|  |  |
| --- | --- |
| CADIT_R2  CAD IT S.p.A.  Via Torricelli, 44/a  37136 Verona  Tel 045 – 8211111  Fax 045 - 8211110  www.cadit.it  cadit@cadit.it | S E T  Sistema Esazione Tributi  **Modello della struttura** |

RIEPILOGO INFORMAZIONI

|  |  |
| --- | --- |
| Titolo | Modello della Struttura |
| Autore | CAD IT |
| Versione del | **30.06.2014** |
| Nome del File | D:\EquitaliaServizi\NuoviProgetti\SET\_DocumentazioneSistema\Documentazione\_descrittiva\02\_01\_ModelloDellaStruttura.docx |
| Modello Impiegato | CADITOPE.DOT |
| Data di Creazione | 30/06/2014 |
| Approvato da |  |
| Firma Responsabile |  |

INDICE

1 INTRODUZIONE 1

2 Struttura di un programma TP 2

2.1 Comportamento di un TP 2

2.1.1 TP di gestione 2

2.1.2 TP di inquiry 5

2.2 Struttura di un TP 8

2.2.1 Parte iniziale 8

2.2.2 Section di Procedure division 11

2.2.2.1 Descrizione delle section standard 16

3 Navigazione 18

3.1 Caratteristiche della Navigazione 18

3.2 Schema di funzionamento della navigazione 19

3.3 Come si utilizza la navigazione 20

4 Gestione delle mappe video 21

4.1 Le mappe video 21

4.2 Utilizzo di un meccanismo di paginazione standard 22

4.3 La paginazione standard CITY 23

4.3.1 Concetti base: ciclo, mappa, unità 23

4.3.2 Il file di appoggio 26

4.3.3 Gestione delle mappe 27

4.3.3.1 Elenco delle primitive di controllo della paginazione 27

5 Accesso a base dati tramite standard CITY 28

5.1 copy di accesso 28

5.1.1 Primitive Virtuali Per L'accesso Ai Dati 28

5.2 copy di smistamento 30

5.3 Le routine di accesso 30

5.3.1 DB2 31

5.3.1.1 copy di accesso 31

5.3.1.2 copy di smistamento 31

5.3.1.3 Routine di accesso 32

6 Gestione delle autorizzazioni 33

6.1 Gli elementi per la gestione delle autorizzazioni 33

6.1.1 Moduli di sicurezza 33

6.1.2 Profili di sicurezza 33

6.2 Metodo di individuazione dei diritti d'accesso 34

6.3 Sicurezza nell'accesso 34

6.3.1 Accesso a funzioni 35

6.3.2 Accesso ai dati 35

7 Gestione delle stampe on-line 38

7.1.1 Modelli di stampa 38

7.2 Routine di formattazione della stampa 38

7.3 Associazione fra Modello di stampa e Routine di formattazione 38

7.4 Stampante utilizzata 39

7.5 Elenco delle funzioni di stampa 40

7.6 Schemi delle fasi di stampa 41

8 Struttura standard di un programma batch 46

8.1.1 Copy zzCPB 53

8.2 Flusso di esecuzione di un programma batch 54

9 Schede Parametro per programmi batch 55

9.1 Definizione delle schede parametri 55

9.2 File cadpar 55

9.2.1 Parametri obbligatori 56

9.2.2 Parametri facoltativi di sistema 56

9.2.3 Parametri utente per il programma 56

9.3 File cadset 57

10 Segnalazioni 58

10.1 La routine di segnalazione zzRBSIG 58

11 Errori 59

11.1 La routine di gestione degli errori zzRBERR 59

12 Gestione delle stampe batch 60

12.1 La routine di stampa zzRBPRT 60

12.2 Composizione della stampa 60

12.2.1 La testata standard 60

12.2.2 La riga standard di fine tabulato 61

13 Raggruppamento in moduli e interazione tra i componenti 62

# INTRODUZIONE

Questo documento descrive la struttura dei programmi batch e tp, il raggruppamento in moduli e l’interazione tra i componenti relativi alle funzionalità della procedura SET secondo gli standard CITY.

## 

# Struttura di un programma TP

Il presente capitolo ha lo scopo di presentare la struttura di massima che deve avere un programma TP implementato secondo lo standard CITY.

Le regole esposte sono da ritenersi valide per i programmi TP che seguono lo standard CITY.

Per semplificare il compito dello sviluppatore, in ambiente CITY è stato predisposto un insieme di strumenti che svolgono la maggior parte delle attività generiche (e quindi indipendenti dalle problematiche dello specifico programma che si deve realizzare).

In questo documento è presentata la struttura di programmi TP di gestione e di inquiry, motivando i criteri organizzativi a cui i programmi stessi devono sottostare.

NOTA: il prefisso "zz" nei nomi dei file indica la sigla della procedura cui il file appartiene; si veda il documento relativo alla Naming Convention *(Regole di attribuzione dei nomi)* per ulteriori indicazioni sull'attribuzione dei nomi secondo lo standard CITY.

## Comportamento di un TP

I programma TP si possono a grandi linee dividere in due classi:

programmi di *gestione*, per la modifica e l'aggiornamento degli archivi

programmi di *inquiry*, per la ricerca e la osservazione di elementi in archivio, senza modifica

Nei paragrafi successivi saranno descritte più in dettaglio le attività svolte da queste due tipologie di programmi.

### TP di gestione

I programmi detti "di gestione" hanno lo scopo di permettere all'operatore di inserire, modificare, cancellare o duplicare dati da un archivio della base dati; il loro comportamento, a meno di casi che necessitino di accessi ad archivi multipli o di controlli complessi sui dati introdotti, è piuttosto ripetitivo.

Lo schema di massima è il seguente:

1. L'operatore indica il tipo di operazione che vuole effettuare:

Inserimento di un nuovo elemento nella base dati

Variazione di un elemento già esistente

Annullamento (cancellazione) di un elemento già esistente

Duplicazione di un elemento già esistente

Inquiry secca su un singolo record (analoga alla variazione ma con tutti i campi protetti)

2. L'operatore indica il valore dei campi di chiave che contraddistinguono l'elemento e permettono di reperirlo nella base dati

3. L'operatore preme invio

4. Il programma controlla la validità dell'operazione richiesta

5. Il programma controlla la validità dei valori dei campi di chiave e verifica l'esistenza di un record con la chiave indicata:

se l'operazione richiesta è Inserimento di un nuovo elemento o Duplicazione, *non deve* esistere in archivio alcun elemento con la chiave indicata

viceversa se l'operazione richiesta è Variazione o Annullamento di un elemento già esistente, l'elemento con la chiave indicata *deve* esistere in archivio

6. A questo punto il comportamento si diversifica a seconda dell'operazione richiesta:

**Inserimento**

6.1. il programma protegge i campi di chiave per evitare che l'operatore possa modificarli

6.2. l'operatore indica il valore degli altri campi e preme invio

6.3. il programma controlla la correttezza dei valori inseriti, quindi richiede conferma per l'inserimento dei dati

**Variazione**

6.1. il programma mostra sul video tutti i dati dell'elemento trovato ma protegge i campi di chiave per evitare che l'operatore possa modificarli

6.2. l'operatore modifica i campi che ritiene opportuni, quindi preme invio

6.3. il programma controlla la correttezza dei valori modificati, quindi richiede conferma per la variazione dei dati

**Annullamento**

6.1. il programma mostra sul video tutti i dati dell'elemento trovato proteggendoli per evitare che l'operatore possa modificarli, quindi richiede subito la conferma per l'annullamento dell'elemento mostrato

**DUPLICAZIONE**

6.1. il programma mostra sul video tutti i dati dell'elemento trovato e lascia sprotetti tutti i campi, compresi quelli di chiave

6.2 l'operatore modifica i campi che ritiene opportuni, di cui almeno uno deve essere di chiave, quindi preme INVIO

6.3 il programma controlla la correttezza dei valori modificati, quindi richiede conferma per la variazione dei dati

6.4 in tutti e quattro i casi, se l'operazione viene confermata, il programma aggiorna l'archivio.

Ad una semplice analisi si osserva come l'intera attività non sia altro che una alternanza di operazioni svolte dai due agenti: l'operatore ed il programma. Rivediamo schematicamente le diverse attività:

|  |  |
| --- | --- |
| **Operatore** | **Programma** |
|  | prepara i campi della prima mappa |
| Indica tipo operazione |  |
| Indica il valore dei campi di chiave |  |
| Preme invio |  |
|  | controlla la validità dell'operazione richiesta |
|  | controlla i valori campi di chiave |
|  | INS: la chiave indicata *non deve* esistere in archivio  VAR/ANN: la chiave indicata *deve* esistere in archivio |
| **Inserimento** |  |
|  | protegge a video i campi di chiave |
| Indica il valore degli altri campi |  |
| Preme invio |  |
|  | controlla correttezza valori inseriti |
|  | richiede conferma per inserimento nuovi dati |
| **Variazione** |  |
|  | protegge a video i campi di chiave |
|  | mostra a video tutti i dati dell'elemento trovato |
| Modifica il valore dei campi |  |
| Preme invio |  |
|  | controlla la correttezza dei valori modificati |
|  | richiede conferma per variazione dei dati |
| **Annullamento** |  |
|  | protegge a video tutti i campi |
|  | mostra sul video tutti i dati dell'elemento trovato |
|  | richiede subito conferma per annullamento dei dati |
| Conferma la modifica |  |
| Preme invio |  |
|  | aggiorna l'archivio |
| **duplicazione** |  |
|  | lascia sprotetti tutti i campi |
|  | mostra a video tutti i dati dell'elemento trovato |
| Modifica il valore dei campi |  |
| Preme invio |  |
|  | controlla la correttezza dei valori modificati |
|  | richiede conferma per duplicazione dei dati |

Considerazioni

Nella maggior parte dei casi un programma di gestione permette l'aggiornamento di *un unico archivio*[[1]](#footnote-1) e richiede *un'unica mappa video*, usata sia per l'inserimento che per la visualizzazione dei dati.

Tale mappa deve contenere, oltre ad un'eventuale intestazione:

un campo per l'indicazione del tipo-operazione

tutti i campi di chiave

tutti i campi non di chiave

Devono essere predisposti due giri di controllo dei campi di mappa:

il primo ha il compito di controllare i campi di chiave e verificare la presenza in archivio del record corrispondente alla chiave indicata;

il secondo ha il compito di controllare i restanti campi e predisporsi all'aggiornamento dell'archivio.

Se i campi sono troppi per stare su un'unica mappa diventa necessario spezzare l'attività su più mappe.

La conferma

La richiesta della conferma al temine dell'operazione è determinata dal valore del flag ZMAPINV associato alla mappa da confermare; nella maggior parte dei casi è sufficiente porre a 0 il valore di questo flag, indicando così alla CITY che la richiesta deve venire gestita in modo automatico.

La richiesta della conferma avviene solo se il relativo campo di mappa è vuoto (SCNF***yyi*** - switch di conferma per la mappa ***i*** del programma zzPT***yy***).

Se viene riscontrato qualche errore il campo viene ripulito; viene comunque pulito in ogni caso alla fine del programma.

Anche durante la richiesta della conferma è possibile spostarsi indietro fra le mappe (se si è agganciati alla paginazione primaria ciò è possibile premendo il tasto PF7)

Ad esempio dopo aver inserito le mappe 1, 2 e 3 è possibile tornare indietro (ad esempio alla mappa 1), modificare i dati e quindi premendo invio, se tutto è a posto, ci si trova alla mappa 4.

### TP di inquiry

I programmi detti "di inquiry" hanno lo scopo di permettere all'operatore di ricercare dati già presenti in archivio sulla base di determinate condizione e di mostrare un elenco sintetico o il contenuto dettagliato dei record trovati; il loro comportamento, come per i programmi di gestione è normalmente piuttosto semplice e ripetitivo.

Lo schema di massima è il seguente:

1. L'operatore indica le condizioni per la ricerca dei dati (in genere si tratta di specificare un intervallo); tali condizioni sono espresse in funzione dei valori dei campi di chiave

2. L'operatore preme invio

3. Il programma cerca in archivio i record che rispettano la condizione di ricerca e ne prepara un elenco

4. Quando tutti gli elementi sono stati trovati, il programma mostra a video l'elenco sintetico

5. L'operatore seleziona il record di cui desidera vedere i dati in dettaglio, muovendosi fra le diverse pagine dell'elenco

6. L'operatore preme invio

7. il programma mostra sul video il dettaglio dei dati del record selezionato, senza possibilità di modificarli

8. l'operatore può continuare a ripetere il processo di selezione e osservazione del dettaglio muovendosi fra le mappe video grazie al meccanismo della paginazione (tasti PF7-PF8, PF5-PF6, invio).

9. l'operatore interrompe l'attività premendo PF2.

Rivediamo schematicamente le diverse attività:

|  |  |
| --- | --- |
| **Operatore** | **Programma** |
|  | prepara i campi della prima mappa |
| indica i campi per la ricerca |  |
| preme invio |  |
|  | ricerca il primo elemento che rispetti le condizioni |
|  | ripete la ricerca fino a che trova elementi in archivio |
|  | mostra a video l'elenco di tutti gli elementi trovati |
| Seleziona un elemento dall'elenco |  |
| preme invio |  |
|  | Verifica gli elementi selezionati |
|  | ricerca gli elementi selezionati |
|  | copia il valore dei suoi campi sulla mappa di dattaglio e la mostra |
| preme PF2 |  |
|  | riparte dall'inizio |

Considerazioni

Nella maggior parte dei casi un programma di inquiry permette l'accesso ad *un unico archivio*[[2]](#footnote-2) e richiede *due mappe video*, usate per la visualizzazione dei dati.

