Un capitolo apposito è dedicato alla descrizione e alle modalità di attivazione del programma ISYxCCD0 che assume importanza centrale nel sistema di sviluppo e cui sono demandati tutti i controlli sui dati ed alcune funzionalità base.

Per quanto riguarda gli ausili previsti per la produzione semiautomatica dei programmi, appositi capitoli sono dedicati alla descrizione del programma *Generatore* con relativo diagramma e al linguaggio da utilizzare negli *Schemi*.

Sono anche descritti in dettaglio tutti gli schemi disponibili tramite i diagrammi dei vari tipi di programmi generabili dove vengono evidenziate le parti appartenenti allo schema e le parti appartenenti alle *Specifiche*.

L'ambiente on-line CICS

Di seguito sono descritti gli aspetti caratteristici del sistema per quanto riguarda lo sviluppo delle applicazioni in ambiente on-line.

Si considera, in tale capitolo, che l'ambiente operativo disponga del CICS, o di un suo emulatore, come monitor on-line. Il CICS, infatti, è venuto affermandosi negli anni sempre di più, e, non solo costituisce la base per le applicazioni on-line di tutti i computer IBM, grandi e piccoli, ma recentemente sono stati realizzati suoi emulatori anche in ambiente UNIX, rendendo efficacemente portabili le applicazioni realizzate nei diversi ambienti.

L'ARCHITETTURA DELLE TRANSAZIONI (ISYOTRN)

Il sistema è disegnato per permettere lo sviluppo dell'applicazione in modalità pseudoconversazionale. Questa modalità disattiva la transazione quando è in attesa della digitazione di dati da parte dell'utente consentendo di abbassare notevolmente il carico del sistema.

In figura 1 è illustrato schematicamente il flusso di una transazione.

Ad ogni transazione è associato in PCT l'unico programma ISYOTRN0 che svolge le funzioni di controllore delle transazioni. Il programma applicativo effettivo viene associato alla transazione in una tabella gestita dal sistema ISY.

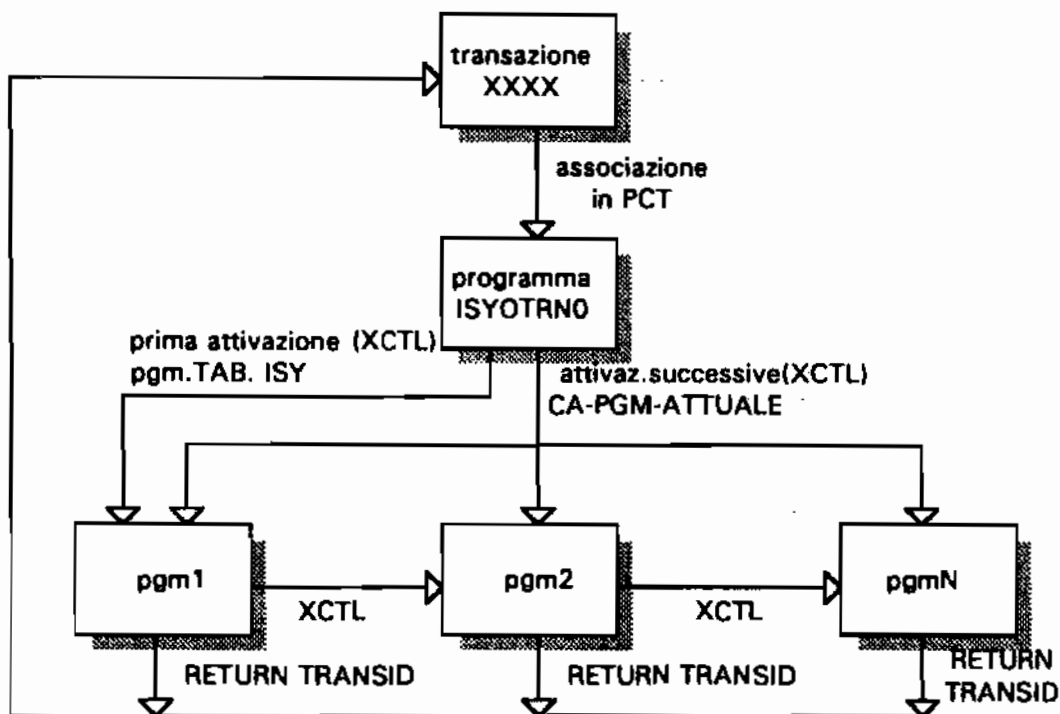


Figura 1 - Flusso tipico di una transazione

Al momento della prima attivazione ISYOTRN0 verificherà l'abilitazione da parte dell'utente all'uso della transazione. Solo in caso di controllo positivo verrà ceduto il controllo al programma applicativo. Il programma ISYOTRN0 provvederà inoltre ad inizializzare le aree di comunicazione di sistema.

La modalità pseudoconversazionale dei programmi applicativi comporta che, dopo aver effettuato la SEND di un pannello, il programma termina riattivando la transazione con un comando di RETURN TRANSID(xxxx).

Alla riattivazione della transazione viene quindi ceduto il controllo nuovamente al programma ISYOTRN0 che in questo caso cederà il controllo al programma che ha effettuato il RETURN TRANSID il cui nome è nella communication area di sistema (vedere descrizione nel capitolo successivo) nel campo CA-PGM-ATTUALE. Questa

tecnica permetterà quindi alle transazioni di strutturarsi con più programmi senza che nessuno dei programmi applicativi si debba occupare di indirizzare l'ultimo programma attivo alla riattivazione della transazione dopo il RETURN TRANSID.

LA STRUTTURA DELLA COMMUNICATION AREA

Il sistema ISY prevede un uso standard della communication area. Questa viene suddivisa in due parti.

La prima è fissa per qualsiasi programma ed è denominata "Communication Area di Sistema". Quest'area contiene campi di base per il funzionamento del sistema ISY, campi di utilità per i vari programmi, e campi di lavoro per programmi di sistema. La maggior parte di questi campi vengono utilizzati ed impostati dai programmi applicativi. L'elenco dei campi contenuti in quest'area con il loro significato è riportato in appendice.

La seconda parte della communication area è invece variabile ed il suo uso è diverso in vari momenti dell'attivazione del programma.

Al momento dell'attivazione di una transazione questa parte può contenere eventuali parametri di attivazione della transazione. Questi parametri potrebbero essere stati forniti con il codice della transazione, nel qual caso il programma ISYOTRNO provvede alla loro estrazione e passaggio nella communication area, o forniti da menù, con la loro acquisizione effettuata dal programma di menù stesso, o costruiti in un programma applicativo (caso di Navigazione). Questi stessi parametri potrebbero anche essere stati forniti sul pannello relativo alla stessa transazione (caso di RICIRCOLO) e quindi spostati in communication area dallo stesso programma applicativo.

Questa tecnica permette di rendere omogenee tutte queste modalità di passaggio dei dati di attivazione, scaricando il programmatore dalla necessità di dover prevedere un trattamento diverso dei dati di ingresso proveniente dalle varie fonti e permettendo alla transazione di essere attivata in una qualsiasi delle modalità descritte.

Un uso diverso di questa seconda parte della communication area si ha all'interno del ciclo del programma. Poiché i programmi funzionano in modalità pseudo-conversazionale, dopo ogni invio del pannello il programma termina rilasciando tutte le aree da esso utilizzate ed in particolare la working storage. Al momento dell'invio dei dati da terminale il programma viene riattivato e tutte le aree vengono reinizializzate. Questa tecnica comporta quindi la necessità di salvare alcuni dati prima di effettuare la chiusura temporanea della transazione. In particolare devono essere salvati tutti i dati acquisiti ad un invio del pannello, in quanto ad ogni nuovo invio arrivano al programma i soli nuovi dati digitati. Il sistema ISY prevede che i dati relativi ad un pannello vengano salvati nella seconda parte della communication area. Prima di effettuare il RETURN TRANSID previsto dalla modalità pseudo-conversazionale, il programma applicativo formerà una communication area costituita dalla parte di sistema seguita dai campi di pannello (in formato interno) utilizzando quest'area nell'istruzione di RETURN TRANSID.

LA PSEUDO-LINK FRA PROGRAMMI

In ambiente CICS le modalità di attivazione di un programma sono di due tipi: XCTL e LINK. La prima modalità prevede che il programma chiamante termini completamente il suo funzionamento cedendo il controllo al chiamato, mentre con la seconda modalità il programma chiamante non termina e verrà riattivato all'istruzione successiva alla LINK quando il programma chiamato termina.

L'uso della LINK deve essere limitato ai casi di effettiva necessità in quanto, mantenendo tutte le risorse acquisite dal programma chiamante, fornisce un appesantimento al CICS. Inoltre se il programma chiamato gestisce un pannello in modalità pseudo-conversazionale la modalità LINK costituisce un inutile aggravio di programmazione per il ritorno al programma chiamato ad ogni riattivazione della transazione.

Il sistema ISY risolve il caso di un sottoprogramma chiamato definendo una modalità di chiamata PSEUDO-LINK.

Per descrivere la sua modalità di funzionamento occorre definire il "LIVELLO DI LINK". Questo è un campo presente nella communication area di sistema (CA-LIVELLO) che contiene un numero di due cifre che rappresenta il livello di profondità della PSEUDO-LINK. Il primo programma attivato con la transazione avrà un livello 0, se questo programma deve richiamare un sottoprogramma che, al suo termine, dovrà tornare al chiamante, sarà di livello 1, e così via. Se un programma deve cedere il controllo ad un altro senza che torri a lui il controllo come previsto normalmente da XCTL, il programma chiamato avrà lo stesso livello logico del chiamante. Quindi in un'istruzione di PSEUDO-LINK il programma chiamato avrà un livello maggiore del chiamante, mentre nella XCTL i due livelli sono uguali.

La gestione della PSEUDO-LINK consiste quindi nell'effettuare l'attivazione del programma successivo comunque con una istruzione XCTL, ma assegnando al programma chiamato un livello superiore. Al completamento del programma chiamato questi provvederà a riattivare il chiamante con la gestione di un indicatore particolare (CA-PGM-MODO) presente in communication area di sistema, che indica la modalità "RIENTRO". Il programma riattivato gestirà questa particolare condizione ricreando le condizioni precedenti alla PSEUDO-LINK.

Dal punto di vista pratico la gestione viene effettuata attraverso i seguenti campi della communication area di sistema:

CA-PGM-ATTUALE	che contiene il nome del programma che ha il controllo
CA-PGM-CHIAM	che contiene il nome del programma chiamante nella pseudo-link o XCTL
CA-PGM-RIENTRO	che contiene il nome del programma a cui deve tornare il controllo quando il programma chiamato termina
CA-LIVELLO	che contiene il livello di link
CA-PGM-MODO	che funziona da indicatore della modalità di attivazione del programma. In particolare: <ul style="list-style-type: none"> • Prima attivazione

- Prima attivazione con passaggio dati in communication area
- Attivazioni successive
- Rientro da pseudo-link

Nel caso di una XCTL il programma chiamante dovrà porre in CA-PGM-ATTUALE il nome del programma chiamato, in CA-PGM-CHIAM il suo nome, mentre lascerà inalterato il campo CA-LIVELLO e CA-PGM-RIENTRO. Il campo CA-PGM-MODO sarà posto al valore di prima attivazione.

Nel caso di PSEUDO-LINK il programma chiamante dovrà porre in CA-PGM-ATTUALE il nome del programma chiamato, mentre porrà il suo nome sia nel campo CA-PGM-CHIAM che in CA-PGM-RIENTRO. Il campo CA-LIVELLO sarà incrementato di 1. Il campo CA-PGM-MODO sarà sempre posto al valore di prima attivazione.

In entrambi i casi il programma chiamato verrà attivato con una XCTL passando la communication area di sistema.

Ogni programma terminerà quindi controllando il campo CA-PGM-RIENTRO. Se in questo campo non trova il nome di un programma (cioè se assume il valore SPACE), interpreta questa condizione come fine della transazione, mentre se vi trova il nome di un programma terminerà riattivando questo programma.

La riattivazione avverrà sottraendo 1 al campo CA-LIVELLO e ponendo il valore di rientro nel campo CA-PGM-MODO.

Questa tecnica, tenendo conto anche del flusso di transazioni già descritto in precedenza, permette di avere sempre attivo solo l'ultimo programma e gestire il rientro al precedente limitando e standardizzando la programmazione.

La gestione del campo CA-LIVELLO permette inoltre di rendere ogni livello indipendente dagli altri. Ogni programma può infatti utilizzare code di temporary storage per memorizzare dati di suo interesse. Quando viene attivato un altro programma a livello superiore questo dovrà poter utilizzare proprie code di temporary storage senza danneggiare quelle del chiamante che le vorrà ritrovare integre al momento della sua riattivazione. Questo viene ottenuto inserendo nel nome della coda di temporary storage il valore di CA-LIVELLO, rendendo così indipendenti le code relative a livelli diversi.

In particolare con questa stessa modalità verrà gestita la coda di temporary storage tttt\$\$ll (tttt è il codice del terminale e ll è il livello), dove deve essere salvata la communication area prima del passaggio ad un livello inferiore. Si è infatti detto che nella communication area vengono poste le informazioni che devono essere mantenute per la riattivazione della transazione in modalità pseudo-conversazionale. Poichè queste informazioni verrebbero distrutte nel passaggio ad un programma successivo, si dovrà procedere al loro salvataggio. Anche la communication area di sistema dovrà essere salvata in modo da consentire un ripristino completo delle condizioni precedenti al momento del rientro. Quando un programma viene riattivato in modalità "rientro" (campo CA-PGM-MODO) questi dovrà immediatamente effettuare il ripristino della communication area e riprenderà il ciclo interrotto.

I TASTI FUNZIONALI (ISYRFUN, ISYRKEY)

L'uso dei tasti funzionali deve essere standardizzato all'interno dei programmi, cioè ad una stessa funzione in transazioni diverse deve corrispondere lo stesso tasto. Gli standard di uso dei tasti sono descritti più avanti in questo documento.

Si vuole qui descrivere la modalità con cui devono essere gestiti i tasti funzionali all'interno del programma. Con questa gestione si vuole inoltre rendere dinamica l'associazione fra tasti funzionali e funzioni in modo da poter modificare tale associazione a livello di sistema senza modifiche sui programmi applicativi.

La parte del programma applicativo destinata all'esecuzione delle funzioni associate ai tasti funzionali opererà secondo codici di funzione e non secondo il particolare tasto funzionale premuto. All'interno dei programmi deve essere contenuta una tabella di nome WS-FUNZIONI composta da 30 righe contenenti i codici delle funzioni gestite all'interno del programma. In communication area di sistema è invece presente una ulteriore tabella (CA-TASTI) le cui righe sono posizionalmente associate ai tasti funzionali, cioè la prima riga è associata al tasto F1, la seconda a F2 e così via. In questa tabella sono contenuti i puntatori ai codici delle funzioni. Come standard la tabella conterrà il valore 01 nella prima riga, il valore 02 nella seconda e così via, facendo quindi corrispondere al tasto funzionale F1 il primo codice funzione, al tasto F2 il secondo e così via. La tabella CA-TASTI viene predisposta dal programma ISYOTRNO all'inizio della transazione. Intervenendo su questo programma è possibile modificare l'associazione. Se infatti tale tabella contenesse i valori 03,02,01,04.... il codice funzione contenuto nella terza riga (END) corrispondente alla fine transazione verrebbe attivato con il tasto funzionale F1, mentre il codice funzione nella prima riga (HLP) verrebbe attivato con F3.

Queste stesse tabelle vengono utilizzate per formare dinamicamente una riga di testo contenente il menù delle funzioni. Questa riga sarà spostata nella ventitreesima riga di pannello riservata a questo scopo. A titolo esemplificativo riportiamo la riga delle funzioni nel caso di un pannello relativo ad una lista che può essere scorsa in avanti o indietro.

F1:HLP F3:END F7:BACK F8:FWD

Si noti che sono presenti solo le PF attive al momento. Dovrà essere gestita l'esclusione delle funzioni non attive in determinate condizioni.

Per la preparazione della riga delle funzioni è stato predisposto il programma ISYRKEY0 richiamabile con una CALL. L'istruzione di attivazione di questo programma sarà:

```
CALL 'ISYRKEY0' USING      CA-AREA
                           WS-FUNZIONI
                           WS-TESTO-PFK.
```

dove WS-FUNZIONI è la tabella delle funzioni attive e WS-TESTO-PFK è la riga delle funzioni.

Nella tabella delle funzioni il programma applicativo dovrà inserire le sole funzioni attive in quel momento lasciando gli altri elementi a SPACE. Il programma ISYRKEY0 formerà infatti la riga inserendo i soli tasti funzionali cui corrisponde nella tabella funzioni un valore diverso da SPACE.

La determinazione di quale funzione è stata richiesta viene effettuata tramite la routine 'ISYRFUNO'. Questa ha in ingresso il campo dell'EIB contenente il codice identificativo del tasto premuto, la tabella delle funzioni e la communication area di sistema. In uscita fornisce nel campo WS-FUNZIONI il codice della funzione associata al tasto.

```
CALL 'ISYRFUNO' USING      CA-AREA  
                           EIBRID  
                           WS-FUNZIONI  
                           WS-TASTO-FUNZIONE.
```

LE USCITE STANDARD DAL PROGRAMMA

In ogni programma applicativo dovranno essere previste le seguenti uscite standard:

010-RETURN-TRANSID

È la routine che gestisce l'uscita pseudo-conversazionale, cioè la disattivazione della transazione dopo l'invio del pannello.

020-USCITA

È la routine che gestisce il normale termine di un programma. Se il programma è stato attivato da un altro con una pseudo-link (campo CA-PGM-RIENTRO diverso da SPACE) viene riattivato quest'ultimo impostando CA-PGM-MODO a rientro. Se invece il programma era il primo della transazione, viene inviato il messaggio che in questo caso deve essere stato impostato obbligatoriamente nel campo WS-MSG-ERRORE e viene terminata la transazione.

020-USCITA-ANN-OPER

È l'uscita dalla transazione quando viene premuto il tasto corrispondente all'annullamento dell'operazione (default tasto CLEAR). È equivalente alla routine 020-USCITA con in più la gestione del messaggio di annullamento operazione (MSG-ANNOPER).

030-XCTL-RETURN

Gestisce la pseudo-link fra programmi. Prima dell'attivazione di questa routine dovrà essere stato impostato il campo WS-PGM-XCTL con il nome del programma da attivare. La routine effettuerà il salvataggio delle aree e l'impostazione dei campi della communication area di sistema come descritto nel capitolo sulla pseudo-link.

030-XCTL-NORET

Gestisce l'attivazione di un altro programma allo stesso livello logico del precedente (XCTL reale). Anche in questo caso il nome del programma da attivare deve essere stato inserito nel campo WS-PGM-XCTL. La routine effettuerà solo la predisposizione dei campi della communication area di sistema, non essendo necessario alcun salvataggio di aree.

040-BLOCCA-ATTIVAZIONE

Viene utilizzata solo in caso di attivazione anomala del programma (senza communication area) e termina la transazione inviando un messaggio di errore.

Di seguito vengono mostrate le istruzioni relative alle uscite standard.

```

*****
*****
*-----USCITE STANDARD DAL PROGRAMMA-----*
*****
*****
* - RETURN TRANSID (USCITA PSEUDOCONVERSATIONALE) *
*****

010-RETURN-TRANSID.
MOVE ALTRE-ENTRATE-K TO CA-PGM-MODO.
MOVE CA-AREA-SISTEMA TO TRANSID-CASYS.

EXEC CICS RETURN TRANSID (KIBTRNID)
      COMMAREA (DATI-TRANSID)
      LENGTH (DATI-TRANSID-LNT) END-EXEC.

010-RETURN-TRANSID-EX.
EXIT.

*****
* - USCITA (USCITE DALLA TRANSAZIONE) *
* * * * *
* - USCITA-ANN-OPER (USCITA PER ANNULLAMENTO OPERAZIONE) *
* * * * *
*****

020-USCITA-ANN-OPER.
MOVE MSG-ANNOPER TO WS-MSG-ERRORE.

020-USCITA.
MOVE WS-MSG-ERRORE TO CCD-ER-CODICE.
IF CA-PGM-RIENTRO NOT = SPACE
MOVE RIENTRO-K TO CA-PGM-MODO
MOVE CCD-ER-CODICE TO CA-RET-CODE
EXEC CICS XCTL PROGRAM (CA-PGM-RIENTRO)
      COMMAREA (CA-AREA)
      LENGTH (CA-AREA-SYSLNT) END-EXEC
ELSE
PERFORM 310-PREP-MESSAGGI THRU 310-PREP-MESSAGGI-EX
EXEC CICS SEND TEXT FROM (CCD-ER-MESS-FULL)
      LENGTH(46)
      ALARM KHASH FREEKB END-EXEC
EXEC CICS RETURN END-EXEC.

020-USCITA-EX.
EXIT.

*****
* - XCTL (PASSAGGIO CONTROLLO AD ALTRO PROGRAMMA) *
* * * * *
* - XCTL-RETURN (PASSAGGIO CONTROLLO CON RITORNO) *
* * * * *
* - XCTL-NORETURN (PASSAGGIO CONTROLLO SENZA RITORNO) *
* * * * *
*****

```

030-XCTL-RETURN.
 PERFORM 170-SAVE-ARRE THRU 170-SAVE-ARRE-EX.

MOVE CA-PGM-ATTUALE TO CA-PGM-CHIAM.
 MOVE CA-PGM-ATTUALE TO CA-PGM-RIENTRO.
 MOVE WS-PGM-XCTL TO CA-PGM-ATTUALE.
 GO TO 030-0-XCTL.

030-XCTL-NORETURN.
 MOVE CA-PGM-ATTUALE TO CA-PGM-CHIAM.
 MOVE WS-PGM-XCTL TO CA-PGM-ATTUALE.

030-0-XCTL.
 MOVE PRIMA-ENTRATA-K TO CA-PGM-MODO.

 EXEC CICS XCTL PROGRAM (CA-PGM-ATTUALE)
 COMAREA (CA-AREA)
 LENGTH (CA-AREA-SYSLMT) END-EXEC.

030-XCTL-EX.
 EXIT.

 * - BLOCCA-ATTIVAZIONE (ERRORE DI ATTIVAZIONE) *
 * *

040-BLOCCA-ATTIVAZIONE.
 MOVE MSG-ERR-ATTIVAZIONE TO CCD-ER-CODICE.

 PERFORM 310-PREP-MESSAGGI THRU 310-PREP-MESSAGGI-EX.

 EXEC CICS SEND TEXT FROM(CCD-ER-MESS-FULL)
 LENGTH(46)
 ALARM ERASE PEEKFB END-EXEC
 EXEC CICS RETURN END-EXEC.

040-BLOCCA-ATTIVAZIONE-EX.
 EXIT.

L'INTERFACCIA PER LA GESTIONE DEL PANNELLO

Per la gestione del pannello devono essere predisposte all'interno del programma le seguenti aree:

Area dei campi in formato interno: è l'area che formerà la communication area nell'istruzione RETURN TRANSID insieme con la communication area di sistema.

Area dei campi in formato esterno: contiene la definizione di tutti i campi di mappa con un nome significativo. Questi campi verranno utilizzati all'interno del programma per formare il pannello da visualizzare rendendo più semplice e chiara la scrittura e rendendo il programma indipendente dal sistema di gestione del pannello (BMS). La routine 240-PREPARA-MAPPA provvederà a spostare questi campi sulla mappa logica prima dell'invio del pannello e la routine 210-PREPARA-CONTR effettuerà l'operazione opposta riempiendo quest'area a partire dal pannello.

Area delle caratteristiche del campo: contiene per ognuno dei campi presenti su pannello le sue caratteristiche e cioè:

- LNT lunghezza del campo
- ATT attributo di protezione/luminosità
- POS posizione in mappa (progressivo del carattere iniziale all'interno del pannello)
- NOM nome esterno del campo
- FUN controllo che deve essere effettuato sul campo

Suddette informazioni sono presenti anche per i tre campi sempre presenti in mappa WS-TESTO-PFK (tasti funzionali), WS-MESSAGGI (messaggi di errore), WS-COMANDI (dati presenti dopo ==>).


```

*-----*
*   CAMPI DI INTERFACCIA CON PANNELLO   *
*-----*

*   CAMPI IN FORMATO ESTERNO

01 CAMPI-MAPPA-E.
02 CAMPI-MAPPA-E-SYS.
04 WS-COMANDI-E          PIC X(30)      VALUE SPACES.
02 CAMPI-MAPPA-E-PGM.
04 nome-campo1-E        PIC X(10)
                          VALUE LOW-VALUE.
04 nome-campo2-E        PIC X(03)
                          VALUE LOW-VALUE.
.....

01 CAMPI-MAPPA-E-LNT     PIC S9(4) COMP  VALUE +271.

*   CAMPI IN FORMATO INTERNO

01 CAMPI-MAPPA-I.
02 CAMPI-MAPPA-I-SYS.
04 WS-COMANDI-I          PIC X(030)     VALUE SPACES.
04 WS-FUNZIONI-I        PIC X(090)     VALUE SPACES.
02 CAMPI-MAPPA-I-PGM.
04 nome-campo1-I        PIC X(010)
                          VALUE SPACES.
04 nome-campo2-I        PIC X(003)
                          VALUE SPACES.
.....

01 CAMPI-MAPPA-I-LNT     PIC S9(4) COMP  VALUE +253.

*-----*
*   CARATTERISTICHE DEI CAMPI IN MAPPA   *
*-----*

01 CAMPI-MAPPA-INF.
02 INF-WS-TESTO-PFK.
04 WS-TESTO-PFK-LNT     PIC 99      VALUE 78.
04 WS-TESTO-PFK-ATT     PIC X       VALUE SPACE.
04 WS-TESTO-PFK-UNPROT  VALUE ' ' 'H' 'I' '<'.
04 WS-TESTO-PFK-POS     PIC 9999    VALUE 01761.
04 WS-TESTO-PFK-NOM     PIC X(18)   VALUE 'TESTO-PFK'.
04 WS-TESTO-PFK-FUN     PIC X(10)   VALUE SPACES.
02 INF-WS-MESSAGGI.
04 WS-MESSAGGI-LNT     PIC 99      VALUE 42.
04 WS-MESSAGGI-ATT     PIC X       VALUE 'Y'.
04 WS-MESSAGGI-UNPROT  VALUE ' ' 'H' 'I' '<'.
04 WS-MESSAGGI-POS     PIC 9999    VALUE 01841.
04 WS-MESSAGGI-NOM     PIC X(18)   VALUE 'MESSAGGI'.
04 WS-MESSAGGI-FUN     PIC X(10)   VALUE SPACES.
02 INF-WS-COMANDI.
04 WS-COMANDI-LNT     PIC 99      VALUE 30.
04 WS-COMANDI-ATT     PIC X       VALUE SPACE.
04 WS-COMANDI-UNPROT  VALUE ' ' 'H' 'I' '<'.
04 WS-COMANDI-POS     PIC 9999    VALUE 01889.
04 WS-COMANDI-NOM     PIC X(18)   VALUE 'COMANDI'.
04 WS-COMANDI-FUN     PIC X(10)   VALUE 'ALSO'.
02 INF-nome-campo1.
04 nome-campo1-LNT     PIC 99      VALUE 10.
04 nome-campo1-ATT     PIC X       VALUE SPACE.
04 nome-campo1-UNPROT  VALUE ' ' 'H' 'I' '<'.

```

```

04 nome-campo1-POS      PIC 9999 VALUE 00275.
04 nome-campo1-NOM      PIC X(18)
                        VALUE 'nome-campo-1'.
04 nome-campo1-FUN      PIC X(10)
                        VALUE 'AL10'.
02 INF-nome-campo2.
04 nome-campo2-INT      PIC 99  VALUE 10.
04 nome-campo2-ATT      PIC X   VALUE SPACE.
04 nome-campo2-POS      PIC 9999 VALUE 00275.
04 nome-campo2-NOM      PIC X(18)
                        VALUE 'nome-campo-2'.
04 nome-campo2-FUN      PIC X(10)
                        VALUE 'AL10'.
.....
01 FILLER REDEFINES CAMPI-MAPPA-INF.
02 INF-FLDATT           OCCURS 21.
04 FLDATT-INT          PIC 99.
04 FLDATT-ATT          PIC X.
04 FLDATT-POS          PIC 9999.
04 FLDATT-NOM          PIC X(18).
04 FLDATT-FUN          PIC X(10).
01 CAMPI-MAPPA-NUM      PIC 999  VALUE 21.
01 MAPPA-INT           PIC 89(4) COMP VALUE +466.

```

IL SIGN-ON (ISYOSON)

Il programma ISYOSON0 è quello che gestisce il sign on all'interno del sistema ISY. Questo si basa su archivio (ISYUTE) che contiene per ogni utente, identificato da un codice di 8 caratteri, la password ed una serie di informazioni aggiuntive. La password è presente sull'archivio in forma crittografata in modo da non poter essere individuata in caso di accesso diretto all'archivio. Poichè la password ed il codice utente sono due informazioni completamente separate con questo sistema è possibile consentire all'utente stesso di poter modificare la propria password.

Il programma ISYOSON0 permette quindi l'accesso al sistema quando l'utente ha fornito il proprio codice e la relativa password. Ogni tentativo di accesso errato effettuato con un codice utente viene conteggiato dal sistema. Al momento di un sign-on vengono fornite informazioni sul sign-on precedente e sugli eventuali tentativi di sign-on su quello stesso codice utente.

Il sign-on consiste nella scrittura di un record di temporary storage avente chiave il codice del terminale su cui il sign-on è effettuato e la costante "\$SON". Viene inoltre gestita un'ulteriore coda unica per tutti i terminali avente come identificatore la stringa costante "****\$SON". Su questa ulteriore coda viene inserito un item al momento del sign-on. L'item contiene come unica informazione il codice del terminale su cui il sign-on stesso è stato effettuato. La lettura di questa coda permette quindi di conoscere i terminali attivati e di formare la chiave delle singole code di temporary storage. Con queste chiavi è possibile accedere al dettaglio delle informazioni di sign-on. Al momento del sign-off, oltre alla cancellazione della coda associata al terminale, viene azzerato l'item corrispondente sulla coda "****\$SON". Il numero dell'item è presente sulla coda di terminale, in cui è salvato al momento del sign-on.

Tra le varie informazioni associate all'utente è previsto un indicatore dell'ambiente sul quale l'utente stesso dovrà lavorare. La presenza di questo indicatore viene interpretata come la necessità di avere delle elaborazioni particolari oltre al sign-on ISY. In questo caso viene ceduto il controllo al programma ISYOPSO0 che dovrà gestire le particolarità dell'utente.

La modalità di funzionamento di ISYOPSO0 vista da ISYOSON0 è la stessa di ISYOCCD0 vista da un programma applicativo. L'area di interfaccia è sempre ISYCCCD. Il programma ISYOHCD0 otterrà in CCD-ESTERNO gli eventuali dati particolari del sign-on (acquisiti in un campo del pannello di sign-on). In caso di errore dovrà impostare il relativo codice in CCD-ERRORE che in caso contrario dovrà essere posto a SPACE. Eventuali informazioni di ritorno possono essere restituite in CCD-ESTERNO. ISYOSON0 provvederà a gestire secondo le modalità standard il messaggio di errore e l'edit del campo.

L'ACQUISIZIONE DEI PARAMETRI NELLE TRANSAZIONI (ISYOPAR)

Ogni transazione ha generalmente necessità di acquisire dei dati necessari alla sua attivazione. Questi dati sono normalmente delle chiavi di accesso agli archivi. I dati di attivazione possono essere forniti con il codice della transazione o al momento del menù o possono essere forniti attraverso un pannello appositamente predisposto.

Il sistema ISY, oltre ad uniformare le prime due modalità con l'uso della communication area (vedi capitolo ad essa relativo) fornisce il programma generalizzato ISYOPAR0 per l'acquisizione di queste informazioni da pannello.

Un programma applicativo che vuole utilizzare le funzionalità di ISYOPAR0 dovrà inserire nella coda di temporary storage tttt\$Pnn (dove tttt è il terminale e nn è il livello di link) un record per ogni parametro che deve essere acquisito. Il tracciato di questo record è contenuto nella copy ISYCPAR ed in generale contiene una stringa di caratteri descrittiva del parametro, un campo per il dato acquisito, il codice della funzione di controllo da effettuare sul campo, l'eventuale indicazione di campo obbligatorio, la lunghezza del campo in mappa, la riga del pannello su cui si vuole far comparire la richiesta del parametro.

Il programma ISYOPAR0 gestisce un pannello generalizzato di 20 righe per l'acquisizione dei parametri. Ogni riga è dedicata all'acquisizione di un parametro ed è divisa in due parti. La parte a sinistra è sempre protetta e su di essa viene spostata la stringa descrittiva del parametro. La parte a destra viene formattata con la maschera di acquisizione del parametro e permette di inserire il valore del parametro.

Ogni parametro viene inserito sulla riga che è stata indicata nel corrispondente record di temporary storage, permettendo quindi di ottenere un pannello di migliore presentazione.

ISYOPAR0 effettua il controllo formale dei dati forniti a terminale gestendo le relative segnalazioni di errore. Quando tutti gli errori sono stati corretti ISYOPAR0 forma con i parametri acquisiti una stringa nel formato previsto per la communication area e riattiva il programma chiamante con questi dati. Questa attivazione è quindi equivalente a quella iniziale ed il programma effettua sui dati i controlli logici. Se il controllo ha esito negativo viene riattivato ISYOPAR0 e ripetuto il ciclo. Nell'altro caso il programma prosegue con la logica richiesta dalla transazione.