La prima mappa deve contenere, oltre ad un'eventuale intestazione:

uno o più campi per l'indicazione dell'intervallo di ricerca in funzione dei campi di chiave

una serie di righe contenenti ciascuna:

un campo per la selezione del record

uno o più campi per la visualizzazione sintetica del record

La seconda mappa deve contenere, oltre ad un'eventuale intestazione tutti i campi dell'archivio (di chiave e non) che devono essere presentati dal programma.

Per la prima mappa è opportuno predisporre due giri di controlli

il primo ha il compito di controllare l'intervallo per cui si vuole effettuare la ricerca ed estrarre l'elenco di tutti gli elementi dell'archivio che rispettano le condizioni indicate;

il secondo ha il compito di determinare fra glie elementi dell'elenco quelli marcati dall'operatore, quindi cercarli in archivio e ricopiarne i campi in dettaglio sulla seconda mappa.

Per la seconda mappa non serve alcuno giro di controllo, poiché i dati prelevati dall'archivio devono solo venire mostrati e non modificati.

Se i campi sono troppi per stare su un'unica mappa diventa necessario spezzare l'attività su più mappe.

## Struttura di un TP

Ogni programma TP prodotto secondo lo standard CITY ha una struttura specifica, costituita da due parti essenziali:

una parte iniziale, in cui vengono definiti i dati e incluse le COPY (di sistema e dell'utente);

una parte operativa in cui vengono implementati i controlli, gli accessi agli archivi, le stampe, ecc.

Analizziamo il contenuto di queste due parti.

### Parte iniziale

Le parti indicate in grassetto indicano i punti che variano in funzione dei compiti del particolare programma che si sta implementando, le altre parti sono pressoché identiche per tutti i programmi.

Le sigle in grassetto hanno il seguente significato:

***yy*** Sigla del programma (due caratteri alfanumerici)

***n*** Numero totale di mappe video usate dal programma

***i*** Numero generico di una mappa (compreso tra 1 e ***n***)

***k*** Dimensioni dell'area di lavoro per i dati delle mappe

***xxx*** Sigla di un archivio usato dal programma. In generale coincide con la sigla del tracciato memorizzato su quell'archivio, ma non sempre: è possibile infatti lavorare con archivi contenenti più di un tracciato.

Una parte delle operazione svolte dal programma dipendono dal monitor TP o dal DBMS che si usano, altre invece sono indipendenti dall'ambiente.

***< \*\*\* SET e DOC \*\*\* >***

ID DIVISION.

PROGRAM-ID. ZZPT***yy***. (***yy*** sono due caratteri alfanumerici e identificano il programma)

COPY ZZWAUT. (autore: CAD)

DATE-WRITTEN. ***<data di realizzazione>***.

DATE-COMPILED.

COPY ZZWREM. (Remarks)

ENVIRONMENT DIVISION.

CONFIGURATION SECTION.

COPY ZZWCFG. (Source-Computer / Object-Computer)

SPECIAL-NAMES.

DECIMAL-POINT IS COMMA.

DATA DIVISION.

WORKING-STORAGE SECTION.

\* +------------------------------------------------------------+

\* + VARIABILI AD USO DEL SISTEMA +

\* +------------------------------------------------------------+

01 ZPGM PIC X(08) VALUE 'ZZPT***yy***'. (nome del programma = PROGRAM-ID)

01 ZVER PIC 9(04) VALUE **1**. (numero della versione)

01 ZREL PIC 9(04) VALUE 1.

\* +------------------------------------------------------------+

\* + VARIABILI PER LA GESTIONE DELLE MAPPE +

\* +------------------------------------------------------------+

01 ZMAPSEG PIC S9(04) COMP VALUE **+*n***. (***n*** = numero di mappe video)

01 ZMAPTAB. (tabella con i valori iniziali dei flag di mappa)

03 ZMAPELE-01. (flag mappa 1)

...

03 ZMAPELE-***n***. (flag mappa ***n***)

...

\* +------------------------------------------------------------+

\* + DATI PER LA GESTIONE MAPPE DEL SISTEMA +

\* +------------------------------------------------------------+

01 ZWRK-MAPSYS.

03 ZMAPNUM PIC 9(04). (numero della mappa corrente)

03 ZMAPMAS PIC 9(04). (numero massimo di mappe)

03 ZWRK-MAPTAB. (tabella con i valori correnti dei flag di mappa)

05 ZWRK-MAPELE OCCURS ***n*** TIMES.

07 ZWRK-MAPGIR PIC 9(02).

07 ZWRK-MAPMAX PIC 9(02).

07 ZWRK-MAPFLG PIC 9(01).

07 ZWRK-MAPINV PIC 9(01).

07 ZWRK-MAPTIP PIC 9(01).

07 ZWRK-MAPTIM PIC 9(04).

07 ZWRK-MAPMAS PIC 9(04).

07 ZWRK-MAPPG1 PIC 9(01).

07 ZWRK-MAPPG2 PIC 9(01).

07 ZWRK-MAPPG3 PIC 9(01).

07 ZWRK-MAPLNP PIC X(03).

07 ZWRK-MAPLQP PIC X(03).

07 ZWRK-MAPUNI PIC 9(04).

\* +------------------------------------------------------------+

\* + DATI DEL SISTEMA DA MANTENERE FRA UN INVIO E L' ALTRO +

\* +------------------------------------------------------------+

01 ZWRKSEG PIC S9(04) COMP VALUE **+*k***. (Dimensioni dell'area di appoggio)

01 ZWRK.

03 ZWRKELE PIC X(500) OCCURS ***k***. (Area di appoggio per le mappe)

01 FILLER REDEFINES ZWRK.

\* +------------------------------------------------------------+

\* + QUI DI SEGUITO VANNO MESSI I MESSAGGI VIDEO UTENTE +

\* +------------------------------------------------------------+

COPY ZZM***yyi***W. (Struttura dati della mappa ***i***)

***<eventuali altri dati (di livello >= 03) da mantenere fra un invio e l'altro>***

\* +------------------------------------------------------------+

\* + COPY DI WORKING UTENTE +

\* +------------------------------------------------------------+

COPY ZZM***yyi***D. (Rappresentazione della mappa ***i*** - da ripetere per tutte le mappe)

...

COPY ZZS***xxx***R. (Struttura del tracciato ***xxx*** - da ripetere per tutti i tracciati)

...

***<eventuali altri dati o COPY di Woking Storage>***

\* +------------------------------------------------------------+

\* + AREE PER LE LINK CON LE ROUTINE DELLA PROCEDURA +

\* +------------------------------------------------------------+

COPY ZZWLNK. (Area per la comunicazione fra programmi)

COPY ZZWIOI. (Dati per le routine di accesso agli archivi)

COPY ZZWCEM. (Dati per le routine di controllo e edit messaggio)

COPY ZZWDAT1. (Area per la gestione di data ora)

\* +------------------------------------------------------------+

\* + \*\*\* NON TOCCARE \*\*\* COPY WORKING SISTEMA +

\* +------------------------------------------------------------+

COPY ZZWMAP. (Ridefinizione standard messaggi video)

COPY ZZSAPPR. (Struttura dell'archivio APP)

COPY ZZWDAT2. (Area di lavoro per data e ora)

COPY ZZWCOST. (Variabili e costanti ad uso del sistema)

\* +------------------------------------------------------------+

\* + LINKAGE SECTION +

\* +------------------------------------------------------------+

LINKAGE SECTION.

COPY ZZWPTL. (LINK o CHG communication area - PCB di I/O - solo per DL1)

COPY ZZWAPPA. (PCB archivio APP lettura e scrittura - solo per DL1)

COPY ZZWAPP1. (PCB archivio APP solo lettura - solo per DL1)

***<eventuali altri PCB per altri archivi>***

\* +------------------------------------------------------------+

\* + PROCEDURE DIVISION +

\* +------------------------------------------------------------+

PROCEDURE DIVISION.

COPY ZZCPTI. (entry point DLITCBL per TP main - solo DL1)

COPY ZZCAPPA. (PCB archivio APP lettura e scrittura - solo per DL1)

COPY ZZCAPP1. (PCB archivio APP solo lettura - solo per DL1)

***<eventuali altri PCB per altri archivi>***

COPY ZZCPTP. (FINE PCB - solo per DL1)

\* +------------------------------------------------------------+

\* + \*\*\* NON TOCCARE \*\*\* COPY PROCEDURE SISTEMA +

\* +------------------------------------------------------------+

COPY ZZCPT. (Controllo generale)

COPY ZZCLNKT. (Link standard per programmi TP)

COPY ZZCCHGT. (Cambio programma TP)

COPY ZZCMAPIO. (Gestione ricezione mappe)

COPY ZZCDATT. (Richiamo routine di gestione delle date)

COPY ZZCIOIT. (Smistamento richiesta accesso archivio su relativa routine)

COPY ZZCABET. (Provoca ABEND del programma TP)

COPY ZZCDAT. (Section di elaborazione data e ora)

COPY ZZCSTOPT. (FINE del programma TP)

COPY ZZLAPP1T. (Operazioni di I/O sull'archivio APP - indice 1)

COPY ZZZDAT. (Azzeramento campi del tracciato usato per elaborazione data e ora)

\* +--------------------------------------------------------------+

\* + COPY DI PROCEDURE UTENTE +

\* +--------------------------------------------------------------+

COPY ZZZ***xxx***T. (Azzeramento del tracciato ***xxx*** - da ripetere per tutti i tracciati)

...

COPY ZZL***xxx***1T. (COPY di accesso all'archivio ***xxx*** - da ripetere per tutti gli archivi)

...

***<eventuali altre COPY di Procedure Division>***

### Section di Procedure division

La sezione principale di controllo di un programma TP è svolta dal codice contenuto nel file di COPY standard **zzCPT**: essa ha il compito di:

controllare i tasti funzionali

gestire i campi standard, posti nelle prime 4 righe di ogni mappa video ed in particolare richiamare il server per il cambio di transazione

coordinare l'interazione fra operatore e programma

Ci si dovrà quindi concentrare sugli aspetti specifici del programma ed in particolare:

pianificare l'organizzazione delle mappe video (numero, ciclicità, sequenzialità, contenuti, ecc.)

predisporre gli algoritmi per i controlli e le elaborazioni dei dati, gli aggiornamenti degli archivi, le stampe

Le porzioni di codice che svolgono i compiti indicati al secondo punto non possono essere scritte arbitrariamente ma devono essere predisposte in modo da rispettare i criteri strutturali richiesti dalla COPY zzCPT.

Essa infatti si aspetta che operazioni quali la preparazione dei campi di mappa, i controlli dei valori introdotti dall'operatore, gli aggiornamenti, siano svolte da apposite section standard del programma.

Analizzeremo ora il flusso di un programma TP e l'ordine di chiamata di queste section standard.

Quando un programma TP parte, prende subito il controllo il codice della COPY zzCPT, che effettua le operazione illustrate in figura 1; la parte "Controlli", che si occupa di verificare la validità dei dati introdotti è dettagliata in figura 2.

Le procedure indicate nei due schemi e riportate di seguito corrispondono ad altrettante section di procedure division del programma:

zprepara

zformat-wrk

zformat-working

zcntl

zcntl-all

zaggiorna

La inizializzazione delle variabili per la gestione delle mappe comporta l'assegnazione dei valori iniziali a

ZMAPNUM (numero della mappa corrente)

ZMAPMAS (contatore del numero massimo di mappe per unità o di pagine per una mappa ciclica)

ZUNINUM (numero della unità corrente)

ZUNIMAS (contatore del numero massimo di unità)

ZMAPMSG (mappa del video)

ZWRK-MAPTAB (flag di mappa, copiati dai valori iniziali di ZMAPTAB)



Figura 1:   Flusso di un programma TP



Figura 2:   Sezione di controllo dei valori digitati a mappa

La sequenza delle attività e degli scambi fra la copy zzCPT e programma che la include è riportata nella tabella seguente:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **COPY zzCPT** | |  | **Programma** | |
| **Section** | **Attività** |  | **Section** | **Attività** |
|  | Il TP monitor cede il controllo |  |  |  |
| zreceive | ricezione della mappa nella variabile zmapmsg |  |  |  |
| zformat | inizializzazione di variabili |  |  |  |
| zaggiorna-variabili | lettura dal file di appoggio dei dati di apertura sessione  (tipo appoggio = ZAPPTER)  se non ci sono, chiamata alla routine per apertura sessione, altrimenti aggiornamento delle relative variabili (sessioni attive, lingua, operatore, ecc.) |  |  |  |
| zdata-ora-oggi | Lettura della data e dell'ora  (tramite la routine zzRTDO2) |  |  |  |
| zlettura-dati | Lettura dal file appoggio dei dati (tipo appoggio = ZAPPDAT) |  |  |  |
|  | Controllo recovery e controllo errore nel cambio transazione |  |  |  |
| zreset | Se è la prima volta, inizializzo le variabili per la gestione delle mappe |  | zprepara | prepara i campi di mappa da presentare precaricati |
|  |  |  | zformat-wrk | Azzeramento dei campi del file di Appoggio |
|  |  |  | zformat-working | Azzeramento dei campi di Working Storage |
| zcntl-io | Controllo / edit campi del messaggio  (tramite la routine zzRTCE) |  |  |  |
|  | Controllo eventuale pressione di tasti funzionali |  |  |  |
|  | Eventuale accessi al server di paginazione |  |  |  |
|  | Se l'utente non ha indicato funzione (sessione), viene impostata la funzione (sessione) attiva |  |  |  |
|  | Confronta funzione (sessione) con quella attiva ed eventualmente cambia chiamando il server di cambio transazione |  |  |  |
| zlettura-dati | Lettura dal file appoggio dei dati (tipo appoggio = ZAPPWRK) |  |  |  |
|  |  |  | zmovinp | copiatura del messaggio contenente la mappa (ZMAPMSG) in area di lavoro ZZM**yyi** |
| zstart-cntl | Controlli della singola mappa  Se tutto è OK, passa al prossimo giro di mappa |  | zcntl | Esecuzione della section specifica per il controllo del valore dei campi della mappa e del giro di mappa attuali |
| zstart-cntl-all | Controlli incrociati |  | zcntl-all | Verifica della presenza di errori incrociati sui valori dei campi delle varie mappe |
| zreset | Inizializzo le variabili per la gestione delle mappe |  | zprepara | prepara i campi di mappa da presentare precaricati |
|  |  |  | zformat-wrk | Azzeramento dei campi del file di Appoggio |
| zaggiorna-mappa | Lettura, modifica e aggiornamento dei dati di mappa sul file appoggio (tipo appoggio = ZAPPMAP) |  | zmovout | copiatura dell'area di lavoro ZZM**yyi**  nel messaggio contenente la mappa (ZMAPMSG) |
| zaggiorna-wrk | Lettura, modifica e aggiornamento dei dati di lavoro sul file appoggio (tipo appoggio = ZAPPWRK) |  |  |  |
| zaggiorna-mappa | Lettura, modifica e aggiornamento dei dati di apertura sessione sul file appoggio  (tipo appoggio = ZAPPTER) |  |  |  |
| zaggiorna-dati | Lettura, modifica e aggiornamento dei dati sul file appoggio  (tipo appoggio = ZAPPDAT) |  |  |  |
| zedit-io | Predisposizione dei campi di mappa relativi alle prime 4 righe standard |  | zmovdis | Copiatura del disegno di mappa nel tracciato di edit messaggio |
| zsend | Invio della mappa al TP monitor |  |  |  |
| zstop | Terminazione del programma TP e cessione del controllo al TP monitor |  |  |  |