Per l'utilizzazione del programma ISYOPAR0 occorrerà predisporre un'area per la definizione dei parametri da acquisire. Quest'area verrà denominata CAMPI-ATTIV ed avrà il seguente formato:

```

01 CAMPI-ATTIV.
02 CAMPI-nome-parametro
04 nome-parametro-D          PIC X(18)
                              VALUE 'stringa descriz. campo'.
04 nome-parametro-E          PIC X(60) VALUE SPACE.
04 nome-parametro-F          PIC X(18) VALUE 'funzione'.
04 nome-parametro-O          PIC X    VALUE '0'.
04 nome-parametro-L          PIC $$$  VALUE 111.
04 nome-parametro-R          PIC 99   VALUE 99.
04 nome-parametro-X          PIC X(60) VALUE LOW-VALUE.
04 nome-parametro-I          REDEFINES nome-parametro-X
                              PIC PFFFF.
    
```

04	nome-parametro-N	PIC X(18)	VALUE 'nome-parametro'
02		
01	FILLER REDEFINES CAMPI-ATTIV.		
02	CAMPI-ATTIV-LEN OCCURS n	PIC X(162)	
01	CAMPI-ATTIV-NUM	PIC 9(4)	VALUE n.
01	CAMPI-ATTIV-PR	PIC 9(4)	VALUE ZERO.

Come si può vedere i campi relativi ad un parametro hanno tutti una radice coincidente con il nome del parametro ed un suffisso di un carattere. In particolare:

- D stringa descrittiva del campo
- E area riservata al valore esterno
- F funzione di controllo
- O indicatore campo obbligatorio (se 'O')
- L lunghezza del campo in mappa
- R riga del pannello generalizzato su cui il campo deve essere richiesto
- X campo riservato al valore in formato interno; viene ridefinito dal seguente
- I campo in formato interno con Picture corrispondente al formato stesso.
- N nome del campo (utilizzato nell'help).

Dovranno inoltre essere predisposte due routine, una per la scrittura della coda di T.S. con i parametri da acquisire (180-WRT-PARAMETRI) ed una per l'estrazione dei parametri stessi dalla communication area (150-ESTRAE-PARAM).

```

*****
* - WRT-PARAMETRI (SCRITTURA PARAMETRI ATTIVAZIONE) *
*****

180-WRT-PARAMETRI.
  MOVE CA-LIVELLO TO TSPAR-LIVELLO.
  MOVE EINTRMID TO TSPAR-TERMINALE.
  MOVE ZERO TO TSPAR-ITEM.

  EXEC CICS IGNORE CONDITION QIDERR END-EXEC.

  EXEC CICS DELETEQ TS QUEUE (TSPAR-KEY) END-EXEC.
  MOVE ZERO TO CAMPI-ATTIV-PTR.

  PERFORM 180-1-WRT-SCHEDA THRU 180-1-WRT-SCHEDA-EX
    CAMPI-ATTIV-MUM TIMES.

180-WRT-PARAMETRI-EX.  EXIT.

180-1-WRT-SCHEDA.
  ADD 1 TO CAMPI-ATTIV-PTR.
  ADD 1 TO TSPAR-ITEM.
  MOVE CAMPI-ATTIV-ELM (CAMPI-ATTIV-PTR) TO TSPAR-RECORD.

  EXEC CICS WRITEQ TS QUEUE (TSPAR-KEY)
    FROM (TSPAR-RECORD)
    LENGTH (TSPAR-RECLNT)
    ITEM (TSPAR-ITEM) END-EXEC.

180-1-WRT-SCHEDA-EX.  EXIT.

*****
* - ESTRAR-PARAM (ESTRAZIONE PARAMETRI DA COMM.AREA) *
*****

150-ESTRAE-PARAM.
  MOVE 1 TO COMMAREA-RESTO-PTR.
  COMPUTE COMMAREA-RESTO-LNT = EIBCALEN - CA-AREA-SYSLST.
  IF COMMAREA-RESTO-LNT NOT > 0
    GO TO 150-ESTRAE-PARAM-EX.

  MOVE ZERO TO WS-ELEMENTO-PTR.

  PERFORM 150-0-ESTRAE-CAR THRU 150-0-ESTRAE-CAR-EX
    UNTIL COMMAREA-RESTO-PTR > COMMAREA-RESTO-LNT
    OR COMMAREA-RESTO-CAR (COMMAREA-RESTO-PTR)
      NOT = SPACE.

  IF COMMAREA-RESTO-PTR > COMMAREA-RESTO-LNT
    GO TO 150-ESTRAE-PARAM-EX.

  MOVE ZERO TO WS-ELEMENTO-PTR.
  MOVE SPACE TO WS-ELEMENTO.

  PERFORM 150-0-ESTRAE-CAR THRU 150-0-ESTRAE-CAR-EX
    UNTIL COMMAREA-RESTO-PTR > COMMAREA-RESTO-LNT
    OR COMMAREA-RESTO-CAR (COMMAREA-RESTO-PTR)
      = ' '.

  MOVE WS-ELEMENTO TO nome-primo-parametro-ex.
  ADD 1 TO COMMAREA-RESTO-PTR.

150-ESTRAE-PARAM-EX.  EXIT.

150-0-ESTRAE-CAR.
  ADD 1 TO WS-ELEMENTO-PTR.
  MOVE COMMAREA-RESTO-CAR (COMMAREA-RESTO-PTR)
    TO WS-ELEMENTO-CAR (WS-ELEMENTO-PTR).
  ADD 1 TO COMMAREA-RESTO-PTR.

150-0-ESTRAE-CAR-EX.  EXIT.

```


L'HELP ON-LINE

La chiamata al programma di HELP è effettuata secondo due modalità: per i campi di input digitando il carattere "?" sul campo stesso, oppure anche per gli altri tipi di campo, posizionando il cursore sul campo e premendo il tasto di HELP (default F1). Mentre per la prima modalità non è necessaria nessuna programmazione in quanto la relativa gestione è completamente a carico del programma di controllo (ISYOCCD0), descritto più avanti in questo documento, per la seconda è necessario inserire nel programma applicativo le istruzioni necessarie alla gestione.

La routine di gestione dell'HELP dovrà essere attivata immediatamente dopo la receive del pannello e la traduzione del tasto funzionale nel relativo codice di funzione. Le istruzioni saranno quindi le seguenti:

```

RECEIVE .....
CALL 'ISYRFUN0' USING CA-AREA
                        KIBRID
                        WS-FUNZIONI
                        WS-TASTO-FUNZIONE.

IF WS-TASTO-FUNZIONE = WS-FUNZ-COD-HELP
PERFORM 205-GEST-HELP THRU 205-GEST-HELP-EX
MOVE WS-FUNZ-COD-ENTER TO WS-TASTO-FUNZIONE.
    
```

La routine di help sarà costituita dalla chiamata al programma **ISYRCUR0** che fornisce, a partire dalla posizione del cursore e dalla tabella con le caratteristiche dei campi le informazioni sul nome del campo e sul controllo che viene effettuato su di esso. Questi dati dovranno essere posti negli appositi campi della copy ISYCCCD che funziona da interfaccia con il programma di HELP. Le istruzioni sono quindi le seguenti:

```

205-GEST-HELP.
CALL 'ISYRCUR0' USING KIBPOS
                        CAMPI-MAPPA-CAR
                        CAMPI-MAPPA-NUM
                        CCD-AREA.

EXEC CICS LINK PROGRAM (CA-ISYONER)
                        COMAREA (CCD-AREA)
                        LENGTH (CCD-AREA-LNT) END-EXEC.

205-GEST-HELP-EX. EXIT.
    
```

Si noti che il nome del programma di HELP è in forma simbolica utilizzando il campo in communication area di sistema contenente il suo nome.

LA GESTIONE DELLE LISTE (ISYOVIS)

Allo scopo di semplificare la realizzazione delle applicazioni è disponibile il programma ISYOVIS0 che gestisce la visualizzazione di dati in formato "lista" quando cioè il formato del pannello consiste in una testata ed in una serie di righe di dettaglio. Con l'uso di questo programma generalizzato il programmatore si dovrà unicamente preoccupare della gestione della lettura degli archivi e della formazione delle righe di visualizzazione e sarà scaricato dalla gestione del pannello di visualizzazione (es. dalla gestione dello scorrimento in avanti e indietro della lista).

La comunicazione tra il programma applicativo ed il programma ISYOVIS0 viene effettuata attraverso la coda di temporary storage tttt\$Vnn. Il programma applicativo scriverà su questa coda tutti i dati che è necessario visualizzare ed alcune informazioni generali. Il programma ISYOVIS0 utilizzerà queste informazioni generali per organizzare il pannello ed accederà alla coda per leggere i dati da visualizzare.

Le informazioni generali dovranno essere inserite dal programma applicativo nel primo item della coda di temporary storage secondo il tracciato contenuto nella copy ISYCVIS. Le righe successive conterranno le varie righe di testata e successivamente tutte le linee di dettaglio progressivamente senza nessuna gestione di salto pagina o altro.

Un elemento di dettaglio può essere anche costituito da più righe di dettaglio, è cioè possibile che le informazioni relative agli elementi della lista non possano essere visualizzati su una sola riga ma ad essi siano dedicate più righe. Comunque nella coda di temporary storage verranno scritte le singole righe indipendentemente dall'aggregazione in elementi.

Su un elemento di temporary storage può essere effettuata anche una selezione con l'attivazione di una funzione particolare su di essa. La funzione viene ottenuta con l'attivazione di un programma applicativo che gestisce la funzione stessa. A questo programma applicativo dovranno essere passati i parametri di attivazione nel modo già indicato per permettergli di operare sui dati selezionati. Nel caso in cui debba essere gestita questa possibilità, per ogni elemento, nel successivo item di temporary storage, devono essere fornite delle chiavi che rappresentano i parametri di attivazione di tutti i programmi attivabili. Ognuna di queste chiavi sarà separata dalla successiva dal carattere "!" ed individuata da un progressivo. Così la prima chiave sarà la "1", la seconda la "2" e così via. L'uso di queste chiavi verrà descritto più avanti.

Il primo item della coda dovrà contenere in particolare le seguenti informazioni:

TSVIS-NRIGHE-TES numero di righe di testata

TSVIS-NRIGHE-REC numero di righe di dettaglio corrispondenti ad un elemento di lista.

Segue una tabella di 6 item contenente:

TSVIS-CAR-SEL (n)	carattere di selezione
TSVIS-PGM-LNK (n)	indicatore della modalità di attivazione del programma (L per pseudo-link , X per XCTL)
TSVIS-PGM-SEL (n)	nome del programma da attivare
TSVIS-PGM-DES (n)	descrizione della funzionalità del programma
TSVIS-PGM-KEY (n)	chiavi che devono essere passate al programma

L'ultimo blocco di informazioni è opzionale e consente di gestire l'attivazione delle funzioni particolari sugli elementi visualizzati sulla lista. Su ogni elemento di lista (sulla prima riga se l'elemento si sviluppa su più righe) viene gestito sul pannello un campo di selezione di un carattere. In questo blocco costituito da una tabella di 6 elementi possono essere forniti fino ad un massimo di 6 caratteri di selezione. Ad ognuno di questi caratteri è associato un programma applicativo. Il programma ISYOVIS0 gestisce questo campo di selezione. Se in questo campo viene digitato uno dei caratteri previsti, ISYOVIS0 passa il controllo al programma associato secondo la modalità indicata in TSVIS-PGM-LNK fornendo in communication area le chiavi indicate in TSVIS-PGM-KEY secondo le normali modalità previste nell'attivazione dei programmi.

La stampa on-line delle liste (ISYOPRT)

Con la pressione del tasto funzione F6 si ottiene la stampa, sulla stampantina collegata logicamente al terminale, di quanto presente in coda di TS e in corso di visualizzazione.

ISYOVIS 'sente' il tasto funzione ed effettua le seguenti operazioni:

- copia la coda TS dandogli come nome il PRNT + il n° di task;
- START TRANSID su TERMD (CA-UTE-STAMP) di TRANSID ('PRNT') FROM (un'area dove è definito il nome della coda TS anzidetto) LENGTH (la lunghezza dell'area anzidetta); PRNT è associata al programma ISYOPRT direttamente tramite PCT;

ISYOPRT effettua le seguenti operazioni:

- RETRIEVE i dati di START TRANSID
- legge la coda e effettua la stampa analogamente a ISYOVIS (testate e salti pagina tenendo conto del numero di righe fisiche del form definito nelle Specifiche)
- cancella la coda utilizzata.

LE STAMPE ON-LINE

Le transazioni di stampa on-line sono, generalmente, gestite da due attivazioni dello stesso programma. La prima (associata al video richiedente) effettua l'acquisizione ed il controllo dei dati di attivazione tramite il colloquio con il programma di utilità ISYOPARO secondo le modalità già viste in precedenza, verificandone la validità formale e sostanziale.

Fatto questo, produce la seconda attivazione indirizzata sulla stampante, il cui codice è stato preventivamente definito per l'utente o da lui modificato, passando i dati di attivazione in Communication Area.

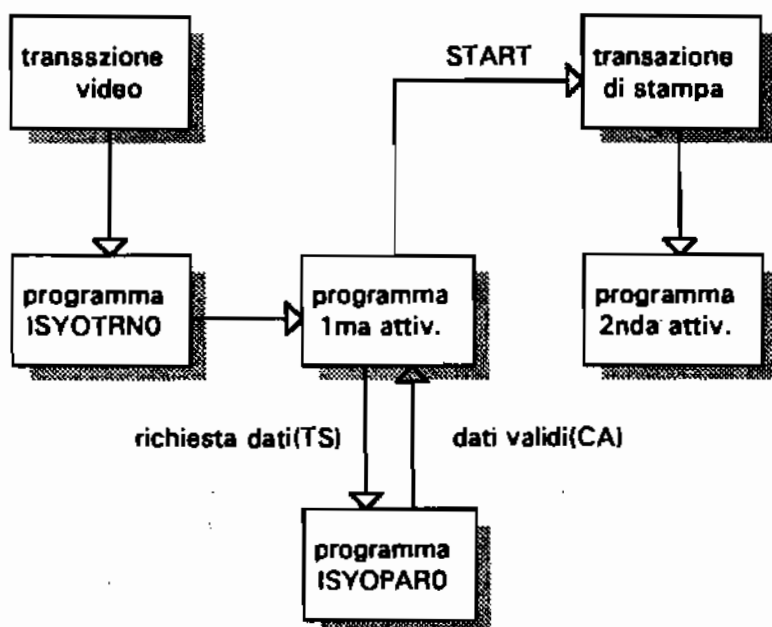


Figura 4 - Flusso tipico di transazioni di stampa on-line

Al fine di omogeneizzare la scrittura di un programma di stampa on-line con quella di un programma di stampa batch, la routine 400-STAMPA-RIGA si assume completamente il carico di gestire il buffer delle stampanti 3270. Questa routine viene richiamata all'interno dei programmi ogni volta che si vuole eseguire la stampa di una riga.

L'ARCHITETTURA DEI PROGRAMMI BATCH

La struttura di programmi batch nel sistema di sviluppo non ha esigenze particolari rispetto a quelle di un normale programma strutturato.

Il sistema di sviluppo mette comunque a disposizione del programmatore il programma generalizzato di controllo (ISYBCCD0) ed un programma per la gestione delle schede parametro e del separatore (ISYBIPM).

ACQUISIZIONE DEI PARAMETRI NEI PROGRAMMI BATCH (ISYBIPM, ISYRIPM)

Il sistema ISY prevede che i programmi debbano poter funzionare sia ricevendo l'input da normali schede parametro inserite nella Job Stream, sia da un archivio in cui siano state registrate le richieste off-line da elaborare.

Per ottenere questo risultato è stata predisposta la routine ISYRIPM0 che gestisce l'acquisizione dei parametri di elaborazione dalle varie fonti possibili.

Questa routine, con la stessa modalità utilizzata per ISYBCCD0, effettua un link-edit dinamico con il programma ISYBIPM0 che contiene la logica di gestione.

Oltre alla funzione di acquisizione parametri, il programma provvederà a stampare sulla stampante di sistema i parametri di elaborazione e sulle stampanti indicate un separatore con il nome dell'utente per facilitare la distribuzione. Questo programma verrà guidato nella sua elaborazione da una scheda parametro presente nella Job Stream.