#### Descrizione delle section standard

Analizziamo ora le section standard di Procedure Division che vengono invocate dalla copy zzCPT e che devono essere presenti in ogni programma TP. Esse sono:

ZFORMAT-WORKING Inizializza i dati di Working Storage del programma

ZFORMAT-WRK Inizializza i dati sul file di appoggio

ZPREPARA Prepara gli eventuali dati da presentare pronti all'inizio sulla mappa

ZMOVDIS Copia il disegno delle mappe nel tracciato per edit del messaggio

ZMOVINP Copia il messaggio standard nelle copy delle mappe

ZMOVOUT Copia la copy delle mappe nel messaggio standard

ZCNTL Effettua i necessari controlli sui valori dei campi di una mappa

M01-G01-CNTL

M01-G02-CNTL

M02-G01-CNTL

...

ZCNTL-ALL Effettua i controlli incrociati fra valori di campi di mappe diverse

ZAGGIORNA Aggiorna il contenuto dell'archivio al termine del programma di gestione

Inoltre molto spesso vengono definite le seguenti section extra:

PRECARICA-MAPPA-*i* Precarica la mappa *i* con i valori di default

VALORIZZA-MAPPA-*i* Carica la mappa *i* con i dati letti da archivio

CARICA-ZZS*xxx*R Carica il tracciato dell'archivio *xxx* con i valori dei campi di mappa

PROTEGGI-MAPPA Protegge i campi della mappa in modo che l'operatore non li possa modificare

# Navigazione

Per semplificare la manutenzione dei programmi e migliorare la flessibilità di una procedura è opportuno, e talvolta necessario, sezionare programmi creati per le grosse gestioni di dati in tante piccole transazioni da collegare in una sequenza logica detta *schema di navigazione.*

## Caratteristiche della Navigazione

Le caratteristiche della navigazione sono:

la sequenza di navigazione può variare in funzione di condizioni che si verificano durante l'esecuzione di una delle transazioni;

si può eseguire la sequenza di navigazione all'indietro, di una transazione alla volta, oppure tornare subito alla prima transazione della sequenza;

si possono eseguire in sequenza fino ad un massimo di 25 transazioni; è possibile comunque aumentare il numero massimo di transazioni di una sequenza modificando alcune copy di sistema;

se si superano le 25 transazioni si può continuare la sequenza; le transazioni eseguite dopo la venticinquesima non vengono però memorizzate perdendone quindi ogni riferimento;

esiste la possibilità di passare un'area dati su appoggio da una transazione all'altra;

il passaggio di una transazione all'altra deve essere gestito all'interno delle transazioni di navigazione mediante le funzioni disponibili;

si può decidere se attivare o disattivare i tasti di navigazione settando a 'N' o 'P' il relativo flag.

## Schema di funzionamento della navigazione

Si riporta di seguito lo schema di funzionamento della navigazione:



Figura 1: schema di funzionamento della navigazione

T1, T2, T3, T4 sono le transazioni che compongono lo schema della navigazione; app T2, app T3, app T4 sono i dati su appoggio (APP) relativi alle transazioni T2, T3, T4;

Trovandosi nella transazione T3, si può:

passare alla transazione successiva T4 mediante la chiamata ZNAV-AVANTI;

tornare alla transazione precedente T2 mediante la chiamata ZNAV-INDIETRO;

tornare alla prima transazione T1 dello schema mediante la chiamata ZNAV-INDIETRO-PRIMO;

salvare su appoggio i dati relativi alla transazione corrente T3 mediante la chiamata   
ZAGGIORNA-APPNAV-ATT;

aggiornare su appoggio i dati relativi alla transazione precedente T2 mediante la chiamata   
ZAGGIORNA-APPNAV-PREC;

salvare su appoggio i dati relativi alla transazione successiva T4 mediante la chiamata  
ZAGGIORNA-APPNAV-NEXT;

eggere da appoggio i dati relativi alla transazione corrente T3 mediante la chiamata  
ZLETTURA-APPNAV-ATT;

leggere da appoggio i dati relativi alla transazione precedente T2 mediante la chiamata  
ZLETTURA-APPNAV-PREC;

## Come si utilizza la navigazione

Le funzioni (definite nella copy ZZCNAV) che vengono offerte per la navigazione sono:

|  |  |
| --- | --- |
| ZLETTURA-APPNAV-PREC | per leggere i dati della transazione precedente |
| ZLETTURA-APPNAV-ATT | per leggere i dati della transazione attuale |
| ZAGGIORNA-APPNAV-PREC | per aggiornare i dati della transazione precedente |
| ZAGGIORNA-APPNAV-ATT | per aggiornare i dati della transazione attuale |
| ZAGGIORNA-APPNAV-NEXT | per aggiornare i dati della transazione successiva |
| ZNAV-AVANTI | per passare alla transazione successiva |
| ZNAV-INDIETRO | per tornare alla transazione precedente |
| ZNAV-INDIETRO-INIZIO | per tornare alla prima transazione della sequenza |
| ZTASTI-NAVIGAZIONE | per eseguire i controlli e le operazioni associate ai tasti di navigazione attivi |

Il nome della transazione successiva da richiamare va impostato nella variabile ZNXTTRX prima di effettuare la chiamata alla section ZNAV-AVANTI.

I tasti di navigazione di default sono:

PF2 per annullare la navigazione;

PF3 per tornare alla transazione precedente;

PF4 per tornare alla prima transazione dello schema.

E' possibile comunque impostare dei tasti di navigazione personalizzati modificando la section ZATTIVA-TASTI nella copy ZZCNAV.

Per attivare un tasto di navigazione è necessario valorizzare con 'N' il corrispondente elemento dell'array ZPFKPGM-CGES.

Di solito l'attivazione dei tasti di navigazione viene effettuata nella section ZFORMAT-WORKING.

# Gestione delle mappe video

Di seguito viene descritta la gestione della paginazione nei programmi TP secondo lo standard CITY.

## Le mappe video

I programmi applicativi on-line sviluppati in ambiente HOST necessitano di particolari strutture atte a rendere operativamente più agevole la interazione fra display e operatore: tali strutture in CICS sono chiamate *mappe*.

Il CICS mette a disposizione un insieme di moduli per la loro gestione che costituiscono il Basic Mapping Support (BMS);

Su HOST una mappa è costituita da un set di macro in assembler da cui un apposito compilatore genera una *mappa fisica* e una *mappa logica*.

La mappa fisica è paragonabile ad un programma che definisce un'area di memoria destinata a racchiudere i dati da visualizzare.

Tali dati sono contenuti nella mappa logica che, a sua volta, deve essere inclusa nel programma applicativo che la utilizzerà.

E' cura dei moduli di gestione delle mappe del monitor TP (BMS per CICS) valorizzare la mappa fisica con i dati della mappa logica creando così la mappa video.

Per le applicazioni PC/LAN interne, la creazione del file MASM (macro assembler) è preceduta dalla compilazione del *disegno di mappa* che, oltre alla mappa fisica e logica, produce una mappa di descrizione simbolica, cioè una struttura dati in linguaggio sorgente che il compilatore usa per risolvere i riferimenti, contenuti nel programma applicativo, ai campi della mappa.

Lo schema seguente illustra il concetto appena espresso:



Figura 3:   Passi per la realizzazione di una mappa

## Utilizzo di un meccanismo di paginazione standard

Le primitive fondamentali del CICS quali la SEND MAP (che espone a video i dati) e la RECEIVE MAP (che mappa i dati del terminale in un area del programma applicativo), prese singolarmente non permettono una soluzione ottimale delle problematiche di interazione video/utente/transazione.

Il CICS mette a disposizione dei comandi di paginazione (Terminal Operator Paging Commands) a fronte di particolari opzioni come PAGING e ACCUM da specificare all'istruzione SEND, però se per esigenze operative si deve premere il tasto di INVIO il CICS abbandona il controllo della transazione: delle mappe associate non resterà quindi alcuna traccia in memoria.

La perdita di tutti i dati ad ogni pressione del tasto INVIO fa sì che non sia possibile permettere all'utente di rivedere una mappa precedentemente esposta a video senza doverla *ricreare da capo*: ciò comporterebbe la necessità di realizzare programmi complessi, perché dovrebbero controllare da soli la visualizzazione delle mappe video.

Per evitare di sviluppare ogni volta routine di controllo delle mappe e renderne più uniforme, coerente e flessibile possibile la gestione, è stato preparato un insieme di strumenti standard che permettono di gestire problematiche variamente articolate.

## La paginazione standard CITY

La metodologia di gestione della paginazione in standard CITY può essere sintetizzata nei seguenti punti:

le proprietà di ogni mappa vengono indicate mediante un insieme di flag, raccolti in una tabella;

una stessa mappa è gestita in modo diverso in funzione dei valori dei suoi flag

i programmi devono predisporre l'organizzazione delle mappe ma **non** si devono curare di gestire direttamente la loro rappresentazione sul video: a questo pensano opportune routine della CITY;

le istanze di ogni mappa, una volta create non vengono perse ma memorizzate in un archivio di appoggio, gestito dalla CITY: questo permette di muoversi aventi ed indietro fra le mappe, ritornando ad esempio su videate precedenti;

La gestione della paginazione in standard CITY è affidata ad una COPY standard di Procedure Division (zzCPT) che, oltre a svolgere varie attività di base (è il "motore" della paginazione), mette a disposizione alcune primitive per il controllo delle mappe.

### Concetti base: ciclo, mappa, unità

Gli strumenti offerti dallo standard CITY permettono di diversificare il comportamento del programma nei confronti delle sue mappe specificando per ciascuna di esse un insieme di parametri (flag di mappa), del cui dettaglio ci occuperemo in un paragrafo successivo.

L'insieme di tutti i parametri di una mappa è raccolto in una tabella decisionale, che viene utilizzata per la gestione dell'evoluzione degli stati della mappa stessa a partire da uno stato iniziale.

Occorre a questo punto illustrare alcuni concetti che conducono alla definizione della tabella decisionale di cui sopra:

|  |  |
| --- | --- |
| CICLO | ripetizione della stessa mappa con dati diversi  Quando è necessario esporre a video più volte la stessa mappa, ad esempio per permettere l'inserimento di dati ripetuti o per mostrare elenchi che sono troppo lunghi per stare in un'unica videata, si definisce un *ciclo* di mappa: la mappa fisicamente utilizzata è una sola ma ne vengono memorizzate diverse istanze (pagine) ciascuna con i propri valori dei campi. |
| MAPPA | corrisponde ad una mappa fisica ed è collegata ad una tabella decisionale; |
| UNITA' | insieme di mappe raggruppate sulla base di particolari opzioni  L'ordine con cui dovranno essere mostrate le mappe e le loro proprietà, come il fatto di essere cicliche, sono stabilite *a priori*.  Può essere però necessario mostrare le mappe in modo diverso (ad esempio saltandone alcune) a fronte di particolari richieste dell'utente: in questi casi bisogna predisporre non una ma *diverse* organizzazioni delle mappe, e durante l'uso del programma stesso si sceglierà quale adottare.  Ogni metodo di organizzazione è chiamato *unità:* le mappe, pur avendo lo stesso disegno, possono apparire con diverse caratteristiche di ciclicità fra un'unità e un'altra oppure possono non apparire affatto in certe unità e comparire invece su altre. |

In breve:

Ad ogni mappa è associata una tabella decisionale, costituita da un insieme di parametri (flag) che ne determina le proprietà;

Tutte le tabelle decisionali sono riunite in un'unica grande tabella che determina l'organizzazione delle mappe e individua una *unità*: cambiare unità significa cambiare questa tabella e quindi modificare le proprietà di ciascuna mappa, cioè il modo in cui sarà gestita;

Ogni *istanza di mappa* può essere indicata mediante tre coordinate:

- il numero di pagina (per mappe non cicliche è sempre 1)

- il numero di mappa (che stabilisce a quale mappa fisica corrisponde)

- il numero di unità (che determina la tabella dei dati corrispondente)

Nel diagramma che segue sono schematizzati questi concetti; le mappe fisiche del programma sono tante quante le colonne, le unità tante quante le righe.