La scheda parametro ha il seguente formato:

```
*##PARxxxxxyyyyyyttttasxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

La variabile xxxxxx rappresenta la provenienza dell'input e può assumere i seguenti valori:

NOPAR	nessun parametro previsto per questa transazione
OFFL	Input dal file delle richieste off-line
RDR	Input dal lettore di schede (default)
END	Fine dei dati in input dal lettore di schede
SAVE	Input dal work-file su cui sono stati salvati i parametri acquisiti in precedenza

La stringa yyyyyy rappresenta le eventuali opzioni. I valori attualmente previsti sono:

SPAZIO	Nessuna opzione
SAVE	Le schede parametro lette in ingresso devono essere salvate sul work-file per successive riletture
NOPRT	Non deve essere effettuata la stampa automatica delle schede di ingresso sulla stampante di sistema
SAVNOP	È la combinazione delle opzioni SAVE e NOPRT

Le opzioni SAVE e SAVNOP non possono essere richieste se sulla provenienza dell'input è indicato SAVE.

Il parametro *tttt* è il codice della transazione associata al programma.

Il parametro *a* è un codice identificativo dell'ambiente su cui l'elaborazione verrà effettuata. Le sigle previste sono:

- T Ambiente di test
- P Ambiente di produzione

La stringa *zzzz...* rappresenta il nominativo dell'utente che può essere fornito per ottenerne la stampa sul separatore se l'input non è dalle richieste off-line (RDR).

Nel caso di acquisizione dal lettore di schede la fine delle schede parametro può essere segnalata da questo programma o pilotata dal programma chiamante. Nel primo caso le schede parametro devono essere seguite da una scheda *###PAREND* gestita dal programma come fine dati. Nel secondo il programma chiamante provvederà a non effettuare ulteriori CALL a ISYRIPM0 dopo avere acquisito l'ultima scheda necessaria; in questo caso NON deve essere inserita la scheda *###PAREND*. Se è richiesta l'opzione SAVE la fine delle schede deve essere pilotata da ISYRIPM0 in quanto deve effettuare la chiusura del work-file.

La routine ISYRIPM0 segnala la fine dei parametri in ingresso con una scheda parametro a LOW-VALUE.

La routine ISYRIPM0 potrà essere richiamata all'interno dei programmi batch con la seguente modalità:

```

*****
WORKING-STORAGE SECTION.
*****
01 SW-STAMPANTI.
02 TXT-PRTO PIC X(20) VALUE SPACE.
02 TXT-PRT1 PIC X(20) VALUE SPACE.
02 TXT-PRT2 PIC X(20) VALUE SPACE.
02 TXT-PRT3 PIC X(20) VALUE SPACE.
02 TXT-PRT4 PIC X(20) VALUE SPACE.
02 TXT-PRT5 PIC X(20) VALUE SPACE.
02 FILLER PIC X(20) VALUE SPACE.
02 FILLER PIC X(20) VALUE SPACE.
02 FILLER PIC X(20) VALUE SPACE.
02 FILLER PIC X(20) VALUE SPACE.
01 SCHEDA-PARAMETRO PIC X(80) VALUE SPACE.
*****

PROCEDURE DIVISION.
*****
MOVE testo-id-stampa TO TXT-PRTO.
MOVE nome-programma TO CA-PCM-ATTUALE.
PERFORM ACQUISIZIONE-SCHEDA THRU
ACQUISIZIONE-SCHEDA-EX
UNTIL SCHEDA-PARAMETRO = LOW-VALUE.
*****

```

```

ACQUISIZIONE-SCHEDA.
CALL 'ISYRIPM0' USING CA-AREA
                                SW-STAMPANTI
                                SCHEDA-PARAMETRO.
IF SCHEDA-PARAMETRO NOT = LOW-VALUE
PERFORM ELABORA-SCHEDA THRU ELABORA-SCHEDA-EX.
ACQUISIZIONE-SCHEDA-EX. EXIT.
    
```

Il campo SW-STAMPANTI è un campo sottodefinito in 10 campi da 20 byte con cui si comunica a ISYRIPM0 la volontà di avere un separatore sulla corrispondente stampante. Un campo diverso da spazio farà effettuare la stampa del separatore. Sul campo deve essere posta una stringa di testo che identifichi la stampa. Tale stringa viene posta sul separatore di fianco al codice della stampante.

L'acquisizione dei parametri potrà seguire in genere due metodologie diverse.

La prima metodologia prevede una individuazione di tipo posizionale cioè legata al progressivo di lettura della scheda. In altre parole sulla prima scheda sono presenti certe informazioni, sulla seconda altre e così via.

La seconda metodologia utilizzabile prevede l'individuazione del parametro tramite una particolare costante presente sulla scheda stessa. In genere questa costante è la descrizione in chiaro del parametro fornito. Si ritiene che questa soluzione sia preferibile alla precedente in quanto più chiara e flessibile. Le costanti possono essere anche standardizzate all'interno di ogni progetto rendendo più facile anche l'inserimento dei parametri se questi dovessero essere forniti da scheda (non si dovrà controllare su quale scheda ed in quale posizione questi devono essere forniti).

Acquisiti i parametri si dovrà attivare la fase di controllo degli stessi. Questa deve essere effettuata attraverso il programma ISYBCCD0 (CCD-OPZIONE = CCD-KOPZ-INP).

Il modo più semplice per effettuare questo controllo è quello di effettuarlo immediatamente dopo l'acquisizione di una scheda parametro utilizzando l'opzione CCD-OPZERR = 'D'. In questo modo il programma ISYBIPM0 effettuerà la stampa della scheda parametro e ISYBCCD0, immediatamente al di sotto di questa, inserirà tutti i messaggi sugli errori riscontrati.

Esempio di acquisizione dati e controllo

A scopo esemplificativo vengono qui riportate le istruzioni relative all'acquisizione dei parametri, al loro controllo ed alla gestione degli errori.

```

*****
MOVE ZERO          TO CCD-TOTERR.          (1)
MOVE 'D'           TO CCD-OPZERR.          (2)
MOVE CCD-KOPI-CHE TO CCD-OPZIONE.          (3)
PERFORM ACQUISIZIONE-SCHEDA THRU
        ACQUISIZIONE-SCHEDA-EX
        UNTIL SCHEDA-PARAMETRO = LOW-VALUE. (4)
IF CCD-TOTERR > 0
GO TO FINE-PROGRAMMA.                      (5)
    
```

```

primo controllo logico (6)
IF errore
  MOVE campo-errato TO CCD-ESTERNO
  MOVE codice-errore TO CCD-ER-CODICE
  GO TO ERR-FINE-PROGRAMMA.

secondo controllo logico
IF errore
  MOVE campo-errato TO CCD-ESTERNO
  MOVE codice-errore TO CCD-ER-CODICE
  GO TO ERR-FINE-PROGRAMMA.

ERR-FINE-PROGRAMMA.
  MOVEER07 'TOCCD-FUNZIONE. (7)
  CALL 'ISYRCCDO' USING CCD-AREA.

FINE-PROGRAMMA.
  MOVE 'CLOSE ' TO CCD-FUNZIONE. (8)
  CALL 'ISYRCCDO' USING CCD-AREA.

.....
  STOP RUN. (9)

ACQUISIZIONE-SCHEDA.
  CALL 'ISYRIPMO' USING CA-AREA (10)
  SW-STAMPANTI
  SCHEDA-PARAMETRO.
  IF SCHEDA-PARAMETRO = LOW-VALUE (11)
  GO TO ACQUISIZIONE-SCHEDA-EX.

  MOVE CA-AREA-SISTEMA TO CCD-AREA-SISTEMA. (12)

  IF prima-scheda-parametro (13)
  PERFORM CONTROLLO-SCHEDA-1 THRU
  CONTROLLO-SCHEDA-1-EX
  GO TO ACQUISIZIONE-SCHEDA-EX.
  IF seconda-scheda-parametro
  PERFORM CONTROLLO-SCHEDA-1 THRU
  CONTROLLO-SCHEDA-1-EX
  GO TO ACQUISIZIONE-SCHEDA-EX.

.....
ACQUISIZIONE-SCHEDA-EX. EXIT.

CONTROLLO-SCHEDA-1.

  MOVE funzione-richiesta TO CCD-FUNZIONE. (14)
  MOVE lunghezza-campo TO CCD-INPLMT. (15)
  MOVE campo-da-controllare TO CCD-ESTERNO. (16)
  CALL 'ISYRCCDO' USING CCD-AREA campo-record. (17)

  MOVE funzione-richiesta TO CCD-FUNZIONE.
  MOVE lunghezza-campo TO CCD-INPLMT.
  MOVE campo-da-controllare TO CCD-ESTERNO.
  CALL 'ISYRCCDO' USING CCD-AREA campo-record.

.....

CONTROLLO-SCHEDA-1-EX. EXIT.

CONTROLLO-SCHEDA-2.

.....
CONTROLLO-SCHEDA-2-EX. EXIT.

```

L'istruzione (1) è l'azzeramento del campo su cui ISYBCCD0 totalizza gli errori trovati. Tale campo viene ricontrrollato al termine dell'acquisizione e dei controlli formali (punto 5) ed in caso di errori il programma viene terminato. L'istruzione al punto (2) richiede a ISYBCCD0 di effettuare il DISPLAY dei messaggi di errore evitandone quindi la gestione sull'applicativo. La riga di display comprende anche il dato controllato. La successiva istruzione indica a ISYBCCD0 che è richiesto il controllo dei dati forniti in CCD-ESTERNO e che il campo in cui dovrà essere ritornato il dato in formato interno è fornito come parametro. La routine di acquisizione viene richiamata secondo le modalità già illustrate (4). Controllata l'assenza di errori formali si passa ai controlli logici per i quali deve essere scritto l'apposito codice. In caso di errore viene mosso l'apposito codice nel campo CCD-ER-CODICE (6) e richiamata la funzione di gestione dei messaggi di errore di ISYBCCD0 (7). Anche in questo caso verrà effettuato il display automatico dell'errore.

Le istruzioni di FINE-PROGRAMMA devono comprendere l'attivazione di ISYBCCD0 con la funzione 'CLOSE' per ottenere la chiusura di eventuali archivi aperti gestiti al suo interno.

La routine di acquisizione delle schede parametro segue lo schema descritto. L'istruzione (11) termina le operazioni quando sono terminate le schede.

L'istruzione (12) deve essere inserita dopo la prima chiamata ad ISYBIPM0 in quanto questo programma provvede al riempimento dei campi di CA-AREA-SISTEMA.

Per ogni scheda, riconosciuta secondo una delle modalità indicate, è definita una apposita routine di controllo. Questa comprende un blocco di istruzioni per ognuno dei campi presenti su di essa. Il blocco consiste in: spostamento della funzione richiesta sul campo CCD-FUNZIONE (14), indicazione della lunghezza del campo da controllare presente sulla scheda parametro (15), passaggio del dato da controllare (16) ed attivazione di ISYBCCD0 (tramite routine ISYRCCD0) passando come parametri l'area di interfaccia ed il campo su cui deve essere posto il risultato.

In ambiente batch si consiglia, per maggiore chiarezza sui messaggi, di gestire applicativamente i campi obbligatori. Se invece il campo obbligatorio è l'unico della scheda parametro si può richiedere anche il controllo di obbligarietà con un'istruzione "MOVE 'O' TO CCD-OPZOB". In questo caso occorre ricordarsi di porre a SPACE lo stesso campo nei controlli non obbligatori.

Controllo e conversione dati (ISYxCCD)

Il programma ISYxCCD0 (ove x è "O" per la versione on-line e "B" per quella batch) concentra tutte le funzioni di controllo e conversione dati necessarie ai programmi applicativi.

Questa centralizzazione dei controlli permette di rendere omogenea questa funzione nei vari programmi scaricando il programmatore della necessità di sviluppare codice.

Ogni funzione è identificata da un codice avente il seguente formato:

fflloooo

dove:

- ff** è un codice identificativo della funzione stessa
- ll** è la lunghezza del campo interno (sul record) in byte
- oooo** sono eventuali ulteriori dati di specifico interesse per ogni funzione (es. numero di decimali per campi numerici, nome della tabella per controlli tabellari, ecc.).

Le funzioni previste vengono descritte dettagliatamente all'interno di questo capitolo.

CONTROLLO ED EDIT DEI DATI ON-LINE (ISYOCCD)

Il programma di controllo ed edit dei dati in on-line (ISYOCCD0) è la versione CICS del programma ISYBCCD0.

La modalità di attivazione prevista è tramite LINK in quanto ha una funzionalità simile a quella di una routine.

L'area di interfaccia (copy ISYCCCD) viene passata in DFHCOMMAREA. I campi di ingresso e quelli di uscita della funzione sono contenuti all'interno di questa interfaccia.

La modalità normale di chiamata è quella con CCD-OPZIONE = CCD-KOPZ-PAN che prevede un controllo da pannello, per cui, in caso di assenza di errori, restituisce sull'ingresso (campo CCD-ESTERNO) il dato in formato editato tale da poter essere di nuovo portato su pannello; nel caso di ingresso a LOW-VALUE (cioè dato non digitato) il programma provvede a riformare il pannello editando il dato dal campo interno. In questa modalità deve essere quindi fornito come dato di ingresso non solo il dato esterno (in CCD-ESTERNO), ma anche il dato in formato interno già acquisito o presente sul record (nell'apposito campo di interfaccia dedicato alla funzione). Per facilitare questa operazione sono state predisposte due routine attivabili tramite CALL, la prima di preparazione dell'input per ISYOCCD0 (ISYRPCD0) e la seconda di trattamento dei dati di uscita (ISYRTCD1). La routine ISYRPCD0 provvede unicamente a spostare il dato interno (fornito come parametro) nell'apposito campo di interfaccia, senza che il programmatore debba conoscere il tipo di funzione richiesta a ISYOCCD0 (la funzione viene normalmente richiesta con una variabile associata al campo). La routine ISYRTCD1 oltre a spostare il dato interno dal campo di interfaccia a quello di record provvede anche alla gestione della mappa, spostando su di essa il campo editato e resettando l'attributo al valore di default. Nel caso di presenza di errori questa stessa routine imposta l'attributo a BRIGHT MDTON ed il campo per il posizionamento del cursore con il valore della posizione del campo errato.

Questa modalità di gestione permette di effettuare delle SEND di pannello in ogni caso completo di fincato e dati, riconoscendo la mancanza di dati digitati (MAPFAIL).

Modalità di controllo dei campi

Qui di seguito vengono riportate, a scopo esemplificativo, le istruzioni necessarie per il controllo dei dati. Le istruzioni vengono inserite nella routine 215-CONTR-CAMPI che dovrà essere richiamata immediatamente dopo la RECEIVE del pannello (e la gestione dell'HELP).

215-CONTR-CAMPI.

MOVE SPACES TO CCD-OPSOBB. (1)
 MOVE EERO TO CCD-TOTERR. (2)
 MOVE 'D' TO CCD-OPERR. (3)

MOVE EIBTRMID TO TSERR-TERMINALE (4)

EXEC CICS IGNORE CONDITION QIDERR END-EXEC.
 EXEC CICS DELETEQ TS QUEUE (TSERR-KEY) END-EXEC. (6)

MOVE CA-AREA-SISTEMA TO CCD-AREA-SISTEMA. (7)

MOVE CCD-KOPE-PAN TO CCD-OPICOME. (8)

PERFORM 220-TRATTA-CAMPI THRU 220-TRATTA-CAMPI-EX. (9)

MOVE SPACE TO CCD-OPSOBB CCD-OPERR. (10)

IF CCD-TOTERR > EERO (11)

MOVE MSG-ERR-FORMALI TO WS-MSG-ERRORE
 GO TO 215-CONTR-CAMPI-EX.

primo controllo logico (12)

secondo controllo logico

215-CONTR-CAMPI-EX. EXIT.

220-TRATTA-CAMPI.

MOVE '0' TO CCD-OPSOBB. (13)

MOVE codice-funzione TO CCD-FUNZIONE. (14)

MOVE lung.h.campo-mappa TO CCD-INPLNT. (15)

MOVE campo-mappa TO CCD-ESTERNO. (16)

CALL 'ISYRCD0' USING CCD-AREA campo-interno. (17)

PERFORM LINK-ISYOCOD THRU LINK-ISYOCOD-EX. (18)

CALL 'ISYRCD1' USING CCD-AREA campo-int. (19)

campo-mappa
 MAP-nome-campo
 WS-MAP-CURSOR.

MOVE spaces TO CCD-OPSOBB. (20)

MOVE codice-funzione TO CCD-FUNZIONE.

MOVE lung.h.campo-mappa TO CCD-INPLNT.

MOVE campo-mappa TO CCD-ESTERNO.

CALL 'ISYRCD0' USING CCD-AREA campo-interno.

PERFORM LINK-ISYOCOD THRU LINK-ISYOCOD-EX.

CALL 'ISYRCD1' USING CCD-AREA campo-int.

campo-mappa
 MAP-nome-campo
 WS-MAP-CURSOR.

LINK-ISYOCOD.

EXEC CICS LINK PROGRAM (CA-ISYOCOD)
 COMAREA (CCD-AREA)
 LENGTH (CCD-AREA-LNT) END-EXEC.

LINK-ISYOCOD-EX. EXIT.

I controlli sui dati acquisiti da pannello vengono suddivisi in due blocchi: il primo è riservato al controllo formale (numericità, valori tabellari, ecc.) che vengono effettuati attraverso il programma ISYOCCD0 ed il secondo ai controlli logici (verifica di esistenza su un archivio, controlli incrociati fra due campi, ecc.) per i quali deve essere sviluppato il necessario codice.

I controlli vengono preceduti da un gruppo di istruzioni di preparazione (1-8). L'istruzione (1) provvede a pulire l'indicatore di campo obbligatorio. L'istruzione (2) azzerà il contatore degli errori che verrà utilizzato al termine dei controlli per rilevare la presenza di errori. L'istruzione (3) richiede la scrittura degli eventuali messaggi di errore nell'apposita coda di temporary storage che viene resettata nelle istruzioni di cui al punto (6). Infine l'istruzione (7) prepara la parte di sistema dell'area di interfaccia con il programma di controllo e l'istruzione (8) richiede l'effettuazione di un controllo da pannello. Queste istruzioni vengono poste in testa per evitare di ripeterle per ognuno dei controlli.

Per quello che riguarda il controllo formale viene mostrato a scopo esemplificativo il controllo di due campi, il primo obbligatorio ed il secondo non obbligatorio facendo notare che la differenza è solo nelle istruzioni (13) e (20) cioè nell'accensione dell'indicatore di campo obbligatorio. Le istruzioni (14)-(16) consistono nel riempimento degli opportuni campi di interfaccia rispettivamente con la funzione richiesta, la lunghezza del campo di mappa ed il valore presente sul campo di mappa. Quest'ultimo sarà LOW-VALUE nel caso in cui il campo non è stato digitato. In queste condizioni il programma ISYOCCD0 provvederà a riformare il pannello riutilizzando il valore precedentemente fornito che viene portato nell'opportuno campo dell'interfaccia dalla routine ISYRPCD0 al punto (17). Il programma ISYOCCD0 viene attivato con una istruzione di LINK richiamata con una PERFORM (18). La routine ISYRTCD1 richiamata nell'istruzione (19) provvede al trattamento dei risultati del controllo. Più esattamente: sposta il dato in formato interno dall'apposito campo dell'interfaccia sul campo interno (normalmente sul record), sposta il campo editato sulla mappa; in caso di errore pone l'attributo del campo al valore DFHUNIMD (luminoso MDT-ON) e nel campo WS-MAP-CURSORE il progressivo di posizione del campo sul pannello mentre in caso contrario l'attributo è posto al valore di default. Sia il valore di default che la posizione del campo sono prelevati dall'area delle informazioni sul campo (WS-nome-campo), già descritta in precedenza, che deve essere fornita in ingresso alla routine. I blocchi di questo tipo devono essere ripetuti per ogni campo presente su pannello.

Una modalità alternativa più efficiente di scrivere la routine 220-TRATTA-CAMPI prevede di effettuare la PERFORM (18), e quindi la LINK CICS a ISYOCCD0, una sola volta, accelerando l'esecuzione del programma. In questo caso tutti i controlli che devono essere effettuati vengono predisposti (13a)-(17a) in una coda di temporary storage ogni item della quale contiene una copia della CCD-AREA preparata per il controllo di un campo. Viene chiamato ISYOCCD0 (18a), che, riconoscendo questa modalità di attivazione, legge gli items di TS e predispose su ciascuno il risultato del controllo. Successivamente viene riletta la coda di TS e, per ogni item, utilizzati i risultati del controllo effettuato (19a).

```

220-TRATTA-CAMPI. (alternativa)

MOVE KIBTRMID          TO CCD-CODA-ITEM.
MOVE CA-LIVELLO        TO CCD-CODA-LIVELLO.
MOVE CCD-AREA-LMT      TO CCD-CODA-LMT.

MOVE '0'               TO CCD-OPICOB.          (13a)
MOVE codice-funzione  TO CCD-FUNZIONE.        (14a)
MOVE lungh.campo-mappa TO CCD-IMPINT.         (15a)
MOVE campo-mappa      TO CCD-ESTERNO.         (16a)
CALL 'ISYRPCD0' USING CCD-AREA campo-interno. (17a)
MOVE x                 TO CCD-CODA-ITEM.
PERFORM 301-SCRIVI-CODA-CCD THRU
          301-SCRIVI-CODA-CCD-EX.              (18a)

.....

MOVE spaces           TO CCD-OPICOB.          (20a)
MOVE codice-funzione  TO CCD-FUNZIONE.
MOVE lungh.campo-mappa TO CCD-IMPINT.
MOVE campo-mappa      TO CCD-ESTERNO.
CALL 'ISYRPCD0' USING CCD-AREA campo-interno.
MOVE x                 TO CCD-CODA-ITEM.
PERFORM 301-SCRIVI-CODA-CCD THRU
          301-SCRIVI-CODA-CCD-EX.              (18a)

PERFORM 303-LINK-ISYOCOD THRU
          303-LINK-ISYOCOD.                    (18a)

MOVE x                 TO CCD-CODA-ITEM.
PERFORM 302-LEGGI-CODA-CCD THRU
          302-LEGGI-CODA-CCD-EX.              (18a)
CALL 'ISYRPCD1' USING CCD-AREA campo-int.     (19a)
                               campo-mappa
                               MAP-nome-campo
                               WS-MAP-CURSOR.

.....

MOVE x                 TO CCD-CODA-ITEM.
PERFORM 302-LEGGI-CODA-CCD THRU
          302-LEGGI-CODA-CCD-EX.              (18a)
CALL 'ISYRPCD1' USING CCD-AREA campo-int.     (19a)
                               campo-mappa
                               MAP-nome-campo
                               WS-MAP-CURSOR.

301-SCRIVI-CODA.
EXEC CICS WRITEQ TS QUEUE (CCD-CODA-KEY)
          FROM (CCD-AREA)
          LENGTH (CCD-CODA-LMT)
          ITEM (CCD-CODA-ITEM)
          END-EXEC.

301-SCRIVI-CODA-EX.
EXIT.

302-LEGGI-CODA.
EXEC CICS READ TS QUEUE (CCD-CODA-KEY)
          INTO (CCD-AREA)
          LENGTH (CCD-CODA-LMT)
          ITEM (CCD-CODA-ITEM)
          NUMITEMS (CCD-CODA-NUMITEMS)
          END-EXEC.

302-LEGGI-CODA-EX.
EXIT.

```



```

303-LINK-ISOCCD.
EXEC CICS LINK PROGRAM (CA-ISOCCD)
      COMAREA (CA-AREA)
      LENGTH (CA-AREA-SYSLFT) END-EXEC.

303-LINK-ISOCCD-EX.
EXIT.
    
```

Dopo l'ultimo controllo, deve essere posta l'istruzione di cui al punto (11) che verifica la presenza di almeno un errore nei controlli precedenti. In questo caso viene inserito nell'apposito campo il codice "ISY020C" corrispondente al messaggio di errori formali. L'istruzione di pulizia dei campi CCD-OPZERR e CCD-OPZOB (10) serve ad evitare che successive chiamate al programma di controllo trovino questi indicatori già impostati e ad esempio non venga scritto il messaggio di errore generico nella coda di temporary storage.

Se i controlli formali non hanno segnalato errori si passa ai controlli logici. Questi saranno specifici per i vari campi e, nel caso di errori ognuna di queste dovrà provvedere alla gestione del messaggio specifico, ponendone il codice nel campo WS-MSG-ERRORE e ponendo la posizione del campo errato (prelevata sempre dall'area delle caratteristiche) in WS-MAP-CURSOR.

L'ultima routine illustrata (SEND-MAPPA) contiene anche le istruzioni di trattamento del messaggio di errore. Le istruzioni ai punti (25)-(29) sono simili a quelle già illustrate per i controlli dei campi. L'istruzione (30) sposta il messaggio sul campo di mappa ad esso riservato.

La SEND del pannello viene effettuata con le opzioni ERASE in quanto il pannello viene completamente riformato (eliminando l'eventuale presenza di pannelli di HELP), e CURSOR (xx) che richiede di porre il cursore nella posizione di pannello indicata nel campo xx.

```

250-SEND-MAPPA.

IF WS-MSG-ERRORE NOT = SPACES
MOVE WS-MSG-ERRORE TO CCD-ER-CODICE
MOVE CA-AREA-SISTEMA TO CCD-AREA-SISTEMA
MOVE 'M' TO CCD-OPERK (25)
MOVE 'KPO7' TO CCD-FUNZIONE (26)
MOVE CCD-KOPE-PAN TO CCD-OPZIONE (27)
MOVE 07 TO CCD-IMPINT (28)
PERFORM LINK-ISOCCD THRU LINK-ISOCCD-EX (29)
MOVE CCD-ER-MESS-FULL TO campo-errore-in-mappa
ELSE
MOVE SPACE TO campo-errore-in-mappa. (30)

EXEC CICS SEND MAPSET (.....)
      MAP (.....)
      CURSOR (WS-MAP-CURSOR)
      ERASE FREEKBS FRSET END-EXEC.

250-SEND-MAPPA-EX. EXIT.
    