Figura 4:   Paginazione: ciclo, mappa, unità

E' possibile dedicare alcuni tasti funzionali al movimento fra le varie istanze di mappa prodotte da un programma lungo ciascuna delle tre coordinate indicate; si parlerà allora di

|  |  |
| --- | --- |
| PAGINAZIONE SU CICLO | La stessa mappa (ciclica) si ripete *n* volte con dati diversi e si gestisce la possibilità di scorrere le *n* pagine; |
| PAGINAZIONE SU MAPPA | La transazione ha prodotto *n* mappe diverse e si gestisce la possibilità di passare dalla mappa corrente alla precedente o alla successiva; |
| PAGINAZIONE SU UNITA' | La transazione permette, ad esempio a partire da una maschera iniziale di selezione, di scegliere diverse serie di mappe che posson essere raggruppate in unità logiche distinte e si gestisce la possibilità di passare da una unità corrente alla successiva. |

La CITY prevede, due tipi di paginazione che fanno riferimento ai tasti funzionali PF7/PF8 (paginazione primaria) e PF5/PF6 (paginazione secondaria) e che possono essere assegnati valorizzando opportuni flag, come spiegato nel paragrafo successivo.

Nella figura seguente è mostrato un esempio di paginazione con due unità: se l'utente, inserendo "X" sulla prima mappa attiverà la prima unità, la mappa M3 non sarà visibile in quanto il suo flag di stato la indica come "da non trattare". Invece se l'utente, inserendo "Y" sulla prima mappa attiverà la seconda unità, la mappa M3 sarà disponibile e verrà visualizzata. Inoltre solo nel secondo caso la mappa M2 sarà ciclica, cioè verrà presentata più volte.

Non c'è invece modo di alterare l'ordine delle mappe nelle due unità.

NOTA: nella figura 5 la prima mappa, comune alle due unità, è rappresentata in modo diverso; ciò è dovuto

solo a motivi di chiarezza, è ovvio che le mappe del programma (e quindi anche le righe della tabella) sono 5.



Figura 5:   Esempio di paginazione su unità

### Il file di appoggio

Come indicato precedentemente, le mappe create dai programmi TP presenti sulle varie sessioni di lavoro di ciascun operatore vengono salvate dalla CITY su un file di appoggio (EITAPPR), pronte per essere recuperate e mostrate in istanti successivi.

Ogni accesso all'archivio APP comporta inserimenti od aggiornamenti ma non cancellazioni (per evitare la perdita di dati potenzialmente utili e ridurre i tempi necessari alla riorganizzazione che questa operazione comporta), di conseguenza è possibile che, in funzione del carico di lavoro del sistema, esso possa crescere fino ad assumere dimensioni significative.

(In particolare ciò può avvenire quando viene lanciata una operazione occasionale di inquiry che produce un elevato numero di mappe video: il sistema predispone una grande quantità di spazio sull'area di lavoro che rimarrà pressoché inutilizzata in seguito. E' stato comunque predisposto un programma Batch per la riorganizzazione periodica di questo archivio)

Oltre alle mappe, sul file APP possono essere salvate dalla CITY anche le tabelle con i dati di mappa (relativi alle diverse unità) ed altri dati di servizio: per differenziare il contenuto in funzione del tipo, i vari record hanno un campo di chiave "Tipo appoggio", che nel caso delle mappe vale 3 e nel caso delle tabelle dei dati di mappa vale 4.

**NOTA**: Poichè lo spazio occupato da tutti i dati di tutte le mappe deve stare su un unico record del file appoggio (le cui dimensioni sono fissate), il massimo numero di mappe gestibili è stato fissato a **50**.

### Gestione delle mappe

Dopo aver predisposto le mappe su cui il programma TP dovrà operare e aver indicato le proprietà di ciascuna mappa mediante i flag della relativa tabella, è necessario fare in modo che il programma gestisca *operativamente* le mappe rispettando l'organizzazione che si era prevista in fase di disegno.

Per questo scopo sono state predisposte alcune primitive (section del file di COPY zzCPT) che realizzano le operazioni di base per la gestione della paginazione; esse sono state pensate per cercare di semplificare la gestione della paginazione e permettono di svolgere con semplicità le operazioni basilari nella maggior parte delle situazioni.

#### Elenco delle primitive di controllo della paginazione

Le funzioni predisposte in ambiente CITY (section della COPY zzCPT) per gestire la paginazione sono:

|  |  |
| --- | --- |
| ZAGGIORNA-MAPPA | Salva la mappa corrente nel file di appoggio. |
| ZRESTORE-MAPPA | Ripristina una mappa precedentemente memorizzata sul file di appoggio. |
| ZRESTORE-UNITA | Ripristina una unità precedentemente memorizzata sul file di appoggio. |
| ZNEXT-MAPPA  ZNEXT-CICLO  ZNEXT-UNITA' | Salvano la mappa, la pagina o l'unità corrente nel file di appoggio e predispongono il programma applicativo al trattamento della prossima mappa o pagina od unità. |
| ZNEXT-GIRO-MAPPA | Passa al giro successivo della mappa corrente |
| ZFINE-PAGING | Termina la paginazione salvando la mappa corrente nel file di appoggio e impostando l'inizio della paginazione sulla prima mappa avente ZMAPTIP = 1. |

# Accesso a base dati tramite standard CITY

Nello standard CITY l'accesso alle basi dati (archivi) avviene sempre tramite appositi *moduli di accesso*; essi permettono di effettuare le operazioni di ricerca, inserimento, aggiornamento e cancellazione di elementi della base dati secondo il modello relazionale, nascondendo i dettagli tecnici di tali operazioni.

I *moduli per l'accesso* ad un archivio sono:

1. copy di accesso, contenenti la definizione delle primitive virtuali di accesso;

2. copy di smistamento, che realizza l'associazione record-archivio (per gli ambienti operativi che lo richiedono);

3. Routine di accesso, che effettuano le operazioni di accesso vere e proprie.

## copy di accesso

Sono definite, per ogni tracciato record, all'interno di opportune copy di Procedure Division, dette *copy di accesso*.

Esiste dunque una section per ogni operazione che si vuole effettuare su un dato elemento (e per ogni indice da usare per l'accesso): in questo modo è possibile intercettare subito alcuni errori nella stesura dei programmi, perchè il compilatore si fermerebbe trovando la chiamata ad una section non definita.

Ad esempio, se un programma contiene per errore l'istruzione PERFORM ZINSERT-ZZSABC2, il compilatore segnalerà che la relativa section non esiste: la copy di accesso infatti *non deve* contenere la dichiarazione di tale operazione perché non è possibile inserire un elemento per indice secondario.

I file di copy per i programmi TP (contententi la definizione delle primitive ZFIND, ZINSERT, ZUPDATE, ZDELETE, ZRANGE, e ZRANGE) si chiamano, secondo le convenzioni standard dei nomi:

zz**L** *<nome-tracciato> <numero-indice>* **T**;

I file di copy per i programm BATCH (contententi la definizione delle primitive ZOPEN-INPUT, ZOPEN-UPDATE e ZCLOSE) si chiamano:

zz**L** *<nome-tracciato> <numero-indice>* **D**;

**Esempio**

La copy TP che contiene la definizione delle primitive di accesso all'elemento "ABC" mediante chiave primaria si chiamerà **zzLABC1T**.

La copy BATCH che contiene la definizione delle primitive di OPEN-INPUT, OPEN-UPDATE e CLOSE per il medesimo elemento si chiamerà **zzLABC1D**.

**NOTA**: anche quando, in funzione della base dati, una copy non servirebbe, deve essere definita lo stesso (vuota) per garantire la completa compatibilità dei diversi ambienti operativi.

### Primitive Virtuali Per L'accesso Ai Dati

Come precedentemente osservato, ogni programma sviluppato secondo le standard CITY si avvale di un insieme di primitive[[3]](#footnote-3) per accedere ad una base dati che sono indipendenti dall'ambiente di sviluppo del programma stesso (TP monitor, DBMS); esse sono:

Primitive di accesso in lettura

ZFIND ricerca di un preciso elemento nella base dati;

ZRANGE impostazione di un sottoinsieme degli elementi della base dati (indicando i limiti di un'intervallo di ricerca per i valori di alcuni attributi) e selezione del primo elemento che vi appartiene (se esiste);

ZNEXT selezione dalla base dati del prossimo elemento appartenente al sottoinsieme specificato con la relativa operazione ZRANGE.

Primitive di accesso in scrittura

ZINSERT inserimento di un elemento (tupla, record) nella base dati;

ZUPDATE aggiornamento di un elemento nella base dati;

ZDELETE cancellazione di un elemento dalla base dati;

La sintassi generale della chiamata a queste primitive è:

**PERFORM** *<operazione>*-*<nome-tracciato><numero-indice>*.

Ad esempio l'operazione di inserimento del tracciato ABC nella base dati viene chiamata tramite il comando:

PERFORM ZINSERT-ZZSABC1

In aggiunta a quelle sopra esposte, nei programmi BATCH esistono altre primitive, per la gestione dei file:

Primitive di accesso per apertura-chiusura di file

ZOPEN-INPUT apre in input il file indicato

ZOPEN-UPDATE apre in update il file indicato

ZCLOSE chiude il file indicato

La sintassi generale della chiamata a queste primitive è:

**PERFORM** *<operazione>*-*<nome-archivio><numero-indice>*.

Ad esempio l'operazione di apertura per inserimento del file XXX viene chiamata tramite il comando:

PERFORM ZOPEN-INPUT-ZZSXXX1

Viene fatta una distinzione fra le operazioni possibili a seconda che impieghino l'indice primario o un indice secondario:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | **Operazione** | **Indice primario** | **Indici secondari** | **Note** |
|  |  | ZFIND | SI | SI\* | \* purché l'indice non sia un riordino dell'indice primario |
| Lettura |  | ZRANGE | SI | SI |  |
|  |  | ZNEXT | SI | SI |  |
|  |  | ZINSERT | SI | NO |  |
| Scrittura |  | ZUPDATE | SI\*\* | NO | \*\* purché non si aggiornino campi dell'indice primario |
|  |  | ZDELETE | SI | NO |  |

**NOTA**: il funzionamento delle primitive è il medesimo in caso di successo per tutte le basi dati; *in caso di errore* si possono avere segnalazioni differenti, anche se comunque il programma termina in abend.

## copy di smistamento

Le primitive di accesso a base dati, definite nelle copy di accesso, fanno riferimento ad un'altra copy di Procedure Division detta *copy di smistamento*; essa ha il compito di associare il tracciato indicato nell'operazione di accesso con la relativa base dati.

Quando un programma chiama una delle section COBOL viste sopra, questa cede il controllo alla copy di smistamento la quale chiama dinamicamente la routine che si occupa dell'accesso alla base dati per l'elemento indicato.

Esistono sempre **due** copy di smistamento:

una impiegata dai programmi TP, che si chiama **zzCIOIT**

una impiegata dai programmi BATCH, che si chiama **zzCIOID**

Queste due copy variano sensibilimente al variare del tipo di base dati utilizzata (addirittura nel caso del DL/I non è neppure utilizzata): si rimanda dunque al capitolo 3 "Dettagli sull'accesso dipendenti dalla base dati" per informazioni specifiche.

La scelta architetturale di avere un passaggio intermedio fra copy di lettura e routine attraverso la copy di smistamento è stata determinata dal vantaggio di avere un punto unico di chiamata delle routine stesse, con la possibilità di dirottare verso eventuali altre routine esterne le operazioni relative a particolari tracciati; inoltre in questo modo le copy di accesso vengono notevolmente semplificate.

Purtroppo non sempre questo approccio è possibile, infatti nel caso del DL/I la chiamata delle routine avviene senza passare per la copy di smistamento, come verrà indicato nel capitolo successivo.

## Le routine di accesso

Le operazioni di accesso vero e proprio alla base dati vengono realizzate operativamente da routine COBOL, che sono chiamate dinamicamente durante l'esecuzione del programma che richiede l'accesso.

Il numero di routine dipende dalla base dati su cui si opera ed è determinato da criteri di semplicità ed efficienza:

semplicità si cerca di isolare il codice nelle routine, in modo da avere una struttura modulare;

efficienza operando con il DB2 si sono tenute separate le routine di lettura da quelle di scrittura: le prime infatti non richiedono un accesso esclusivo alla base dati e dunque non ne bloccano l'accesso da parte di altri programmi in esecuzione concorrente. Le altre invece necessitano dell'uso esclusivo della risorsa condivisa, per evitare di operare simultaneamente sugli stessi dati generando situazioni di inconsistenza.

Le routine chiamate dai programmi TP si chiamano zz**QT** *<nome-tracciato> <numero-indice>*;

Le routine chiamate dai programmi BATCH si chiamano zz**QD** *<nome-tracciato> <numero-indice>*.

La struttura ed il contenuto di queste routine dipendono pesantemente dal tipo di base dati utilizzata, in particolare per quanto riguarda le modalità di passaggio dei parametri, ma per una stessa base dati sono in generale piuttosto ripetitivi (si veda più avanti).

Anche se le routine di accesso per un determinato tipo di base dati sono più o meno tutte uguali, possono esistere delle necessità particolari, determinate dal programma o dalla procedura, per cui una routine debba essere sviluppata secondo criteri personalizzati.

È possibile in tal caso uscire dagli schemi standard e realizzare routine esterne la cui organizzazione sia anche molto diversa da quanto indicato in questo documento.