```

L'HELP di campo (ISYOHCD)

In modalità on-line il programma ISYOCCD0 gestisce l'help sui campi acquisiti da pannello. L'help consiste in un testo di spiegazione del controllo formale effettuato sul campo stesso.

Se sul campo è previsto un controllo di tipo tabellare vengono visualizzati tutti i valori accettabili sul campo stesso. In questo caso è anche possibile che l'utente posizioni il cursore su una delle righe premendo un tasto funzionale specifico. Il valore corrispondente verrà in questo caso posto sul campo su cui è stato richiesto l'help ed acquisito.

L'help viene richiesto dall'operatore con un carattere "?" digitato sul campo, seguito da spazi (o da un CANCEL FC). Il programma ISYOCCD0, richiamato in modalità CCD-KOPZ-PAN e secondo le modalità descritte, al momento in cui viene richiesto il controllo del campo stesso, riconosce questa particolare sequenza di caratteri e cede il controllo (XCTL) al programma ISYOHCD0 che gestisce l'HELP di campo. Questo provvede a caricare un opportuno testo di help in una coda di TS associata al terminale (\$HCD). Nel caso in cui sia richiesto un controllo tabellare viene richiesto a ISYOCCD0 (chiamato secondo la solita modalità di LINK) l'elenco dei codici accettabili (CCD-OPZIONE = CCD-KOPZ-HLP). ISYOCCD0 aggiunge questi valori alla stessa coda di TS per cui, al ritorno al chiamante, ISYOHCD0 potrà gestire la visualizzazione di questa coda su cui è presente l'intero testo di help.

Al termine della visualizzazione viene riattivato il programma ISYOCCD0 in modalità XCTL con gli stessi dati iniziali ed il campo CCD-ESTERNO a LOW-VALUE. In questo modo viene riformato il campo su pannello a partire dall'eventuale dato già acquisito ed al ritorno al programma applicativo si saranno ripristinate le condizioni antecedenti alla richiesta di HELP.

Per i campi tabellari in ISYOHCD0 è prevista la funzionalità aggiuntiva di selezione di uno dei valori previsti tramite posizionamento del cursore e pressione di un tasto funzionale. L'unica differenza rispetto al flusso indicato è che in questo caso prima della riattivazione di ISYOCCD0 invece di un reset del campo CCD-ESTERNO su di esso viene spostato il valore selezionato per cui questo viene acquisito come se fosse stato digitato da pannello.

Il flusso completo viene schematizzato in figura 2.

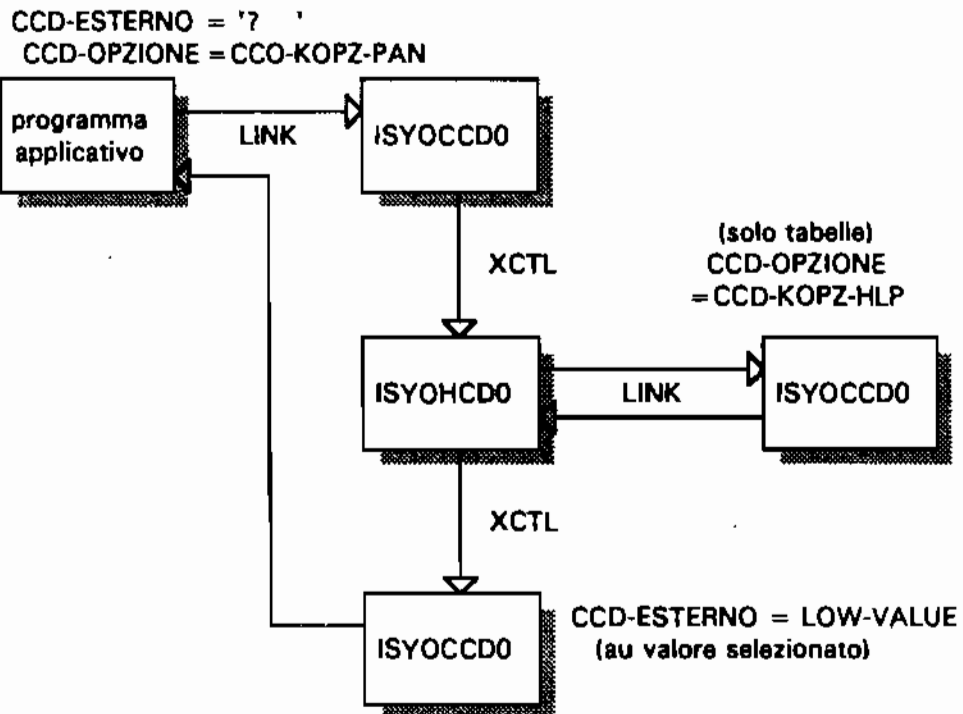


Figura 2 - Flusso dell'help di campo

I messaggi di errore (ISYOHER)

In ogni pannello viene riservata ai messaggi di errore una sola riga. Nel caso in cui siano presenti errori su più campi in questa riga potrà essere fornito un messaggio generico indicante la presenza di errori sui campi che sul pannello sono ad alta luminosità (messaggio "ISY020C CAMPI IN RISALTO ERRATI OD OMESSI"). Questa segnalazione può essere comunque sufficiente ad individuare gli errori e ad effettuare le correzioni.

Il programma ISYOHER0 permette invece di visualizzare il messaggio di errore specifico per ogni campo. Questi messaggi vengono gestiti automaticamente all'interno del programma ISYOCCD0 per quello che riguarda i controlli formali e sono a carico dell'applicativo per gli altri controlli.

La modalità di funzionamento del programma ISYOHER0 è molto semplice. Esso gestisce la visualizzazione della coda di Temporary Storage avente come identificatore il codice del terminale e la costante "\$ERR", con possibilità di scroll avanti ed indietro quando i messaggi sono superiori a quelli contenuti in una videata.

La scrittura degli item all'interno della coda è effettuata all'interno di ISYOCCD0 quando il campo CCD-OPZERR viene posto al valore 'D'. La scrittura è automatica sia per gli errori rilevati in fase di controllo che per quelli gestiti dall'applicativo con la funzione "ER07".

Rimane comunque a carico dell'applicativo la cancellazione della coda prima di un nuovo ciclo di controlli. Per le istruzioni necessarie si può far riferimento al punto (6) dell'esempio di controllo dei campi. Si noti inoltre che l'attivazione del programma ISYOHER0 viene effettuata prima del ciclo di controlli in quanto l'HELP deve essere sempre relativo ad i messaggi di errore dell'ultimo pannello inviato, indipendentemente dalle ulteriori digitazioni fatte successivamente su di esso.

La gestione dei campi obbligatori

Un'ulteriore possibilità offerta dal programma ISYOCCD è quella del controllo dei dati obbligatori forniti. Questo può essere richiesto ponendo al valore "O" il campo CCD-OPZOBB. Nel caso in cui venga riscontrata la mancanza del dato (spazio o low-value per campi alfabetici, zero per campi numerici) viene segnalato un errore.

CONTROLLO ED EDIT DEI DATI IN BATCH (ISYBCCD)

Il programma di controllo e conversione dati ISYBCCD0 presenta le stesse funzionalità previste per l'on-line. Anche la sua modalità di attivazione richiama le funzionalità on-line in quanto viene caricato dinamicamente in memoria al momento della prima richiesta. Questa modalità permette quindi, in caso di modifica di ISYBCCD0, di far lavorare automaticamente tutti i programmi con la nuova versione senza dover procedere ad una nuova fase di link-edit.

La richiesta di servizi ad ISYBCCD0 viene effettuata tramite la routine ISYRCCD0 che ne richiede il caricamento in memoria cedendogli il controllo. Viene cioè effettuato un link-edit dinamico di ISYBCCD0 al programma chiamante.

L'interfaccia di comunicazione fra il programma applicativo ed ISYBCCD0 è contenuta nel copy ISYCCCD e viene trasmessa in Linkage Section secondo le modalità normali per sottoprogrammi richiamati con CALL.

Per semplificare il riempimento dell'interfaccia è possibile sfruttare la possibilità offerta nel solo ambiente batch di poter passare i dati in formato interno direttamente fornendo il campo come parametro della CALL. Per permettere al programma ISYBCCD0 di distinguere questa modalità di passaggio dati rispetto al riempimento completo dell'interfaccia sono previste le due opzioni CCD-KOPZ-CHB e CCD-KOPZ-EDB rispettivamente per richiedere il controllo di un dato o il suo edit.

Le modalità di attivazione sono descritte dettagliatamente nel capitolo riservato alla descrizione di ISYxCCD0.

GESTIONE DEGLI ERRORI

Il programma ISYxCCD0 contiene anche una gestione di errori e di messaggi di errore.

Quando, in seguito ad una richiesta di controllo, il programma rileva un errore nei dati forniti, segnala questa situazione con un codice di errore nel campo CCD-ERRORE (SPACE in questo campo significa nessun errore rilevato).

Il codice di errore ha il formato standard dei codici di errore. In particolare è del tipo:

xxxpppt

dove *xxx* è il codice del progetto, *ppp* è un progressivo e *t* è la gravità dell'errore, e cioè:

- D errore di sistema
- C errore bloccante
- W o I messaggio informativo.

Si ricorda che l'errore è comunque identificato univocamente dal codice di progetto e dal progressivo.

Il campo CCD-ERRORE viene reinizializzato ad ogni chiamata del programma per cui contiene sempre l'ultimo (eventuale) codice di errore.

Ad ogni errore viene contemporaneamente gestito il contatore CCD-TOTERR. Questo viene incrementato di 1 nel caso in cui il programma riscontra un errore. L'azzeramento di questo campo è a carico del programma chiamante. Il suo uso tipico è quello di controllare la presenza di un errore in uno qualsiasi di una sequenza di controlli. Supponiamo ad esempio che si debba controllare un record. Verrà richiesto il controllo per ognuno dei campi presenti sul record.

Se prima dei controlli viene azzerato il campo CCD-TOTERR, può essere controllata la presenza di un errore in uno qualsiasi dei campi semplicemente controllando se il valore di CCD-TOTERR è maggiore di zero.

Il programma ISYxCCD0 permette inoltre di ottenere una gestione automatica dei messaggi di errore.

Se al momento della richiesta di una funzione a ISYxCCD0 viene impostato lo speciale indicatore CCD-OPZERR al valore "M", in caso di errori riscontrati durante il controllo, il programma provvede a fornire automaticamente, oltre al codice in CCD-ERRORE, anche il relativo messaggio nel campo CCD-MESERR.

Esiste inoltre l'opzione "D" che ha funzionamento diverso tra batch ed on-line. In modalità batch, indicando questo valore nel campo CCD-OPZERR, oltre a quanto già descritto, il programma provvede ad effettuare un'istruzione di DISPLAY di una riga contenente il codice di errore, il messaggio associato, il campo presente in CCD-ESTERNO. In modalità on-line il programma effettua invece la scrittura di un record in una coda di temporary storage (nome della coda tttt\$ERR dove tttt è il codice del terminale). Questa opzione, associata all'uso del contatore CCD-TOTERR permette di ottenere una semplice gestione dei messaggi di errore durante il controllo di dati.

Nel caso di errori di sistema (tipo "D") ed in ambiente batch oltre al display sulla stampante di sistema viene anche inviato un messaggio a console.

Nel caso in cui si debbano gestire ulteriori errori all'interno dei programmi applicativi (i cui codici devono sempre seguire la regola indicata) esiste una particolare funzione che permette, fornendo il codice di errore, di ottenere il relativo messaggio.

I codici di errore ed i messaggi associati sono contenuti nelle tabelle xxx999 (dove xxx è il codice del progetto). Tutti tali codici devono iniziare con la sigla xxx del progetto. Se questi codici non vengono trovati, da ISYOCCD0, nella tabella xxx999 di progetto, vengono cercati nella tabella ISY999 (per compatibilità con la vecchia gestione che prevedeva i messaggi di errore di tutti i progetti nella tabella ISY999).

I messaggi possono contenere anche dati variabili. Questi sono identificati dai caratteri "\$" nel messaggio stesso. Al momento dell'uso del messaggio questi vengono sostituiti con i valori correnti.

INTERFACCIA PER LE FUNZIONI DI ISYXCCD0

L'interfaccia con il programma ISYxCCD0 sono i campi contenuti nella copy ISYCCCD.

In particolare esistono campi di interfaccia che sono comuni per tutte le funzioni e campi specifici di ogni funzione.

I campi comuni sono i seguenti:

CCD-FUNZIONE	è il campo in cui deve essere spostato il codice identificativo della funzione richiesta
CCD-ESTERNO	è il campo in cui deve essere spostato il valore da controllare o su cui viene restituito il formato editato
CCD-OPZIONE	è la modalità in cui il programma deve funzionare; sono previste le seguenti modalità: <ul style="list-style-type: none"> • Controllo dei dati presenti in CCD-ESTERNO • Editing dei dati forniti nel campo interno specifico della funzione sul campo CCD-ESTERNO • Controllo dei dati presenti in CCD-ESTERNO e successivo editing nello stesso campo
CCD-INPLNT	deve contenere la lunghezza del campo esterno da controllare o su cui deve essere effettuato l'edit
CCD-ERRORE	contiene un codice di errore quando il controllo dei dati risulta non positivo
CCD-MESERR	contiene il messaggio di errore quando questo è richiesto
CCD-OPZOBB	indicatore di campo obbligatorio (se contiene il valore "O")
CCD-OPZERR	indicatore per richiedere il trattamento del messaggio di errore (se impostato a "M" o "D")
CCD-TOTERR	totalizzatore del numero di errori

Dei campi relativi alla gestione degli errori si è già diffusamente parlato in precedenza.

Gli altri campi di interfaccia sono di specifico interesse per ognuna delle funzioni disponibili sul programma ISYxCCD0. Questi campi, che comprendono tra l'altro il campo in formato interno, vengono mostrati qui di seguito con l'illustrazione della funzione.

Viene inoltre indicato il formato interno, cioè la PICTURE che deve essere utilizzata su record, per ognuna delle funzioni.

Queste informazioni sono riportate schematicamente nella figura 3 alla fine del paragrafo.

Funzione di acquisizione per campo alfanumerico

Questa funzione viene richiesta con il codice "ALn" dove "n" 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 di interfaccia sono i seguenti:

CCD-AL-INTERNO campo acquisito

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

Funzioni di gestione per campi numerici

Queste 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 e vengono trattati i decimali.

Il numero può essere scritto in qualunque punto del campo di ingresso e con qualsiasi numero di decimali (purchè 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.

In particolare le funzioni sono le seguenti:

NZlld Acquisizione/Controllo di un campo numerico con trasformazione in formato ZONED

NPlld Acquisizione/Controllo di un campo numerico con trasformazione in formato PACKED

NBlld Acquisizione/Controllo di un campo numerico con trasformazione in formato BINARIO

SZlld Come per il controllo NZ con in più la gestione del segno

SPlld Come per il controllo NP con in più la gestione del segno

SBlld Come per il controllo NB con in più la gestione del segno

In tutti "l" rappresenta la lunghezza del campo in formato interno, cioè il numero dei byte occupati da questo, e "d" il numero dei decimali. "l" 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.

I campi di interfaccia sono i seguenti:

CCD-NZd numero in formato interno per NZ o SZ.

CCD-NPd numero in formato interno per NP o SP.

CCD-NBd numero in formato interno per NB o SB.

In tutti "d" è il numero dei decimali in quanto i campi sono definiti con la virgola virtuale nel punto corrispondente.

Oltre ad i campi indicati esistono ancora i seguenti campi di interfaccia che permettono di trattare campi numerici binari rispettivamente di un byte o di 3 byte.

Per campi di un byte:

CCD-NBX1-C Campo numerico.

Per campi di tre byte:

CCD-NBX3-C Campo numerico.

I campi ai quali è associato questo controllo devono avere PICTURE dipendente dal controllo specifico. Per controlli di tipo zoned SZ e NZ il formato sarà S9(l-d)V9(d). Per i formati packed SP e NP S9(2*l-1-d)V9(d) COMP-3. Per i formati binari NB02, SB02 S9(i)V9(d) COMP dove i+d vale 4. Per i formati binari NB04, SB04 il formato è S9(i)V9(d) COMP dove i+d vale 9. Per i formati NB01, SB01, NB03, SB03 il formato record dovrà essere X(l).

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 GGMAA
- DT05** Acquisizione/Controllo e Visualizzazione di una data con conversione in formato PACKED GGMAAAA
- DT06** Acquisizione/Controllo e Visualizzazione di una data con conversione in formato ZONED GGMAA
- DT08** Acquisizione/Controllo e Visualizzazione di una data con conversione in formato ZONED GGMAAAA
- 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

I campi di interfaccia sono i seguenti:

CCD-xxxx data in formato interno

CCD-DT-GIORNO giorno della settimana

Il simbolo **xxxx** rappresenta il formato di data richiesto.

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**.

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**, in caso di funzione di controllo (**CCD-KOPZ-CHB** o **CCD-KOPZ-INP** o **CCD-KOPZ-PAN**), viene trascurato qualsiasi dato presente nel campo **CCD-ESTERNO** e viene utilizzata la data di sistema, portandola nel campo interno nel formato previsto. Nel caso di edit (**CCD-KOPZ-EDT** o **CCD-KOPZ-EDB**) questa opzione non ha alcun effetto e viene editato il contenuto del campo interno.

Funzioni di gestione ore

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 nonsignificativi, 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

L'unico campo di interfaccia è:

CCD-xxxx time in formato interno

Il simbolo xxxx rappresenta il formato di ora richiesto.

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, in caso di funzione di controllo (CCD-KOPZ-CHB o CCD-KOPZ-INP o CCD-KOPZ-PAN), viene trascurato qualsiasi dato presente nel campo CCD-ESTERNO e viene utilizzato il time di sistema, portandolo nel campo interno nel formato previsto. Nel caso di edit (CCD-KOPZ-EDT o CCD-KOPZ-EDB) questa opzione non ha alcun effetto e viene editato il contenuto del campo interno.

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 TBlxxxxxx dove l è la lunghezza del campo e xxxxxx è la sigla identificativa della tabella di valori accettabili.

I campi di interfaccia sono i seguenti:

CCD-TB-INTERNO valore in formato interno

CCD-TB-DESCRIZ descrizione corrispondente al valore

CCD-TB-CAMPI altre informazioni associate al valore

Il contenuto del campo CCD-TB-CAMPI non è controllato dal programma di gestione delle tabelle. L'uso di questo è quindi a completo carico dell'applicativo.

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

Su questa funzione, in ambiente on-line, è possibile richiedere lo scarico di tutti i valori presenti in tabella. Questa modalità di funzionamento si ottiene con il valore 'H' (CCD-KOPZ-HLP) nel campo CCD-OPZIONE. La tabella viene scaricata in una coda di T.S. con identificatore terminale + "SHCD". Il primo item è la testata della tabella. Questa modalità viene utilizzata dall'help di campo.

Abilitazione di valori all'utente

Per i campi su cui è previsto un controllo di tipo tabellare è possibile sfruttare l'ulteriore possibilità di abilitare i singoli valori agli utenti. È cioè possibile indicare al sistema che determinati valori possono essere digitati solo da alcuni utenti e non da altri.

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 di interfaccia sono i seguenti:

- CCD-BT-INTERNO campo in formato interno (byte)
- CCD-BT-ESPANSO campo espanso su 8 byte
- CCD-BT-ESPANSO-x sottodefinitore del campo espanso (x può valere da 7 a 0 secondo il byte)
- CCD-BT-EDITATO campo editato in formato x,x,x...
- CCD-BT-CAMPI simboli corrispondenti ai bit

In caso di attivazione per controllo, il campo in formato esterno sarà di 8 byte e dovrà essere mosso nel campo CCD-BT-ESPANSO. La routine effettuerà la compressione del campo accendendo nel byte CCD-BT-INTERNO i bit corrispondenti ai byte diversi da spazio del campo di ingresso. I byte di CCD-BT-ESPANSO rappresentano i singoli bit in ordine da quelli di peso maggiore (7) a quello di peso minore (0). Nel caso di attivazione per editing il campo da editare dovrà essere mosso in CCD-BT-INTERNO. La routine espanderà gli 8 bit negli 8 byte corrispondenti. In corrispondenza ad un bit acceso verrà mosso nel byte corrispondente il carattere associato al bit nella tabella di conversione.

Viene inoltre preparato il campo editato che conterrà i soli caratteri corrispondenti ad i bit accesi separati da virgole. Il campo editato viene spostato in CCD-BT-EDITATO e CCD-ESTERNO.

Nel campo CCD-BT-CAMPI vengono forniti i simboli corrispondenti ai vari bit. Il primo carattere è la lunghezza di ogni simbolo. Dal secondo carattere in poi vengono forniti i simboli associati ai singoli bit in ordine decrescente di peso del bit (da 7 a 0). Nel caso in cui non viene trovata la tabella di associazione dei caratteri ai bit, vengono utilizzati i

valori di peso, per cui se è acceso il bit di peso più elevato viene utilizzato il valore 7, 6 per il successivo e così via fino al carattere 0 per il bit di peso più basso.

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

Funzione di gestione del messaggio di errore

Questa funzione permette di ottenere il messaggio associato ad un codice di errore. Questo messaggio è contenuto nella tabella ISY999.

La funzione viene richiesta con il formato ER07. Il codice di errore deve essere fornito nel campo CCD-ER-CODICE. Il campo CCD-ESTERNO può essere posto a SPACE o può contenere la stringa di caratteri su cui l'errore è avvenuto.

In modalità batch e se è il campo CCD-OPZERR è stato messo al valore "D" viene infatti effettuato un DISPLAY dell'errore con, a lato, il contenuto del campo CCD-ESTERNO. In modalità on-line e nelle stesse condizioni il messaggio viene scritto nella coda di Temporary Storage "\$ERR" per la successiva gestione con il programma ISY0HER0.

Il messaggio di errore viene restituito nel campo CCD-ER- MESS.

I messaggi di errore possono contenere anche dei dati variabili. Questi vengono identificati dai caratteri "_" presenti sul messaggio stesso. I dati variabili vengono sostituiti con i valori forniti nel campo di interfaccia CCD-ER-VARIABLE. La sostituzione avviene da sinistra verso destra carattere per carattere. Il primo carattere "_" del messaggio viene cioè sostituito dal primo carattere del campo CCD-ER-VARIABLE, il secondo dal secondo e così via.

È inoltre definito il campo CCD-ER-MESS-FULL. Questo è composto dal codice di errore e, separato da uno spazio, il messaggio di errore. Questo campo può essere quindi utilizzato direttamente per visualizzare l'errore.

In questa funzione il contenuto del campo CCD-OPZIONE non ha alcun effetto.

TABELLA RIASSUNTIVA FUNZIONI ISY-CCD0		
funzione	PICTURE campo record	campi interfaccia (CCD-...)
AL11	X(11)	-AL-INTERNO
NZ11 (d)	9 (11-d) V9 (d)	-NRd
NP11 (d)	9 (2*11-1-d) V9 (d) COMP-3	-NPd
SP11 (d)	89 (2*11-1-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)	89 (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)	89 (9-d) V9 (d)	-NBd
DT02 (SYSTEM)	XX	-DT02 -DTGIORNO -DTMESH
DT04 (SYSTEM)	9 (6) COMP-3 qgmmaa	-DT04 -DTGIORNO -DTMESH
DT05 (SYSTEM)	9 (8) COMP-3 qgmmaaaa	-DT05 -DTGIORNO -DTMESH
DT06 (SYSTEM)	9 (6) qgmmaa	-DT06 -DTGIORNO -DTMESH
DT08 (SYSTEM)	9 (8) qgmmaa	-DT08 -DTGIORNO -DTMESH
DI04 (SYSTEM)	9 (6) COMP-3 aammyy	-DI04 -DTGIORNO -DTMESH
DI05 (SYSTEM)	9 (8) COMP-3 aaaaamyy	-DI05 -DTGIORNO -DTMESH
DI06 (SYSTEM)	9 (6) aammyy	-DI06 -DTGIORNO -DTMESH
DI08 (SYSTEM)	9 (8) aaaaamyy	-DI08 -DTGIORNO -DTMESH
TM04 (SYSTEM)	9 (6) COMP-3 HHMMSS	-TM04
TM06 (SYSTEM)	9 (6) HHMMSS	-TM06
TB11:cccccc	X(11)	-TB-INTERNO -TB-DESCRIZ -TB-CAMPI
BT11:cccccc	X	-BT-INTERNO -BT-ESPANSO -BT-EDITATO -BT-CAMPI
ER07	non previsto	-ER-CODICE -ER-VARIABILE -ER-MESS -ER-MESS-FULL

Figura 3. Tabella riassuntivi dei tipi di controllo

MODALITÀ DI ATTIVAZIONE ON-LINE

Il programma ISYOCCD0 viene attivato con una istruzione di LINK dal programma chiamante. I campi di interfaccia sono contenuti nella copy ISYCCCD.

Il programmatore dovrà quindi provvedere a riempire opportunamente questa interfaccia a seconda delle modalità di attivazione sottoriportate.

Controllo dati da pannello

La modalità tipica di chiamata nel caso in cui sia richiesto il controllo di dati da pannello è il seguente:

```

MOVE CA-AREA-SISTEMA TO CCD-AREA-SISTEMA (1)
MOVE funzione-richiesta TO CCD-FUNZIONE (2)
MOVE CCD-ROPE-PAN TO CCD-OPZIONE (3)
MOVE campo-lunghezza TO CCD-INPLNT (4)
MOVE campo-mappa TO CCD-ESTERNO (5)
MOVE campo-record TO campo-uscita-funzione (6)
EXEC CICS LINK PROGRAM (CA-ISYOCCD) (7)
        COMMAREA (CCD-AREA) LENGTH (CCD-AREA-LNT)
END-EXEC
IF CCD-ERRORE = SPACES
    MOVE CCD-ESTERNO TO campo-mappa (8)
    MOVE campo-uscita-funz TO campo-record (9)
ELSE
    MOVE -1 TO campo-mappa-lunghezza (10)
    MOVE CCD-ERRORE TO campo-errore-mappa (11)
    MOVE CCD-MSGERR TO campo-messaggio-mappa (12)
    
```

Si ricorda che in questa modalità il campo in CCD-ESTERNO viene controllato secondo le regole previste dalla funzione e trasformato nel formato interno nel campo opportuno. Se non sono stati trovati errori il dato viene editato nel formato previsto in CCD-ESTERNO.

La linea (1) passa al programma di controllo la parte di Communication Area di sistema contenente tra l'altro il nome del programma che richiede il servizio ed il nome dell'utente.

Nella linea (2) il campo indicato come "funzione-richiesta" è il campo associato al campo del record contenente il tipo di controllo da effettuare.

La linea (3) richiede di effettuare il controllo ed il successivo editing del campo.

Nella linea (4) il campo "campo-lunghezza" è il campo di mappa con suffisso L che contiene la lunghezza del campo digitato.

La linea (5) è lo spostamento del campo di mappa nell'opportuno campo di interfaccia.

La linea (6) inserisce nell'interfaccia il valore già acquisito in precedenza. ISYOCCD0, in caso di valore di ingresso pari a low-value, provvede infatti a rigenerare il pannello editando automaticamente il dato presente sul campo interno.

Al punto (7) è presente la chiamata al programma di controllo. Si noti che la chiamata viene effettuata in forma simbolica utilizzando il campo di communication area che contiene il nome effettivo del programma. L'attivazione è effettuata con LINK.

Al ritorno dal controllo viene verificata l'esistenza di un errore. Nel caso in cui non esistano errori viene riportato sulla mappa il campo editato (8) ed il campo di uscita sul record (9). Il campo di uscita è diverso per ogni tipo di controllo. Per il relativo nome si può far riferimento alla descrizione specifica di ogni singola funzione.

Le righe (10), (11) e (12) trattano il caso di errore permettendo di gestire il cursore ed i messaggi di errore. Si ipotizza che la SEND del pannello venga effettuata con l'opzione CURSOR che pone il cursore sul primo campo avente campo lunghezza uguale a -1.

Parte delle funzioni indicate possono essere effettuate automaticamente tramite le due routine ISYRPCD0 e ISYRTCD0. In questo caso il blocco di istruzioni diventa:

```

MOVE CA-AREA-SISTEMA      TO CCD-AREA-SISTEMA
MOVE funzione-richiesta  TO CCD-FUNZIONE.
MOVE CCD-KOPE-PAN        TO CCD-OPZIONE.
MOVE campo-lunghezza     TO CCD-INDINT.
MOVE campo-mappa         TO CCD-ESTERNO.
CALL 'ISYRPCD0' USING CCD-AREA campo-record.
EXEC CICS LINK PROGRAM (CA-ISYOCDD)
          COMMAREA (CCD-AREA) LENGTH(CCD-AREA-LMT)
END-EXEC.
CALL 'ISYTPCD0' USING CCD-AREA campo-record campo-mappa.
    
```

Solo controllo on-line

La modalità tipica di chiamata nel caso in cui sia richiesto il solo controllo dei dati sarà la seguente:

```

MOVE CA-AREA-SISTEMA      TO CCD-AREA-SISTEMA.
MOVE funzione-richiesta  TO CCD-FUNZIONE.
MOVE CCD-KOPE-IMP        TO CCD-OPZIONE.
MOVE lunghezza-campo     TO CCD-INDINT.
MOVE campo-da-controllare TO CCD-ESTERNO

EXEC CICS LINK PROGRAM (CA-ISYOCDD)
          COMMAREA (CCD-AREA) LENGTH(CCD-AREA-LMT)
END-EXEC.
IF CCD-ERRORE = SPACES
  MOVE campo-uscita-funz  TO campo-record
ELSE
  trattamento dall'errore.
    
```

Il significato delle linee è simile a quello del controllo da pannello.

Edit on-line

La modalità tipica di chiamata nel caso in cui sia richiesto l'editing dei dati è la seguente:

```

MOVE CA-AREA-SISTEMA      TO CCD-AREA-SISTEMA.
MOVE funzione-richiesta  TO CCD-FUNZIONE.
MOVE CCD-ROPE-EDT        TO CCD-OPERIONE.
MOVE lung-campo-mappa    TO CCD-INPLNT.
MOVE campo-record        TO campo-uscita-funs
EXEC CICS LINK PROGRAM (CA-ISYOCCD)
                        COMAREA (CCD-AREA) LENGTH (CCD-AREA-LNT)
END-EXEC.
MOVE CCD-ESTERNO         TO campo-mappa.
    
```

Anche in questo caso per il significato delle righe si può fare riferimento al caso già descritto.

In particolare si deve far notare che in CCD-INPLNT deve essere mossa una costante contenete la lunghezza del campo su cui si dovrà spostare il campo editato. Il programma ISYOCCD0 editerà il dato tenendo conto di questa lunghezza.

MODALITÀ DI ATTIVAZIONE BATCH

Nel caso batch l'attivazione del programma di controllo viene effettuata con una CALL al programma ISYRCCD0. Questo programma provvederà a caricare in memoria ISYBCCD0 al momento dell'esecuzione e a cedere a questo il controllo. Questa modalità operativa simula in ambiente batch la funzione di LINK consentendo di non avere il programma di controllo Linkeditato insieme a quello applicativo, con riduzione di spazio in libreria e riallineamento automatico in caso di modifiche a ISYBCCD0.

Solo controllo batch

La modalità tipica di chiamata nel caso in cui sia richiesto il solo controllo dei dati sarà la seguente:

```

MOVE CA-AREA-SISTEMA      TO CCD-AREA-SISTEMA.
MOVE funzione-richiesta  TO CCD-FUNZIONE.
MOVE CCD-KOPE-IMP        TO CCD-OPERIONE.
MOVE lunghezza-campo     TO CCD-INPLMT.
MOVE campo-da-controllare TO CCD-ESTERNO.
CALL 'ISYRCCD0' USING CCD-AREA.
IF CCD-ERRORE = SPACES
    MOVE campo-uscita-funz TO campo-record
ELSE
    trattamento dell'errore.
    
```

Il formato è quindi simile a quello on-line con l'eccezione dell'istruzione di LINK.

Edit batch

La modalità tipica di chiamata nel caso in cui sia richiesto l'editing dei dati è la seguente:

```

MOVE CA-AREA-SISTEMA      TO CCD-AREA-SISTEMA.
MOVE funzione-richiesta  TO CCD-FUNZIONE.
MOVE CCD-KOPE-EDT        TO CCD-OPERIONE.
MOVE lungh-campo-mappa   TO CCD-INPLMT.
MOVE campo-record        TO campo-uscita-funz
CALL 'ISYRCCD0' USING CCD-AREA.
MOVE CCD-ESTERNO         TO campo-mappa.
    
```

Attivazione semplificata

Nel caso di attivazione batch sono anche previste delle modalità di chiamata per edit di tipo semplificato. Il formato è in questo caso il seguente:

```
MOVE funzione-richiesta TO CCD-FUNZIONE.
MOVE CCD-KOPZ-EDB TO CCD-OPTIONE.
MOVE lung-campo-mappa TO CCD-INPLNT.
CALL 'ISYRCCDO' USING CCD-AREA campo-record.
MOVE CCD-ESTERNO TO campo-stampa.
```

In questo modo la modalità di chiamata non è più dipendente dalla funzione e quindi può essere effettuata indipendentemente dalla conoscenza del tipo di editing che verrà effettuato sul campo. In tutti i casi precedenti era infatti necessario conoscere la funzione richiesta per poter effettuare lo spostamento del dato da editare sul campo opportuno dell'area di interfaccia (CCD-AREA). Con questa modalità, richiesta con l'opzione CCD-KOPZ-EDB è lo stesso programma che si preoccupa di effettuare lo spostamento in maniera guidata dalla funzione.

Una funzione simile è disponibile anche in fase di controllo. La modalità di chiamata è in questo caso:

```
MOVE CA-AREA-SISTEMA TO CCD-AREA-SISTEMA.
MOVE funzione-richiesta TO CCD-FUNZIONE.
MOVE CCD-KOPZ-CHB TO CCD-OPTIONE.
MOVE lunghezza-campo TO CCD-INPLNT.
MOVE campo-da-controllare TO CCD-ESTERNO.
CALL 'ISYRCCDO' USING CCD-AREA campo-record.
```


Norme generali

Questo capitolo contiene alcune regole che devono essere rispettate, da una parte per rendere più omogenei i programmi prodotti, dall'altra per poter utilizzare le funzionalità disponibili nell'ambiente di sviluppo.

Si tenga in ogni caso presente che i nomi di tutti gli elementi di un progetto dovranno attenersi alle regole riportate nel documento **NORME DI CODIFICA**.

DOCUMENTAZIONE DEI PROGRAMMI

In testa al programma deve essere inserita la documentazione tecnica del programma stesso. La struttura di questa documentazione è fornita in Figura. Come si può vedere la documentazione viene scomposta logicamente in più parti:

DENOMINAZIONE

Descrizione sintetica delle funzioni svolte dal programma (1-2 righe).

DESCRIZIONE GENERALE

Descrizione funzionale dettagliata del programma con l'indicazione delle operazioni che vengono svolte sugli archivi e, nel caso di programmi interattivi, delle PF gestite con le operazioni ad esse associate.

INPUT

Descrizione dei dati di attivazione del programma con la loro provenienza (pannello, Communication Area, ecc.).

OPERAZIONI

Descrizione del flusso dettagliato del programma.

OUTPUT

Descrizione degli eventuali campi di uscita del programma verso successivi programmi o delle operazioni effettuate sugli archivi.

In fondo alla documentazione deve essere presente una tabella in cui vengono registrate successivamente le operazioni di manutenzione svolte sul programma con l'indicazione della persona che ha effettuato l'intervento.

```

*****
* DENOMINAZIONE *
*   pppass0 - descrizione sintetica *
*****
* DESCRIZIONE GENERALE *
*   Descrizione funzionale del programma. *
*   USO DELLE PF *
*   - PF x : descrizione *
*   - PF x : descrizione *
*   ..... *
*****
* INPUT *
*   Descrizione dei dati in ingresso al programma *
*****
* ELABORAZIONE *
*   Flusso elaborativo del programma *
*****
* OUTPUT *
*   Descrizione dei dati in uscita dal programma *
*****
* DATA | NOMINATIVO | INTERVENTO *
*-----|-----|-----*
* ..../.. | ..... | RILASCIO *
* ..../.. | ..... | ..... *
* ..../.. | ..... | ..... *
* ..../.. | ..... | ..... *
*****

```

Figura 4. Documentazione del programma

TRACCIATI RECORD

Le copy dei tracciati record relativi ad archivi o a segmenti DL/1 sono composti di due parti.

La prima parte conterrà il tracciato record vero e proprio ed avrà la prima riga nel seguente formato:

```
01  xxxx-RECORD.
```

dove xxxx è la sigla identificativa del record.

La seconda parte contiene i dati identificativi del record, cioè la sua lunghezza, la lunghezza della chiave ed il suo nome esterno. Il nome esterno è quello che deve essere utilizzato nelle istruzioni CICS. Il formato di questa parte è il seguente:

```
01  xxxx-RECLNT      PIC S9(4) COMP VALUE nnn.
01  xxxx-KEYLNT     PIC S9(4) COMP VALUE nnn.
01  xxxx-EXTNAME    PIC X(8)  VALUE 'xxxxxxxx'.
```

In appendice sono riportati degli esempi di tracciato record per i file di sistema.

USO DELLE SYSXXX NEI PROGRAMMI BATCH

I programmi batch dovranno rispettare i seguenti standard nell'uso delle SYSxxx.

SYS001 - 009	sono riservate e quindi non possono essere utilizzate applicativamente. La SYS001 è riservata al SORT.
SYS010 - 039	ad uso delle applicazioni per i vari archivi costituenti la base dei dati. Finchè è possibile dovrà essere mantenuta la stessa unità logica per lo stesso file utilizzato da vari programmi all'interno dell'applicazione
SYS040 - 046	dovranno essere utilizzate per le stampanti. La SYS040 coinciderà con la SYSLST.
SYS050 - 055	dovranno essere utilizzate per le unità nastro
SYS060 - 070	potranno essere utilizzate per file di lavoro temporanei in uso all'interno di una job stream
SYS100	File degli utenti
SYS101	File tabelle
SYS102	File delle abilitazioni
SYS103	File delle transazioni
SYS104-106	File delle richieste offline
SYS107	File di salvataggio parametri di ingresso (da ISYBIPM0)

USO DELLE PF NEI PROGRAMMI ON-LINE

L'uso dei tasti funzionali all'interno delle transazioni deve seguire questo standard:

ANNUL	Fine della transazione senza effettuare nessuna operazione
F1	Tasto di help
F3	Fine normale della transazione o ritorno al livello precedente per transazioni su più livelli (es. pannello multiriga e pannello di dettaglio).
F7	Pannello precedente (liste)
F8	Pannello successivo (liste)

Gli altri tasti funzionali possono avere significato diverso in dipendenza della transazione.

STRUTTURA DEI PANNELLI

I pannelli delle transazioni a video dovranno avere la struttura indicata in Figura 5.

Sulla prima riga dovrà essere indicato il codice della transazione (1) e la sua descrizione (2). Tali campi potranno essere variabili su cui il programma applicativo muoverà i valori opportuni richiedendoli al sistema ISY.

Il diagramma mostra un rettangolo con angoli arrotondati, contenente 18 linee orizzontali. Le etichette numerate sono posizionate come segue:

- (1) è centrata sopra la prima linea.
- (2) è centrata sotto la seconda linea.
- (5) è centrata sopra la quinta linea.
- (6) è centrata sopra la sesta linea.
- (4) è centrata sopra la diciannovesima linea.
- (3) è centrata sotto la diciannovesima linea.

Figura 5 - Layout pannello acquisizione dati di attivazione transazioni

L'ultima riga è riservata ai messaggi (3) e ad un campo (4) in cui possono essere indicati parametri opzionali associati alle PF o richieste funzioni.

La riga precedente (5) contiene una breve indicazione dei tasti funzionali previsti.

La parte rimanente (6) sarà il corpo del pannello.

Funzioni di utilità

TRASFORMAZIONE SOURCE BATCH IN ON-LINE E VICEVERSA

Nel caso in cui lo stesso programma debba avere una doppia versione per poter funzionare sia in ambiente batch che in ambiente on-line si può utilizzare una tecnica di scrittura che consenta di gestire la doppia versione con un unico sorgente.

Questa soluzione consiste nell'inserire nel sorgente del programma scritto per uno dei due ambienti anche le istruzioni relative all'altro ambiente anche se inserite come commento (con un asterisco a colonna 7). Un programma di utilità consentirà di ottenere da questa l'altra versione scambiando le istruzioni commentate con quelle operative. Il programma ISYBBOT0 permette di gestire con un unico sorgente programmi che possono funzionare sia in batch che in on-line.

Il programma sorgente deve contenere al suo interno sia le istruzioni per l'elaborazione batch che quelle per l'elaborazione on-line. I blocchi di istruzioni diversi per i due ambienti devono essere caratterizzate da una scheda con le seguenti caratteristiche:

- Prima di un blocco di istruzioni da utilizzare solo per l'on-line deve essere inserita una scheda con la stringa di caratteri *\$\$ONLSTR a partire da colonna 7.
- Al termine di un blocco di istruzioni da utilizzare solo per l'on-line deve essere inserita una scheda con la stringa di caratteri *\$\$ONLEND a partire da colonna 7.
- Prima di un blocco di istruzioni da utilizzare solo per il batch deve essere inserita una scheda con la stringa di caratteri *\$\$BTCSTR a partire da colonna 7.
- Al termine di un blocco di istruzioni da utilizzare solo per il batch deve essere inserita una scheda con la stringa di caratteri *\$\$BTCEND a partire da colonna 7.
- Eventuali commenti che devono essere inseriti all'interno dei blocchi così individuati devono avere, oltre all'asterisco in colonna 7, anche un asterisco in colonna 8.

Con questa regola potrà essere quindi realizzato solo un programma in una delle due modalità. Questo programma conterrà anche le istruzioni necessarie per l'altra modalità ma queste presenteranno un asterisco a colonna 7 e quindi verranno considerate commenti.

La modalità di funzionamento del programma è determinata dal suo nome che per il batch deve presentare una "B" come quarto carattere mentre per l'on-line questo carattere è "O".

Il programma ISYBBOT0, funzionante in ambiente CMS ed attivabile con la procedura EXEC PGMTRANS, permette di ottenere da uno dei due programmi l'altro, purchè siano state seguite le modalità di scrittura suindicate.


```

-----PROGRAMMA BATCH ORIGINALE-----
**$BTCSTR**INIZIO PARTE BATCH*****
      READ ISYDFTAB
      INVALID KEY GO TO CCD-TB-FILE-EX.

**$BTCEND**FINE  PARTE BATCH*****

**$ONLSTR**INIZIO PARTE ONLINE*****
*
*   EXEC CICS HANDLE CONDITION
*           NOTFND (CCD-TB-FILE-EX)   END-EXEC.
*
**** ISYDFTAB È UNA VARIABILE E DEVE ESSERE IMPOSTATA
*
*   EXEC CICS READ   DATASET (ISYDFTAB)
*           RIDFLD (TB-CHIAVE)
*           INTO (TAB-RECORD)         END-EXEC.
*
**$ONLEND**FINE  PARTE ONLINE*****

      MOVE SPACES      TO ERRORE.