### DB2

Il DB2 è un DBMS relazionale, dunque rispetta in pieno tutte le considerazioni teoriche fatte nella prima parte del documento; le complesse operazioni di confronto fra campi di chiave non sono necessarie in quanto la correttezza nell'accesso è garantita dal DB2 stesso.

L'organizzazione dei moduli di accesso agli archivi nel caso DB2 è rappresentato nello schemi che segue.



Figura 7:   Schema di organizzazione delle chiamate da un programma TP per l'accesso ad una base dati DB2

#### copy di accesso

Le copy di accesso per i programmi TP si limitano a definire le section corrispondenti alle diverse primitive di accesso ad un dato archivio, richiamando alla fine la section principale della copy di smistamento (valorizzando i campi di sistema ZIOIFUN e ZIOIIND rispettivamente con il codice dell'operazione scelta e la chiave di accesso).

Le copy di accesso per i programmi BATCH sono vuote.

#### copy di smistamento

Le due copy di smistamento (una per i TP, una per i BATCH) hanno il compito di chiamare dinamicamente la routine giusta in funzione del nome dell'elemento da manipolare e del tipo di operazione da compiere

(nei file Batch il prefisso delle routine di accesso diventa **zzQD** anziché **zzQT**).

|  |  |  |  |
| --- | --- | --- | --- |
| **COPY smistamento** | **Tipo operazione** | **Nome operazione** | **Routine chiamata** |
| **zzCIOIT** | Lettura | ZFIND  ZRANGE  ZNEXT | **zzQT** *<nome-elemento> <numero-indice>* |
| (TP) | Inserimento  Aggiornamento  Cancellazione | ZINSERT  ZUPDATE  ZDELETE | **zzQT** *<nome-elemento>* **A** |
| **zzCIOID** | Lettura | ZFIND  ZRANGE  ZNEXT | **zzQD** *<nome-elemento> <numero-indice>* |
| (BATCH) | Inserimento  Aggiornamento  Cancellazione | ZINSERT  ZUPDATE  ZDELETE | **zzQD** *<nome-elemento>* **A** |

#### Routine di accesso

Per ogni tabella della base dati devono essere predisposte tante routine di accesso quanti sono gli indici previsti più una; le prime svolgono le operazioni di lettura sul relativo indice, l'altra svolge le operazioni di variazione della base dati (Scrittura / Aggiornamento / Cancellazione).

Le operazioni di accesso vengono effettuate tramite semplici chiamate al DB2:

EXEC SQL (SELECT, DELETE, FETCH, INSERT, OPEN, CLOSE, ecc.)

...

END EXEC

Si predispongono routine separate per programmi TP e programmi BATCH anche se sono del tutto uguali per i seguenti motivi:

in CICS le modalità di chiamata sono diverse (EXEC CICS per i TP, CALL per i BATCH)

# Gestione delle autorizzazioni

Quando c'è la necessità di gestire una grossa quantità di dati si pone il problema di limitare gli accessi su tali dati al solo personale direttamente interessato.

Una procedura è costituita di più transazioni; al suo interno ci possono essere delle sotto-procedure. È normale, quindi, che nasca l'esigenza di limitare l'utilizzo di una una o più transazioni solo a un gruppo di persone con un incarico ben preciso.

Una procedura in standard CITY dispone di una serie di transazioni di gestione e di inquiry dati per rispondere a queste esigenze.

## Gli elementi per la gestione delle autorizzazioni

Per far fronte al problema della gestione delle autorizzazioni sono state introdotte le seguenti entità:

moduli di sicurezza,

profili di autorizzazione

### Moduli di sicurezza

Si tratta di un'entità, identificata da un codice (lungo al massimo otto caratteri) al quale viene associato il controllo esclusivo su un singolo dato, o comunque su un gruppo di dati ben distinto.

Esempio:

*Si supponga di dover concedere all'operatore Mario Rossi l'abilitazione ad eseguire delle operazioni bancarie con data diversa da quella di oggi. E' necessario innanzi tutto preparare un modulo (che si chiamerà ad esempio "ABILDSES" - abilitazione a forzare la data della sessione di lavoro) cui corrisponda questa autorizzazione (si vedrà in seguito come fare).*

*Poi bisognerà inserire l'associazione "Mario Rossi / "ABILDSES"; la presenza di tale associazione indica la presenza dell'autorizzazione. Se l'associazione non è presente l'autorizzazione non sarà concessa.*

Ad uno stesso operatore è possibile associare più moduli di sicurezza (anche nessuno).

L'autorizzazione all'*accesso ai dati* avviene sempre tramite i moduli di sicurezza.

### Profili di sicurezza

Un profilo di sicurezza è un insieme di autorizzazioni di accesso a *dati* o *funzioni*, generalmente in relazione fra loro.

Esempio:

*La procedura Esattoria può comprendere una serie di programmi per la gestione di alcune operazioni di sportello. Si potranno raggruppare queste transazioni sotto un profilo chiamato "sportello". Gli operatori che dovranno essere abilitati all'utilizzo di questi programmi avranno associato il profilo "sportello" sull'archivio di gestione della sicurezza.*

Sarebbe sconveniente dover indicare esplicitamente per ogni operatore quali funzioni e quali dati egli può vedere, anche perché molto spesso l'accesso ad un dato o ad una funzione comporta automaticamente l'accesso ad altri dati e funzioni.

E` molto conveniente individuare degli insiemi di diritti di accesso e raccoglierli in altrettanti profili una volta per tutte.



Figura 12:   Schema dell'organizzazione degli elementi che realizzano il controllo dell'accesso

## Metodo di individuazione dei diritti d'accesso

Per determinare se un operatore ha o no i diritti di accesso ad un dato o ad una funzione è necessario, come si è visto, determinare la presenza dell'associazione operatore-modulo o operatore-funzione nell'archivio della sicurezza.

Quando un operatore si collega alla CITY, aprendo la prima sessione di lavoro, viene scandito l'insieme dei profili a lui associati e per ogni diritto di accesso (modulo o funzione) viene creata nell'archivio della sicurezza un'associazione diretta, marcata come "associazione automatica", fra l'operatore ed il relativo elemento (modulo o funzione).

Ciò produce una gran quantità di dati in archivio ma assicura ottime prestazioni al momento del controllo.

Per evitare di dover sovraccaricare inutilmente il sistema, questo "spacchettamento" di informazioni all'atto dell'apertura della prima sessione di lavoro viene svolto SOLO se dall'ultima volta sono cambiati i diritti di accesso per quell'operatore (o per più operatori, nel caso di profili).

## Sicurezza nell'accesso

E' possibile distinguere due tipologie di accesso:

1. Accesso a funzioni (ossia ai programmi di gestione, caricamento e visualizzazione dati), dette anche transazioni interattive.

2. Accesso ai dati (ossia controllo sull'inserimento, variazione e annullamento di un certo dato da parte di un certo operatore).

### Accesso a funzioni

Per passare di programma in programma lo standard CITY prevede un campo di 4 caratteri in alto a sinistra su ogni mappa sul quale può essere digitato un "codice funzione" associato ad una precisa transazione.

Il programma che riceverà questa informazione, provvederà a richiamare un programma che fa da server per il cambio transazione (zzPT97).

Il programma di cambio transazione, per prima cosa, andrà a verificare che il codice funzione sia stato precedentemente caricato sulle tabelle funzioni (tramite il programma di gestione transazioni "GTRX" zzPT26).

Se la ricerca avrà esito positivo verrà eseguito un controllo sull'autorizzazione dell'operatore a richiedere il programma associato alla funzione digitata tramite un'apposita routine (zzRTCA).

E' importante notare che il controllo sull'accesso ad una funzione è realizzato in modo completamente automatico dalla CITY, senza bisogno di alcuna modifica alla funzione stessa.



Figura 13:   Sequenza di attivazione di una funzione

### Accesso ai dati

In ogni procedura esiste il problema di controllare se un operatore è autorizzato o meno ad accedere a dei dati. Per accesso si intende una qualsiasi operazione di ricerca, inserimento, cancellazione o modifica del dato stesso.

La concessione o meno dell'autorizzazione viene effettuata attraverso l'utilizzo dei moduli di sicurezza.

La creazione di un modulo è realizzata con una apposita transazione (si tratta della funzione di gestione moduli di sicurezza GMOD zzPT39).

Il principio di funzionamento è simile a quello effettuato per le funzioni. Si controlla se sull'archivio della sicurezza esiste l'associazione tra l'operatore richiedente l'autorizzazione e il modulo di sicurezza che rappresenta i dati da autorizzare.

L'associazione può essere diretta (operatore-modulo) oppure indiretta (operatore-profilo).

A differenza del controllo sull'accesso alle funzioni, in questo caso è compito dello sviluppatore individuare a priori dei livelli logici di sicurezza del proprio programma e gestirli, come verrà indicato più avanti, tramite la chiamata ad un'apposita primitiva (ZSICUREZZA). Non è possibile prevedere alcun meccanismo di controllo automatico, dato che virtualmente esistono infiniti modi diversi di limitare e controllare l'accesso a dati.

Pertanto l'introduzione di nuovi moduli non ha alcun effetto se non è accompagnata dal cambiamento dei programmi interessati.



Figura 14:   Controllo di accesso a dati

Elenco programmi CITY per la gestione delle autorizzazioni

Nella tabella seguente sono riportati i programmi predisposti in ambiente CITY che sono coinvolti nella gestione delle autorizzazioni: per ciascuno di essi è indicato il nome del programma, il suo codice ed una breve descrizione.

|  |  |  |
| --- | --- | --- |
| **Prog.** | **Codice** | **FUNZIONE SVOLTA** |
| zzPT39 | GMOD | Gestione moduli di sicurezza  Permette di inserire, variare e cancellare un "modulo di sicurezza".  Il modulo verrà usato dai programmi per eseguire dei controlli sull'abilitazione degli operatori ad avere accesso a un dato o ad un gruppo di dati. |
| zzPT44 | IMOD | Inquiry moduli di sicurezza  Permette di visualizzare tutti i moduli di sicurezza con la loro descrizione, oppure solo una parte specificando un range di ricerca nei campi "da modulo" e "a modulo". |
| zzPT15 | GPRF | Gestione profili di sicurezza  Permette l'inserimento, variazione e cancellazione di un profilo indicando una descrizione breve e una descrizione estesa. |
| zzPT32 | IPRF | Inquiry profili di sicurezza  Permette la visualizzazione dei profili con la rispettiva descrizione specificando un range di ricerca |
| zzPT13 | GAUT | Gestione autorizzazioni  Consente di autorizzare un operatore, un profilo a possedere delle autorizzazioni sui dati o sulle funzioni. |
| zzPT10 | IOPR | Innquiry operatori  Visualizza tutte le informazioni riguardanti gli operatori, comprese le autorizzazioni concesse specificandone la provenienza (profilo o inserita manualmente) |
| zzRTCA | ZZRTCA | Routine di accesso alla tabella della sicurezza  Permette l'accesso vero e proprio alla tabella zzSSICR (tabella sicurezza), dove si trovano le associazioni operatore-modulo/funzione. |

# Gestione delle stampe on-line

Questo documento ha lo scopo di spiegare il meccanismo di funzionamento delle stampe On-Line secondo lo standard CITY.

Il problema della gestione delle stampe on-line è piuttosto comune e per evitare soluzioni non standard è stato predisposto un insieme di strumenti volti a facilitarne il compito.

Tali strumenti consistono in un insieme di funzioni per la preparazione ed il lancio delle stampe (da chiamare dall'interno dei programmi) e di alcune transazioni che permettono di stabilire delle associazioni fra Terminale, Programma, Modello di stampa e Stampante in modo da rendere il controllo delle stampanti e dei modelli di stampa flessibile e soprattutto indipendente (per quanto possibile) dal programma stesso.

### Modelli di stampa

Volendo stampare dei dati con un certo formato, ci si trova a dover gestire due entità distinte:

i dati da stampare

il formato con cui i dati verranno stampati (che d'ora innanzi si chiamerà *modello* di stampa)

I dati dipendono inevitabilmente dal programma che si sta realizzando e devono quindi essere gestiti all'interno del programma stesso.

Il modello invece può essere svincolato dallo specifico programma che si sta sviluppando: possono essere previsti più formati per gli stessi dati (che magari potrebbero venire stampati su moduli diversi) oppure al contrario diversi programmi potrebbero stampare sullo stesso modulo con il medesimo modello.

Perciò la determinazione dell'effettivo formato di stampa è opportuno che sia svolta da una routine esterna ed autonoma, richiamabile da qualunque programma.

## Routine di formattazione della stampa

Per ogni modello di stampa deve essere predisposta una routine con il compito di disporre i dati che riceve in input in una pagina pronta per la stampa; tale routine verrà poi chiamata dalla transazione di gestione delle stampe (non direttamente dal programma) quando sarà richiesta una stampa.

## Associazione fra Modello di stampa e Routine di formattazione

Ciascuna routine di formattazione della stampa implementa un Modello di stampa: in altre parole modello e routine individuano la stessa cosa; il modello rappresenta il formato che si vuol ottenere e la routine specifica *come* ottenerlo.

Dall'interno del programma ci si riferisce al formato indicando il nome del Modello, ma ovviamente in fase di stampa il sistema deve lanciare in esecuzione la routine corrispondente; per fare ciò è necessario indicare esplicitamente le associazioni fra i nomi dei modelli di stampa e le corrispondenti routine di formattazione.

Questa attività viene svolta tramite due programmi predisposti nella CITY che hanno il compito rispettivamente di:

inserire nuove associazioni (progr. zzPT24, GMDL - Gestione Modelli di stampa);

mostrare quelle già esistenti nel sistema (progr. zzPT25, IMDL - Inquiry Modelli di stampa).

Si osservi che il collegamento fra programma che deve stampare e routine di formattazione è dinamico e può essere modificato senza alterare e ricompilare il programma.

## Stampante utilizzata

Importante quando si prepara un programma che deve poter lanciare stampe on-line è poter determinare quale stampante verrà usata effettivamente, oppure imporre l'uso di una particolare stampante a fronte del fatto che è stato lanciato il programma xxxx sul terminale yyyy.

La scelta della stampante avviene in base a

Modello di stampa impiegato

Transazione che richiede la stampa

Terminale su cui gira la transazione

Quando si richiede la stampa, la routine di formattazione ricerca sull'archivio PRT la stampante associata a quella terna Modello-Transazione-Terminale.

Per gestire l'archivio delle associazioni suddette sono stati predisposti in ambiente CITY tre programmi che hanno il compito rispettivamente di:

inserire nuove associazioni (progr. zzPT20, APRT - Gestione Associazione Stampanti);

mostrare quelle già esistenti nel sistema (progr. zzPT27, IPRT- Inquiry Associazione Stampanti);

controllarne una in particolare (progr. zzPT21, CPRT - Controllo Associazione Stampanti).

E' anche possibile omettere uno o più degli elementi della terna <Modello-Transazione-Terminale>: ad esempio associando la stampante 'PRT1' al Modello 'MOD' e alla transazione 'TRX1', detta stampante verrà selezionata ogni volta che si cercherà di stampare con quel modello e da quella transazione, *indipendentemente* dal terminale impiegato.

Ciò offre una grandissima flessibilità ma introduce un potenziale indeterminismo nel criterio di scelta, dato che è possibile trovare in archivio più associazioni che rispettano le condizioni di ricerca.

Per risolvere questo indeterminismo è stata introdotta una priorità nel criterio di ricerca, che è la seguente:

1. Modello-Transazione-Terminale

2. Modello-Transazione

3. Modello-Terminale

4. Modello

5. Transazione

6. Terminale

**ESEMPIO**

Supponiamo che le sole associazioni presenti in archivio siano le seguenti:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| MODELLO | TRANSAZIONE | TERMINALE | STAMPANTE | STAMPANTE GRAFICA |
| mod1 | trx1 | t100 | pr01 |  |
| mod2 | trx1 | t100 | pr02 |  |
| mod1 |  | t100 | pr03 |  |
| mod1 |  | t105 | pr04 | PRTGRAF1 |
|  |  | t100 |  | PRTGRAF2 |

Se un utente lavorasse al terminale T100 e lanciasse una stampa usando il modello MOD1 e tale modello fosse definito come testo e non grafico, la stampante scelta sarebbe:

PR01 se la transazione che ha richiesto la stampa fosse TRX1.

PR03 per ogni altra transazione.

Se l'utente si spostasse sul terminale T105 sarebbe PR04, indipendentemente dalla transazione.

Se invece il modello MOD1 fosse definito come grafico, la stampante scelta sarebbe PRTGRAF2 per qualsiasi modello/terminale.

Se l'utente si spostasse sul terminale T105 sarebbe PRTGRAF1, indipendentemente dalla transazione.

Si osservi che, essendo le suddette associazioni gestite in modo il più possibile svincolato dal programma che deve effettuare la stampa, nella maggior parte dei casi è possibile controllare lo smistamento delle stampe sulle diverse stampanti *senza* modificare il programma stesso.

## Elenco delle funzioni di stampa

Le funzioni predisposte in ambiente CITY per il controllo delle stampe sono:

ZPRT-DEFAULT Legge il nome della stampante di default

ZPRT-DEFAULT-FORZA Legge il nome della stampante di default forzando la modalità di stampa

ZPRT-CNTL Verifica che il nome di stampante indicato sia valido

ZPRT-INIZIALIZZA Inizializza le variabili comuni utilizzate dalla routine di stampa

ZPRT Segnala l'inizio di una sessione di stampa con Spooler

ZPRT-NOSPOOL Segnala l'inizio di una stampa diretta (senza Spooler)

ZPRT-OK La sessione di stampa è terminata, pronta per essere effettivamente realizzata

ZPRT-CANCEL Annulla la sessione di stampa in corso



Figura 15:   Organizzazione delle chiamate per la gestione delle funzioni di stampa.

## Schemi delle fasi di stampa

Per poter effettuare delle stampe on-line bisogna:

1. Definire un modello

2. Scrivere una routine di formattazione per quel modello

3. Associare Modello e Routine (transazione GMDL)

4. Associare <Modello-Transazione-Terminale> ad una stampante (transazione APRT)

quindi le fasi per effettuare una stampa senza Spool da programma sono:

4. Inizializzazione

5. Scelta della stampante

6. Stampa

invece usando lo Spool sono:

4. Inizializzazione

5. Scelta della stampante

6. Stampa logica (Invio dei dati sullo Spool)

7. Stampa fisica

Gli schemi che seguono rappresentano le varie fasi della stampa; la determinazione della stampante da usare in funzione della terna <Modello-Transazione-Terminale> non è rappresentata perchè avviene prima dell'invio dei dati sullo Spool e non in fase di stampa vera e propria.



Figura 16:   Schema delle fasi di stampa con lo Spool.



Figura 17:   Schema delle fasi di stampa senza lo Spool.

Elenco programmi CITY per la gestione delle stampe

Nella tabella seguente sono riportati i programmi predisposti in ambiente CITY che sono coinvolti nella gestione delle stampe: per ciascuno di essi è indicato il nome del programma, il suo codice ed una breve descrizione.

|  |  |  |
| --- | --- | --- |
| **Prog.** | **Codice** | **FUNZIONE SVOLTA** |
| zzPT20 | APRT | Gestione Associazione stampanti  Permette di inserire una associazione fra una terna  <Modello-Transazione-Terminale> ed una stampante (cfr par. 3) |
| zzPT27 | IPRT | Inquiry Associazione stampanti  Permette di scandire tutte le suddette associazioni esistenti nel sistema. |
| zzPT21 | CPRT | Controlla l'associazione della stampante  Permette di verificare l'esistenza di una particolare associazione, quando non si è sicuri di quale stampante verrà effettivamente usata. |
| zzPT24 | GMDL | Gestione Moduli di stampa  Permette di inserire una associazione fra un modello di stampa ed una routine di formattazione (cfr par. 1.3), indicando alcune opzioni:  - Esecuzione (Immediata / Differita / Bloccata)  - Modulo (Continuo / Discreto)  - Densità (CPP e LPP) |
| zzPT25 | IMDL | Inquiry Moduli di stampa  Permette di scandire tutte le suddette associazioni esistenti nel sistema. |
| zzPT26 | GTRX | Gestione Transazioni  Permette di inserire una associazione fra un programma ed una transazione, indicando il codice della transazione ed alcune opzioni. |
| zzPT45 | ITRX | Inquiry Transazioni  Permette di scandire tutte le suddette associazioni esistenti nel sistema. |
| zzPT01 | GTER | Gestione Terminali |
| zzPT07 | ITER | Inquiry Terminali |
| zzPT28 | GSPO | Gestione Spool di stampa  Permette di verificare le stampe presenti sullo Spool, fornendo per ciascuna di esse:  - Modello  - Transazione  - Operatore  - Terminale  - Ora  - Data |

|  |  |  |
| --- | --- | --- |
| zzPT29 | ISTA | Inquiry visualizzazione stampe |
| zzWPRTT |  | COPY delle funzioni di stampa (parte dati) |
| zzCPRTT1 |  | COPY delle funzioni di stampa (parte codice) |
| zzRT01 |  | Routine di Gestione delle stampe |
| zzPT95 |  | Programma di gestione delle stampe vere e proprie |

# Struttura standard di un programma batch

Il presente capitolo ha lo scopo di presentare la struttura di massima che deve avere un programma BATCH implementato secondo lo standard CITY.

Le parti indicate in grassetto indicano i punti che variano in funzione del particolare programma che si sta implementando. Le altre parti sono pressoché uguali per tutti i programmi.

Le sigle in grassetto hanno il seguente significato:

**yy** sigla del programma

**xxx** sigla di un archivio usato dal programma

Il programma

\*\*\*SET\*OGGETTO=PROGRAMMA

\*\*\*SET\*MODULO=BASE

\*\*\*SET\*MONITOR=NO

\*\*\*SET\*DATABASE=SI

\*\*\*SET\*VERSIONE=SRC

\*\*\*DOC\*TITOLO

\* FRAME PROGRAMMA BATCH IN STANDARD CITY

\*\*\*DOC\*DESCRIZIONE

\* DESCRIZIONE FRAME PER PROGRAMMA BATCH

\*\*\*DOC\*AGGIORNAMENTI

010180\* NESSUNO

\*\*\*DOC\*FINE

ID DIVISION.

PROGRAM-ID. zzPB**yy**.

COPY zzWAUT.

DATE-WRITTEN. MARZO 93.

DATE-COMPILED.

COPY zzWREM.

ENVIRONMENT DIVISION.

CONFIGURATION SECTION.

COPY zzWCFG.

SPECIAL-NAMES.

C01 IS SALTO

DECIMAL-POINT IS COMMA.

INPUT-OUTPUT SECTION.

FILE-CONTROL.

\* +------------------------------------------------------------+

\* + DATA DIVISION +

\* +------------------------------------------------------------+

DATA DIVISION.

FILE SECTION.

\* +------------------------------------------------------------+

\* + W O R K I N G S T O R A G E +

\* +------------------------------------------------------------+

WORKING-STORAGE SECTION.

01 ZPGM PIC X(08) VALUE 'zzPB**yy**'.

01 ZVER PIC 9(04) VALUE 1.

\* +------------------------------------------------------------+

\* + EVENTUALI TRACCIATI DI STAMPA +

\* +------------------------------------------------------------+

\*

\* +------------------------------------------------------------+

\* + COPY PER LA GESTIONE DELLE SKEDE PARAMETRO +

\* +------------------------------------------------------------+

\* COPY ZZWSKE.

\* +------------------------------------------------------------+

\* + ZSKENOM-NN - NOME DEL PARAMETRO +

\* + ZSKELST-NN - VALORE PER DEFAULT E/O PER CONTROLLO +

\* + SI POSSONO INSERIRE UNA SERIE DI VALORI +

\* + DELIMITATI DA " PER INCLUDERE SPAZI E +

\* + SEPARATI DA ; +

\* + IL PRIMO DELLA SERIE VIENE USATO PER IL +

\* + DEFAULT SE RICHIESTO +

\* + ZSKEDIG-NN - VALORE DIGITATO PER IL PARAMETRO +

\* + NEL CASO IL PARAMETRO SIA UNA DATA ALLORA +

\* + VERRA' RITORNATO ZSKEDIG-NN COSI' +

\* + SOTTO-DEFINITO: +

\* + 09 ZSKEDIG-XX-SAMG PIC S9(08). +

\* + DATA IN INPUT NEL FORMATO SSAAMMGG +

\* + +

\* + 09 ZSKEDIG-XX-AMG PIC S9(06). +

\* + DATA IN INPUT NEL FORMATO AAMMGG +

\* + +

\* + 09 ZSKEDIG-XX-GMSA PIC S9(08). +

\* + DATA IN INPUT NEL FORMATO GGMMSSAA +

\* + +

\* + 09 ZSKEDIG-XX-GMA PIC S9(06). +

\* + DATA IN INPUT NEL FORMATO GGMMAA +

\* + +

\* + 09 ZSKEDIG-XX-GMSA-E PIC X(10). +

\* + DATA IN INPUT NEL FORMATO GG/MM/SSAA +

\* + +

\* + 09 ZSKEDIG-XX-GMA-E PIC X(08). +

\* + DATA IN INPUT NEL FORMATO GG/MM/AA +

\* + +

\* + 09 ZSKEDIG-XX-GMMMSSAA-E PIC X(17). +

\* + DATA IN INPUT NEL FORMATO GG/MMM/SSAA +

\* + +

\* + 09 ZSKEDIG-XX-GGG-E PIC X(09). +

\* + NOME DEL GIORNO +

\* + +

\* + ZSKEDEF-NN - SWITCH DI DEFAULT +

\* + S - SI DEFAULT +

\* + N - NO DEFAULT +

\* + N.B. SE NON E' PREVISTO IL DEFAULT +

\* + IL PARAMETRO E' OBBLIGATORIO +

\* + ZSKETIP-NN - TIPO PARAMETRO +

\* + A - ALFANUMERICO +

\* + N - NUMERICO +

\* + D - DATA +

\* + L - DATA LAVORATIVA +

\* + ZSKEINT-NN - NUMERO INTERI (SOLO ZSKETIP-NN = 'N') +

\* + ZSKEDEC-NN - NUMERO DECIMALI (SOLO ZSKETIP-NN = 'N') +

\* + ZSKEMIN-NN - VALORE MINIMO (SOLO ZSKETIP-NN = 'N') +

\* + ZSKEMAS-NN - VALORE MASSIMO (SOLO ZSKETIP-NN = 'N') +

\* + ZSKEHLP-NN - CODICE PER HELP +

\* + ZSKEERR-NN - FLAG DI ERRORE INDICANTE : +

\* + 0 - NESSUN ERRORE +

\* + 1 - PARAMETRO NON DIGITATO +

\* + 2 - DATO DIGITATO OVERFLOW INTERI +

\* + 3 - DATO DIGITATO OVERFLOW DECIMALI +

\* + 4 - SCELTA NON PRESENTE IN ZSKELST +

\* + 5 - DATA ERRATA +

\* + 6 - DATA FESTIVA (LA SI VOLEVA LAVORATIVA) +

\* + 7 - SOTTO IL MINIMO +

\* + 8 - SOPRA IL MASSIMO +

\* + 9 - DATO DIGITATO NON NUMERICO +

\* + 10 - IN ZSKELST VI SONO PIU' DI 40 SCELTE +

\* + 11 - DATO DIGITATO CON PIU' DI 18 CIFRE +

\* + 12 - MINIMO NON NUMERICO +

\* + 13 - MINIMO OVERFLOW INTERI +

\* + 14 - MINIMO OVERFLOW DECIMALI +

\* + 15 - MINIMO CON PIU' DI 18 CIFRE +

\* + 16 - MASSIMO NON NUMERICO +

\* + 17 - MASSIMO OVERFLOW INTERI +

\* + 18 - MASSIMO OVERFLOW DECIMALI +

\* + 19 - MASSIMO CON PIU' DI 18 CIFRE +

\* + 20 - ZSKEDEF-NN ERRATO +

\* + 21 - ZSKETIP-NN ERRATO +

\* + 22 - ZSKEINT-NN NON NUMERICO +

\* + 23 - ZSKEDEC-NN NON NUMERICO +

\* + 24 - ZSKEMSG-NN NON NUMERICO +

\* + 25 - DEFAULT ACCETTATO +

\* + 11XX - XX ELEMENTO LISTA NON NUMERICO +

\* + 12XX - XX ELEMENTO LISTA OVERFLOW INTERI +

\* + 13XX - XX ELEMENTO LISTA OVERFLOW DECIMALI +

\* + 14XX - XX ELEMENTO LISTA CON PIU' DI 18 CIFRE +

\* + N.B. XX = NUMERO PROGRESSIVO LISTA +

\* +------------------------------------------------------------+

\* +------------------------------------------------------------+

\* + COPY DI WORKING UTENTE +

\* +------------------------------------------------------------+

\* STRUTTURA TRACCIATO DA RIPETERE PER TUTTI I TRACCIATI

COPY zzS**xxx**R.