-----PROGRAMMA ON-LINE GENERATO-----
**$BTCSTR**INIZIO PARTE BATCH*****
*
*   READ ISYDFTAB
*   INVALID KEY GO TO CCD-TB-FILE-EX.
*
**$BTCEND**FINE  PARTE BATCH*****

**$ONLSTR**INIZIO PARTE ONLINE*****

      EXEC CICS HANDLE CONDITION
              NOTFND (CCD-TB-FILE-EX)   END-EXEC.

**** ISYDFTAB È UNA VARIABILE E DEVE ESSERE IMPOSTATA

      EXEC CICS READ   DATASET (ISYDFTAB)
              RIDFLD (TB-CHIAVE)
              INTO (TAB-RECORD)         END-EXEC.

**$ONLEND**FINE  PARTE ONLINE*****

      MOVE SPACES      TO ERRORE.

```

Figura 6. Esempio di trasformazione da batch a On-line

La modalità di attivazione è la seguente:

PGMTRANS nome-programma

Il nome di programma che deve essere fornito è quello da trasformare. Se questo presenta il carattere "B" in quarta posizione si assume che il programma è batch e deve essere ottenuto l'on-line il cui nome ha il carattere "B" sostituito con "O". Viene assunta l'altra condizione in caso contrario.

La trasformazione avviene solo per programmi COBOL presenti sul disco 103. I programmi possono essere anche in formato compresso purchè in questo caso il Filetype sia PACKC. La procedura provvede automaticamente alla sua espansione trasformando il Filetype in COBOL.

Il programma da generare non deve già esistere nè in formato COBOL nè PACKC. Nel caso esista l'operazione di trasformazione non ha luogo.

L'operazione di trasformazione da batch a on-line segue quindi le seguenti regole:

1. Da ogni istruzione del primo programma dopo un'istruzione ***\$BTCSTR** viene generata sul secondo programma un'istruzione con lo stesso contenuto ma di tipo commento ponendo un asterisco a colonna 7 fino a quando non viene incontrata un'istruzione ***\$BTCEND**
2. Da ogni istruzione del primo programma dopo un'istruzione ***\$ONLSTR** viene generata sul secondo programma un'istruzione con lo stesso contenuto forzando
3. uno spazio a colonna 7 fino a quando non viene incontrata un'istruzione ***\$ONLEND**
4. La forzatura di cui al punto precedente non avviene se l'istruzione ha un asterisco a colonna 8.
5. Tutte le altre istruzioni vengono trasferite immutate sul secondo programma.

In questa trasformazione si presuppone quindi che tutte le istruzioni fra la ***\$ONLSTR** e la ***\$ONLEND** nel programma iniziale presentino un asterisco a colonna 7. Siano cioè dei commenti.

La trasformazione inversa avviene con le stesse modalità tranne le ovvie differenze.

La parte rimanente delle righe ***\$xxxxxx** è libera e può essere utilizzata per commento.

Gestione delle tabelle di controllo

Il programma di controllo gestisce anche il controllo tabellare dei campi, cioè permette di verificare che un determinato campo contenga solo valori appartenenti ad un insieme predefinito (tabella di valori).

Ogni tabella è identificata da una sigla di 6 caratteri. Il controllo tabellare viene richiesto con un codice funzione composto dalla costante "TB" e dalla sigla di 6 caratteri.

Le tabelle sono contenute in un archivio apposito di nome "ISYTAB". La gestione di un'ulteriore tabella può essere effettuato unicamente inserendo gli opportuni record all'interno di questo archivio. È sufficiente questa operazione per poter richiedere il controllo con una sigla pari a quella con cui sono stati inseriti questi record.

TABELLE PER GESTIONE DEI SINGOLI BIT DI UN BYTE

Queste tabelle vengono utilizzate dalla funzione "BT" per assegnare un carattere identificativo del significato ad i singoli bit di un byte.

Il formato di queste tabelle è quindi diverso da quello delle normali tabelle. In particolare esiste un record con TAB-TIPOC = "E" e TAB-CODICE = spazio che nei primi 8 byte di TAB-CAMPI contiene i valori in formato esterno corrispondenti agli 8 bit. Se ad esempio questi valori sono "ABCDEFGH" quando i singoli bit vengono espansi in 8 byte, il primo byte avrà il valore "A" se il corrispondente bit è acceso, il secondo il valore "B", e così via.

Gli ulteriori elementi della tabella avranno in TAB-CODICE uno dei valori degli 8 byte e sulla descrizione il significato corrispondente.

CONTROLLO DELL'ABILITAZIONE

A fronte di una tabella può essere richiesto che in fase di controllo venga verificato che l'utente che in quel momento sta lavorando sia abilitato a fornire il valore digitato.

Per una tabella su cui è previsto questo ulteriore controllo devono essere inseriti dei record opportuni all'interno dell'Archivio Abilitazioni.

Il campo ABI-SIGLA sarà il nome della tabella di controllo (presente nel campo funzione). Per ogni utente in questo archivio dovranno essere indicati i valori che questo può digitare.

È possibile indicare delle ulteriori limitazioni nei campi ABI-LIMITAZIONI. Può essere indicato in questi un e massimo di 20 transazioni per le quali vale l'abilitazione.

Sono previsti tre possibili livelli di abilitazione che consentono di semplificare la gestione dell'archivio delle abilitazioni.

Il livello più generale si ha quando il valore viene associato all'utente "PUBLIC". Valori che si trovano in queste condizioni non sono sottoposti ad alcuna limitazione e possono essere digitati da qualsiasi utente.

Il successivo livello è quello di GRUPPO. Ogni utente può essere infatti assegnato ad un raggruppamento di utenti individuato da un codice sul record dell'archivio utenti. I valori associati ad un utente coincidente con il codice del gruppo possono essere digitati da tutti gli utenti di quello stesso gruppo.

Il livello più dettagliato di abilitazione è quello di utente. Ogni utente può digitare i valori associati al suo codice identificativo.

Ogni utente può in altre parole digitare i valori associati al suo identificatore, al codice del gruppo cui appartiene e tutti quelli associati a "PUBLIC".

Contenuto del file scheletro

Il file scheletro (o schema) contiene la struttura di base di una tipologia di programma, cioè le istruzioni che gestiscono il flusso delle informazioni. Nello scheletro saranno inoltre presenti tutte le istruzioni legate al sistema operativo, per consentire di rendere le specifiche indipendenti dalla piattaforma target dell'applicazione. Inoltre si fa presente che spesso da una stessa specifica, tramite schemi differenti, possono essere ottenuti programmi differenti (ad esempio, una stampa on-line ed una analoga batch).

All'interno del file scheletro esistono tre categorie di istruzioni:

- *istruzioni fisse* che devono essere riportate inalterate sul programma sorgente
- *istruzioni contenenti dati variabili* che verranno sostituite con il loro valore dal programma generatore
- *funzioni* che consentono di pilotare la generazione.

ISTRUZIONI FISSE

Un'istruzione fissa è una riga del programma scheletro sulla quale non siano presenti variabili (vedi paragrafo successivo).

Il programma generatore trasporta una istruzione fissa sul programma sorgente generato esattamente come è presente nel file scheletro.

ISTRUZIONI CON DATI VARIABILI

All'interno delle istruzioni presenti sul file scheletro possono essere presenti variabili. Una variabile è una stringa di caratteri (massimo 8) preceduta dal carattere & e seguita dal carattere < (es. &PROGRAM<).

Quando nell'istruzione il programma generatore trova una variabile, questa viene sostituita dal suo valore. Se non è stata definita nessuna variabile con quel nome, il programma generatore emette un messaggio di errore e non effettua la traduzione.

Il valore corrente della variabile viene sostituito a partire dalla posizione in cui iniziava la variabile ed unicamente nello spazio fino all'inizio della successiva parola nell'istruzione che verrà comunque mantenuta nella stessa posizione. Se la lunghezza del valore è tale da non entrare nello spazio previsto, viene effettuato un troncamento ed inviato un messaggio di errore.

È previsto che in coda alla variabile possano essere inseriti altri caratteri (es. &PROGET<O&PROGRAM<&VERS oppure &MAPPA<-O). In questi casi i caratteri in coda verranno reinseriti in coda al valore senza spazi intercorrenti e comunque nei limiti dati dalla successiva parola. Nella stringa di caratteri in coda ad una variabile può anche essere presente un'altra variabile per la quale avverrà la traduzione.

Le variabili possono provenire da tre diverse fonti:

- file di inizializzazione di variabili per default, letto inizialmente prima delle specifiche e dello schema
- informazioni fornite nelle specifiche
- impostazioni a livello di file scheletro con l'apposito comando.

Una classe particolare di variabili proveniente dal file scheletro verrà descritta con la funzione di ripetizione.

IMPOSTAZIONE DI UNA VARIABILE (&SET)

La funzione &SET consente di definire una variabile associandole un valore. Il suo formato è:

&SET variabile = valore

dove variabile deve avere il formato già descritto, cioè iniziare con & e terminare con <. Il valore può essere sia numerico che carattere.

CALCOLO DI UNA VARIABILE (&COMP)

La funzione &COMP consente di associare ad una variabile il risultato di un calcolo su due soli operandi. Il suo formato è:

&COMP variabile = operando-1 operatore operando-2

dove i due operandi possono essere variabili o valori numerici costanti ed operatore può essere uno di quelli standard per la definizione delle operazioni elementari (+, -, *, /).

INSERIMENTO DI UNA ROUTINE DA FILE SPECIFICHE (&INCLUDE)

La funzione &INCLUDE permette di inserire un blocco di istruzioni provenienti dal file specifiche e che segue una delle schede di identificazione ivi definite. Il suo formato è:

&INCLUDE=scheda-identificazione

dove scheda-identificazione deve essere una di quell già indicate nella descrizione del file specifiche.

L'inserimento avviene a partire dal punto in cui è presente la funzione. Le istruzioni prelevate dal file specifiche verranno riportate inalterate sul programma generato.

SALTO INCONDIZIONATO (&GOTO)

L'istruzione &GOTO consente di modificare il flusso di generazione posizionandosi sulla riga successiva contraddistinta da una particolare label. Il suo formato è il seguente:

&GOTO label

La label è una stringa alfanumerica (massimo 8 caratteri) che inizia con un carattere -. La label potrà essere presente su una qualsiasi delle righe successive a quella in cui è presente l'istruzione &GOTO. La label dovrà essere l'unica stringa di caratteri presente sulla riga ed iniziare con il primo carattere della riga.

La funzione &GOTO opera solamente in avanti, non è cioè possibile effettuare salti a righe precedenti a quella corrente. Tutte le righe contenute fra il GOTO e la label non verranno trattate dal programma generatore.

CONFRONTO (&IF)

La funzione &IF consente di confrontare due variabili o una variabile con un valore e di intraprendere un'azione in caso di esito positivo. Il formato è:

&IF operando-1 operatore operando-2 azione

dove gli operandi possono essere variabili o costanti e operatore può essere EQ, NE, GT, GE, LT, LE. L'unica azione prevista è la funzione &GOTO.

CICLO (&LOOP)

La funzione &LOOP consente di ripetere le istruzioni successive fino alla &ENDLOOP il numero di volte specificato. Il suo formato è il seguente:

&LOOP nn

RIPETIZIONE DI UN BLOCCO DI ISTRUZIONI (&REPEAT)

La funzione &REPEAT consente di ripetere in loop il blocco di istruzioni compreso tra questa scheda e quella contenente &ENDREP. Il loop è relativo ad uno dei blocchi di definizione del file specifiche. Durante ognuno dei cicli saranno disponibili delle variabili speciali relative ad uno degli aspetti definiti nel blocco. Il suo formato è il seguente:

&REPEAT=xxxxxx

dove xxxxxx può assumere uno dei seguenti valori:

*FIELD è associato al blocco di definizione del pannello. Vengono effettuati tanti cicli quanti sono i campi variabili presenti sul pannello (quindi ad esclusione dei campi fissi). Ogni ciclo corrisponderà ad uno dei campi e saranno disponibili tutte le variabili ad esso relative. Le variabili sono:

FIELD(RIG)<	numero di riga di pannello su cui è il campo
FIELD(COL)<	colonna di inizio del campo
FIELD(NOM)<	nome del campo
FIELD(ATT)<	attributi del campo
FIELD(ATT1)<	attributo di protezione
FIELD(ATT2)<	attributo di luminosità
FIELD(OPR)<	campo opzioni
FIELD(OP1)<	primo carattere del campo opzioni
FIELD(OP2)<	secondo carattere del campo opzioni
FIELD(OP3)<	terzo carattere del campo opzioni
FIELD(OP4)<	quarto carattere del campo opzioni
FIELD(OP5)<	quinto carattere del campo opzioni
FIELD(FUN)<	campo funzione
FIELD(LIN)<	lunghezza del campo
FIELD(POS)<	posiz. progress. (num. di caratt) su pannello
FIELD(PIC)<	picture del campo in formato interno
FIELD(INI)<	valore iniziale in campo interno

```

$FLD (NOM) <: valore iniziale in mappa
$FLD (DNL) <: lunghezza del campo interno (byte)
$FLD (TRK) <: tipo blocco (chiave, gruppo,...)
$FLD (BLK) <: nome blocco
$FLD (MLT) <: moltiplicatore blocco
$FLD (NUM) <: progressivo del campo
    
```

*MAPPA è simile al caso precedente. Vengono effettuati tanti cicli quanti sono i campi su pannello indipendentemente dal fatto che siano fissi o variabili. Le variabili a disposizione sono le stesse indicate per il caso precedente.

*INPUT è associato al blocco di definizione dei parametri di attivazione. Viene effettuato un ciclo per ognuno dei campi di attivazione. Le variabili disponibili in ogni ciclo sono:

```

$INP (NOM) <: nome del campo
$INP (OPE) <: campo opzioni
$INP (OP1) <: primo carattere del campo opzioni
$INP (OP2) <: secondo carattere del campo opzioni
$INP (OP3) <: terzo carattere del campo opzioni
$INP (OP4) <: quarto carattere del campo opzioni
$INP (OP5) <: quinto carattere del campo opzioni
$INP (FUN) <: campo funzione
$INP (LNT) <: lunghezza del campo
$INP (IDE) <: stringa di identificazione parametro
$INP (PIC) <: picture del campo in formato interno
$INP (INI) <: valore iniziale in campo interno
$INP (RIG) <: numero di riga su cui verrà visualizzato
$INP (INL) <: lunghezza del campo interno (byte)
$INP (NUM) <: progressivo del campo
    
```

*TASTI è associato al blocco di specifiche relativo alla definizione dei tasti funzionali. Viene effettuato un ciclo per ognuno dei tasti definiti. Le variabili sono:

```

$TAS (NOM) <: identificatore tasto
$TAS (OPE) <: campo opzioni
$TAS (OP1) <: primo carattere del campo opzioni
$TAS (FUN) <: sigla identificativa della funzione
$TAS (PGM) <: programma o routine da attivare
$TAS (DES) <: descrizione della funzione
$TAS (KEY) <: chiavi di attivazione (per ISYOVISO)
$TAS (NUM) <: progressivo del campo
    
```

*STAMPA è associato alla definizione delle righe di stampa (o in genere di visualizzazione a lista). Viene effettuato un ciclo per ognuno dei campi fissi o variabili presenti sulle varie linee di stampa. Le variabili sono le seguenti:

```

$STM (RIG) <: numero progressivo riga
$STM (NOM) <: nome del campo
$STM (OPE) <: campo opzioni
$STM (OP1) <: primo carattere del campo opzioni
$STM (OP2) <: secondo carattere del campo opzioni
$STM (OP3) <: terzo carattere del campo opzioni
$STM (OP4) <: quarto carattere del campo opzioni
$STM (OP5) <: quinto carattere del campo opzioni
    
```

```

&STM(FUN)<: campo funzione
&STM(LMT)<: lunghezza del campo
&STM(PIC)<: picture del campo in formato interno
&STM(INI)<: valore iniziale in campo interno
&STM(KMP)<: valore costanti di stampa
&STM(INL)<: lunghezza del campo interno (byte)
&STM(TBK)<: tipo del blocco (HP o DT)
&STM(BLK)<: nome del blocco
&STM(MLT)<: moltiplicatore del blocco
&STM(NUM)<: progressivo del campo
    
```

*GRUPPI è associato alla definizione delle chiavi di gruppo. Una chiave di gruppo è costituita da tutti i campi dichiarati come appartenenti al gruppo. Viene effettuato un ciclo per ognuno di tali campi e vengono fornite le seguenti variabili:

```

&STM(NOM)<: nome del campo
&STM(OPZ)<: campo opzioni
&STM(OP1)<: primo carattere del campo opzioni
&STM(OP2)<: secondo carattere del campo opzioni
&STM(OP3)<: terzo carattere del campo opzioni
&STM(OP4)<: quarto carattere del campo opzioni
&STM(OP5)<: quinto carattere del campo opzioni
&STM(FUN)<: campo funzione
&STM(LMT)<: lunghezza del campo
&STM(PIC)<: picture del campo in formato interno
&STM(INI)<: valore iniziale in campo interno
&STM(INL)<: lunghezza del campo interno (byte)
    
```