\* AREA PER GESTIONE STAMPANTI

COPY zzWPRT.

\* AREA PER GESTIONE SEGNALI

COPY zzWSIG.

\* AREA PER COSTANTI SISTEMA

COPY zzWCOST.

\* AREE PER GESTIONE DATA

COPY zzWDAT1.

COPY zzWDAT2.

\* AREA PER COMUNICAZIONE TRA ROUTINE

COPY zzWLNK.

\* AREA PER GESTIONE INTERFACCIA AI DATI

COPY zzWIOI.

\* +------------------------------------------------------------+

\* + PROCEDURE DIVISION +

\* +------------------------------------------------------------+

PROCEDURE DIVISION.

\* +------------------------------------------------------------+

\* + FRAME STANDARD PER PROGRAMMI BATCH +

\* +------------------------------------------------------------+

COPY zzCPB.

\* +------------------------------------------------------------+

\* + FORMATTAZIONE/INIZIALIZZAZIONE CAMPI SISTEMA +

\* +------------------------------------------------------------+

ZFORMAT-USR SECTION.

ZFORMAT-USR-INIT.

ZFORMAT-USR-EXIT.

EXIT.

\* +------------------------------------------------------------+

\* + APERTURA FILES PROGRAMMA +

\* +------------------------------------------------------------+

ZAPRI-FILE SECTION.

ZAPRI-FILE-INIT.

PERFORM ZOPEN-INPUT-zzSTAB1.

ZAPRI-FILE-EXIT.

EXIT.

\* +------------------------------------------------------------+

\* + CHIUSURA DEI FILES DI PROGRAMMA +

\* +------------------------------------------------------------+

ZCHIUDI-FILE SECTION.

ZCHIUDI-FILE-INIT.

PERFORM ZCLOSE-zzSTAB1.

ZCHIUDI-FILE-EXIT.

EXIT.

\* +------------------------------------------------------------+

\* + FORMATTAZIONE/INIZIALIZZAZIONE CAMPI DI UTENTE +

\* +------------------------------------------------------------+

ZFORMAT-WORKING SECTION.

ZFORMAT-WORKING-INIT.

ZFORMAT-WORKING-EXIT.

EXIT.

\* +------------------------------------------------------------+

\* + CONTROLLO/IMPOSTAZIONE DEI CAMPI DIPENDENTI DALLE SKEDE +

\* +------------------------------------------------------------+

ZCNTL-SKEDE SECTION.

ZCNTL-SKEDE-INIT.

ZCNTL-SKEDE-EXIT.

EXIT.

\* +------------------------------------------------------------+

\* + CONTROLLO/IMPOSTAZIONE DEI CAMPI DIPENDENTI DALLE SKEDE +

\* +------------------------------------------------------------+

ZFORMAT-IST-USR SECTION.

ZFORMAT-IST-USR-INIT.

ZFORMAT-IST-USR-EXIT.

EXIT.

\* +------------------------------------------------------------+

\* + ELABORA I DATI DEL TABULATO TRANSAZIONI +

\* +------------------------------------------------------------+

ZELABORA SECTION.

ZELABORA-INIT.

ZELABORA-EXIT.

EXIT.

\* +------------------------------------------------------------+

\* + INIZIO COPY DI PROCEDURE DIVISION +

\* +------------------------------------------------------------+

\* GESTIONE ARCHIVIO TABELLE

COPY zzLTAB1D.

\* GESTIONE ARCHIVI DA RIPETERE PER TUTTI GLI ARCHIVI

COPY zzL**xxx**1T.

\* AZZERAMENTI TRACCIATO XXX - DA RIPETERE PER OGNI TRACCIATO

COPY zzZ**xxx**1T.

\* SECTION PER GESTIONE STAMPANTI

COPY zzCPRT.

\* SECTION PER GESTIONE SEGNALI

COPY zzCSIG.

\* SECTION PER GESTIONE DATA

COPY zzZDAT.

COPY zzCDAT.

COPY zzCDATB.

\* SECTION PER ESECUZIONE ROUTINE

COPY zzCLNKB.

\* SECTION PER ROUTINE DI INTERFACCIA AI DATI

COPY zzCIOID.

\* SECTION PER GESTIONE ABEND

COPY zzCABEB.

\* SECTION PER GESTIONE SKEDE PARAMETRO

COPY zzCSKE.

\* SECTION PER STOP RUN

### Copy zzCPB

La copy zzCPB contiene il vero e proprio programma principale di una qualsiasi applicazione batch. Da questa copy vengono eseguite, tra le altre, tutte le section del programma utente.

Nella figura, viene indicato il flusso di esecuzione delle section contenute nella copy zzCPB:



Figura 1:   flusso esecuzione zzCPB

## Flusso di esecuzione di un programma batch

La figura che segue indica il flusso di esecuzione di un qualsiasi programma batch, ponendo attenzione alla lettura delle skede parametro.



Figura 2:   lettura skede parametro

# Schede Parametro per programmi batch

L'input dei dati ai programmi batch è effettuato mediante un file denominato '*schede parametro'.* Esso contiene oltre ai parametri, alcune variabili per identificare gli stessi parametri e associarli alle opportune elaborazioni.

NOTA: il prefisso "zz" identifica il nome della procedura a cui il programma appartiene. Nel documento viene usato il prefisso "zz", per identificare lo standard CITY, ma dovrà in fase di programmazione essere sostituito con il prefisso specifico della procedura.

## Definizione delle schede parametri

La gestione dei parametri standard CITY è realizzata con caratteristiche particolari non devono essere digitati sequenzialmente nel modo tradizionale, ma ognuno di essi è caratterizzato da una parola chiave che ne permette l' identificazione da parte della routine di gestione (zzRDSK), fungendo da chiave di ricerca.

I due file utilizzati per la gestione dei parametri sono:

|  |  |
| --- | --- |
| CADPAR | contiene tutti i parametri utilizzati da tutte le procedure o da una singola procedura (è possibile creare più file CADPAR) |
| CADSET | contiene un'unica serie di parametri che permette di individuare in CADPAR una serie di parametri in modo univoco. |

La creazione dei file di cui sopra avviene utilizzando un metalinguaggio che permette di fornire ai programmi batch tutte le informazioni di cui necessitano.

In particolare dovremo utilizzare le seguenti parole/caratteri chiave:

**\*** (asterisco) posizionato a colonna 1, permette di inserire qualsiasi tipo di commento.

**$PAR** divide i gruppi logici di parametri all'interno di CADPAR (sono possibili più $PAR in CADPAR)

**$SET** identifica una serie univoca di parametri all'interno di CADSET (è possibile un solo SET in CADSET)

**=** (uguale) è elemento di assegnazione: alla variabile viene assegnato un valore

**;** (punto e virgola) è elemento di separazione tra un parametro e quello successivo (indica fine parametro)

**" "** (doppi apici) sono utilizzati in tutti i casi in cui a far parte della valorizzazione del parametro intervengano dei blank (il programma non tiene conto di tutti i blank non compresi tra doppi apici)

**. : / -** (punto, doppio punto, slash, meno) nei campi numerici vengono indifferentemente utilizzati come elementi di separazione tra gruppi di cifre

## File cadpar

In particolare nel file *cadpar* sono definiti tutti i parametri che sono utilizzati dal programma. In pratica il file *cadpar* deve essere di volta in volta implementato con i valori che si vogliono far elaborare dal programma.

All'inizio di ogni serie di parametri, come identificativo viene usata la parola chiave *$PAR*.

Esistono tre diverse tipologie di parametri da utilizzare in ogni serie (identificata da *$PAR*) e sono:

parametri obbligatori di sistema

parametri facoltativi

parametri utente per il programma

### Parametri obbligatori

|  |  |  |
| --- | --- | --- |
| PROCEDURA | = | nome della procedura |
| JOB | = | nome del job |
| STEP | = | identificativo del passo eseguito |
| PROGRAMMA | = | nome del programma |
| ISTITUTO | = | identificativo dell'istituto |
| ELABORAZIONE | = | numero progressivo dell'elaborazione |

dove

|  |  |
| --- | --- |
| PROCEDURA | è un nome che identifica, all'interno dell'ambiente aziendale, la procedura |
| JOB | è il nome del job che userà questa scheda parametro |
| STEP | permette di associare la scheda parametro allo step del programma. Se non viene specificato lo step il suo valore sarà NOSTEP. |
| PROGRAMMA | è il nome del programma che utilizzerà questa scheda parametro |
| ISTITUTO | identifica l' istituto che utilizzerà questa scheda parametro. Se viene specificato un asterisco \* verranno elaborati tutti gli istituti censiti. |
| ELABORAZIONE | è un numero progressivo che permette di eseguire più volte lo stesso programma |

.

### Parametri facoltativi di sistema

|  |  |  |
| --- | --- | --- |
| TRACE | = | attivazione (non attivazione) di una serie di opzioni di debug |
| SPOOL | = | chiave utente da inserire in coda alla chiave di sistema per la gestione di stampe in SPOOL |
| SEGNALE | = | valore numerico dell'intervallo al raggiungimento del quale viene fornita una segnalazione (il valore di default e' 1000; il parametro e' passato alla routine di segnalazione) |

### Parametri utente per il programma

|  |  |  |
| --- | --- | --- |
| nome parametro 1 | = | valore del parametro 1 |
| nome parametro 2 | = | valore del parametro 2 |
| nome parametro .. | = | valore del parametro ..... |
| nome parametro 50 | = | valore del parametro 50 (numero massimo di parametri) |

## File cadset

Nel file *cadset* deve essere definita una serie di informazioni contraddistinta da *$SET*. Non devono esistere più file *cadset* con gli stessi parametri JOB e STEP e non devono essere usati più *$SET* nello stesso file di *cadset.*

In pratica il file *CADSET* viene riportato in tutti i nuovi programmi e al suo interno i campi relativi a JOB e STEP devono essere modificati in relazione alle esigenze specifiche.

All'inizio di ogni serie di parametri, come identificativo, viene usata la parola chiave *$SET*.

I parametri obbligatori da utilizzare in *$SET* sono:

|  |  |  |
| --- | --- | --- |
| JOB | = | nome del job |
| STEP | = | identificativo del passo eseguito |

quelli facoltativi:

|  |  |  |
| --- | --- | --- |
| INPUT | = | può assumere i valori: |
|  |  | JCL se i parametri devono essere cercati in un file (valore di default) |
|  |  | CONSOLE se i parametri devono essere digitati a video/console |
| STAMPANTE | = | può avere i seguenti valori: |
|  |  | IMPATTO (valore di default) |
|  |  | LASER |
|  |  | SPOOL (coda di stampa su file) |
| RIGHE | = | numero di righe per pagina (default=60) |
| ISTITUTO | = | può assumere i valori: |
|  |  | TABELLA se la routine *zz*RDSK accede agli archivi tp per la lettura dei dati di sistema necessari. E' il valore di default. |
|  |  | SEQUENZIALE se la routine accede agli archivi sequenziali per la lettura dei dati di sistema. In questo caso prima di eseguire il batch è necessario scaricare da archivi tp su sequenziali alcune tabelle City richiamando il batch di scarico *zz*PB00. |

Deve esistere un file *cadset* per ogni esecuzione (step) dell'elaborazione (cioè deve esserci un file *cadset* per ogni accoppiata JOB, STEP). Il valore del parametro STEP del file *cadset* deve avere un corrispondere valore nel parametro STEP di una scheda del file *cadpar*.

# Segnalazioni

Le segnalazioni permettono di avere in report, dei conteggi di eventi a cura dello sviluppatore. Questo significa che lo sviluppatore dispone di contatori (fino a 50) e delle operazioni su di essi che gli permettono di contare e gestire degli eventi. Sui contatori è possibile fare incrementi e decrementi, salvarne lo stato e visualizzare un messaggio di descrizione sia quando il numero di conteggi raggiunge un valore predefinito, sia in modo forzato, mediante una specifica perform.

Tutto questo è possibile tramite routine o sottoprogrammi definiti dallo standard CITY.

## La routine di segnalazione zzRBSIG

Questa routine ha le seguenti funzioni:

aprire e chiudere il file di segnalazione

inserire una testata standard con i parametri identificativi del programma

ripetere la testata standard con i parametri identificativi del programma

ripetere la testata standard ad ogni cambio di elaborazione

permettere di forzare la segnalazione

gestire una riga testata-descrizione a ogni segnalazione forzata

# Errori

Le gestione degli errori permette di avere in report, dei tabulati che riportano gli errori che si desidera vengano segnalati.

Tale gestione effettuata da una apposita routine chiamata da una perform eseguita nel programma utente permette di stampare un tabulato composto al massimo da 50 errori.

## La routine di gestione degli errori zzRBERR

Questa routine si occupa delle seguenti funzioni:

aprire e chiudere il file degli errori

inserire una testata standard con i parametri identificativi del programma

stampare una segnalazione di errore

gestire una riga di descrizione a ogni report di errore

# Gestione delle stampe batch

Scopo di questo capitolo è descrivere le modalità offerte dallo standard CITY per la realizzazione di stampe da programmi batch.

## La routine di stampa zzRBPRT

Un programma batch scritto nello standard CITY può gestire fino a nove tabulati diversi, le cui strutture dati (da includere nella working storage section) sono generate in modo automatico compilando un disegno che descrive come i dati dovranno essere stampati.

La stampa vera e propria verrà eseguita dalla routine specifica che è: zzRBPRT.

La routine di stampa ha le seguenti funzioni:

aprire e chiudere i file di stampa (fino a 9)

gestire una testata standard con i parametri identificativi del tabulato

gestire una riga di fine tabulato con alcuni parametri identificativi

gestire dieci righe di testata opzionali a disposizione

gestire il numero di righe per pagina di ciascun tabulato

gestire il numero di righe da saltare (default=1)

gestire il numero di colonne per tabulato

gestire il salto pagina

## Composizione della stampa

La stampa definitiva è composta dalle seguenti parti:

testata standard

eventuale seconda testata standard

tabulato

riga standard di fine tabulato

### La testata standard

Essa, a meno che specifiche esigenze non ne escludano l'utilizzo, è sempre presente nel tabulato e ne contiene tutti gli identificativi:

nome programma

numero progressivo tabulato

nome job

nome step

descrizione istituto

codice istituto

codice dipendenza

ora

data

codice identificativo sistema

versione programma

descrizione tabulato

numero elaborazione

numero pagina

### La riga standard di fine tabulato

La riga standard di fine tabulato riporta alcune delle indicazioni già presenti nella testata standard e ha funzione di chiusura del tabulato o di separazione tra i diversi cambi di elaborazione all'interno di uno stesso tabulato. Sono contenuti nella riga standard di fine tabulato:

codice istituto

codice dipendenza

nome job

nome step

nome programma

numero progressivo tabulato

versione programma (release)

Si può evitare, se esigenze particolari lo richiedono, che compaiano la testata standard e la riga di fine tabulato, ma se ne raccomanda l'uso al fine di mantenere un certo tipo di uniformità nelle stampe.

# Raggruppamento in moduli e interazione tra i componenti

Precisiamo che, il raggruppamento in moduli e l’interazione tra i componenti è descritto nei Manuali Utente pubblicati da CAD IT SpA elencati analiticamente nel capitolo in questione sia per Ruoli ante riforma che per post riforma, in una specifica tabella che contiene le informazioni:

* Area procedurale di riferimento (ordinamento alfabetico decrescente)
* Descrizione sintetica Procedura
* Descrizione delle funzionalità generali della procedura contenute nel manuale
* Nome del file relativo al manuale utente

| ***Area*** | ***Procedura*** | ***Descrizione*** | ***Id. Manuale*** |
| --- | --- | --- | --- |
| Anagrafica | Anagrafica Contribuenti | Anagrafe generale - Funzionalità della procedura di gestione anagrafica contribuenti | Anagrafica.doc |
| Anagrafica | Certificazione CCIAA | Certificazione dati anagrafici Persone Giuridiche da CCIAA | Certificazione Anagrafica da CCIAA.doc |
| Anagrafica | Scambio Informazioni Anagrafiche | Procedura di gestione dello scambio Dati Anagrafici standard fra AdR ed Equitalia (ex-EQS) | Scambio\_informazioni\_anagrafiche.doc |
| Anagrafica | Stradario | Normalizzazione Archivio Anagrafico con definizione zone territoriali e strade comunali | Stradario.doc |
| Ancillari | Processi schedulati - Multischeduling | Gestione delle procedure batch di acquisizione Fascicoli Area Terzi e Immobiliare - Parametri di configurazione | CO@TTIVO - Processi Schedulati-Multitasking.doc |
| Ancillari | Immobiliare - Ipoteca Gestore | Modulo di interfaccia con la Procedura GESTORE di Equitalia per le Ipoteche Immobiliari | CO@TTIVO - Area Immobiliare - Ipoteca Gestore.doc |
| Ancillari | Mappa Flussi interscambio Gestore | Documento riepilogativo con la mappa dei flussi di interscambio con la Procedura GESTORE di Equitalia | CO@TTIVO - Area Immobiliare - Ipoteca Gestore Flussi Interscambio.doc |
| Ancillari | Elementi generali Co@ttivo | Impostazioni generali e gestione Tabelle di base per la configurazione del sistema Co@ttivo (Ancillari) | CO@TTIVO - Elementi Generali.doc |
| Ancillari | Processi schedulati - Area presso Terzi | Gestione delle schedulazioni per elaborazioni massive Procedura presso Terzi | CO@TTIVO - Area Presso Terzi - Processi Schedulati.doc |
| Ancillari | Pignoramenti 28-Ter - Area presso Terzi | Gestione della procedura di pignoramento presso Terzi ai sensi del 28-Ter | CO@TTIVO - Area Presso Terzi - Pignoramenti 28Ter.doc |
| Ancillari | Immobiliare - Recupero CPI off-line | Procedura per il recupero delle CPI emesse off-line (procedura transitoria) | CO@TTIVO - Area immobiliare - Recupero attività transitoria CPI off-line.doc |
| Ancillari | Processi schedulati - Area Immobiliare | Gestione delle schedulazioni per elaborazioni massive Procedura Immobiliare | CO@TTIVO - Area Immobiliare - Processi schedulati.doc |
| Ancillari | Processi schedulati - Co@ttivo | Gestione generale schedulazione processi ancillari - Emissione massiva documenti | CO@TTIVO - Processi Schedulati.doc |
| Ancillari | Interscambio Flussi SET - Area Terzi | Gestione flussi di interscambio con Sistema SET per Piognoramento Terzi e Pignoramento P.A. (48-Bis) | CO@TTIVO - Interscambio Flussi SET.doc |
| Ancillari | Gestione Ammortamento - Area Terzi | Funzionalità di controllo dei Pagamenti da parte dei Terzi pignorati per l'ammortamento del debito iniziale | CO@TTIVO - Area Presso Terzi - Gestione Ammortamento.doc |
| Ancillari | Immobiliare - Elementi Tipici | Tabelle generali e gestione Schede archivi principali (Immobili,etc..) - Concetti di base procedura Immobiliare | CO@TTIVO - Area Immobiliare - Elementi Tipici.doc |
| Ancillari | Pignoramento PA 48-bis Area Terzi | Configurazione Iter operativi per la gestione della Procedura di Pignoramento presso Terzi della P.A. (48-Bis) | CO@TTIVO - Area Presso Terzi - Pignoramenti PA.doc |
| Ancillari | Area presso Terzi - Elementi Tipici | Funzionalità generali e dettaglio gestione Archivi principali della Procedura di Pignoramento presso Terzi | CO@TTIVO - Area Presso Terzi - Elementi Tipici.doc |
| Ancillari | Area presso Terzi - Operatività Massiva | Gestione Flussi per acquisizione batch dati Procedura presso terzi ed emissione massiva documenti e verbali | CO@TTIVO - Area Presso Terzi - Operatività Massiva.doc |
| Ancillari | Fermo Amministrativo - Tracciati Record | Specifiche dei tracciati record di interfaccia per la gestione delle Procedure di fermo Amministrativo | CO@TTIVO - Area Fermo Amministrativo - Tracciati.doc |
| Ancillari | Area Concorsuale - Elementi Tipici | Tabelle generali e gestione Ripartizione crediti (privilegio - chirografo) per le Procedure Concorsuali | CO@TTIVO - Area Concorsuale - Elementi Tipici.doc |
| Ancillari | Immobiliare - Intervento e Surroga | Gestione Procedure di Intervento e di Surroga nell'ambito dell'area procedurale Immobiliare | CO@TTIVO - Area Immobiliare - Intervento\_Surroga.doc |
| Ancillari | Immobiliare - Vendita | Gestione Procedura esecutiva di Vendita nell'ambito dell'area procedurale Immobiliare | CO@TTIVO - Area Immobiliare - Vendita.doc |
| Ancillari | Immobiliare - Operatività Massiva | Gestione Flussi interscambio per alimentazione automatica Fascicoli Immobiliare - Stampa massiva documenti | CO@TTIVO - Area Immobiliare - Operatività Massiva.doc |
| Ancillari | Fermo Amministrativo - Processi schedulati | Gestione delle schedulazioni per elaborazioni massive richiesta di Iscrizione Fermi verso ACI-PRA | CO@TTIVO - Area Fermo Amministrativo - Processi Schedulati.doc |
| Ancillari | Fermo Amministrativo - elementi tipici | Elementi generali dei processi di gestione dei fermi Amministrativi e delle pratiche presentate ad ACI-PRA | CO@TTIVO - Area Fermo Amministrativo - Elementi Tipici.doc |
| Ancillari | Postalizzazione Documenti | Schemi della codifica dei Documenti procedura Co@ttivo per la gestione della Postalizzazione | CO@TTIVO - Postalizzazione.doc |
| Ancillari | Iter Procedura e Iter Operativo | Documenmtazione delle Tabelle per la creazione degli Iter Procedurali e degli Iter Operativi | CO@TTIVO - Atti - Stampe - Iter.doc |
| Ancillari | Files di configurazione procedura Co@ttivo | Composizione dei files di configurazione della Procedura Co@ttivo (formato XML) | CO@TTIVO - file di configurazione Istituto.xml.doc |
| Ancillari | Censimento Tabelle in attivazione | Documentazione Tabelle generali della Procedura Co@ttivo per la sua attivazione | CO@TTIVO - Censimento Tabelle in attivazione.doc |
| Ancillari | Attivazione nuovo Ambito | Documentazione per l'attivazione di un nuovo Ambito per la gestione della Procedura Co@ttivo | CO@TTIVO - Attivazione Nuovo Ambito.doc |
| Ancillari | Immobiliare - Migrazione Fascicoli | Procedura transitoria di gestione della migrazione dei Fascicoli dalla procedura Immo@web a Co@ttivo | CO@TTIVO - Area Immobiliare - Migrazione Fascicoli Immoweb Co@ttivo.doc |
| Antiriciclaggio | Interfaccia Procedura RM | Moduli di interfaccia movimenti giornalieri con la Procedura RM di gestione Antiriciclaggio | InterfacciaRM.doc |
| Contabilità | Contabilità Generale | Documentazione con le funzionalità generali della procedura di Contabilità Generale | Contabilità Generale.doc |
| Contabilità | Tabella Codici Importo | Tabella guida dei Codici Importo contabile per alimentazione interfaccia contabile | Contabilità\_CodiciImporto.doc |
| Contabilità | Partite Contabili | Censimento e gestione (scarico / chiusura) delle Partite Contabili | Partite\_contabili.doc |
| Contabilità | Partite Viaggianti | Censimento delle transazioni - Apertura, interrogazione e presa in carico Partite Viaggianti | Partite\_viaggianti.doc |
| Contabilità | Quadro di Cassa | Gestione della circolarità, Gestione casse, movimenti e chiusura quadro di cassa per operatore / sportello | QuadroCassa.doc |
| ICI | Incassi diretti ICI - IMU | Procedura di gestione delle Riscossioni e dei riversamenti ai Comuni dell'ICI e dell'IMU | ici.doc |
| ICI | Specifiche per Data Entry ICI - IMU | Specifiche Tecniche - Tracciato Record procedura Data-Entry per registrazione versamenti ICI - IMU | TRK data entry ICI ISCOP.doc |
| Ruoli | Procedure Rilevanti | Procedura di alimentazione dei dati contribuenti morosi con il package Equick di Equitalia | Equick.doc |
| Ruoli | Verifiche AVCP | Gestione Tabelle propedeutiche alla verifica del debito per le segnalazioni e verifiche di AVCP | Verifiche AVCP.doc |
| Ruoli | Rendicontazione Enti Ruoli | Transazioni di gestione Tabelle decodifica e flussi - Elaborazioni batch per creazione Flussi da trasmettere a Rendiweb | RENDIWEB\_RUOLI.doc |
| Ruoli | Rendicontazione Enti ICI-IMU | Elaborazioni batch per la predisposizione flussi da trasmettere a Rendiweb | RENDIWEB - ICI.doc |
| Ruoli | Adempimenti Garante della Privacy | Procedura per l'elaborazione e cancellazione Contribuenti in riferimento agli adempimenti richiesti dal Garante della Privacy | Adempimenti Garante della Privacy.doc |
| Ruoli | Anticipazioni - Gestione | Procedura per la gestione completa delle Anticipazioni - Funzionalità TP-On Line e Funzionalità Batch | Anticipazioni.doc |
| Ruoli | Cartellazione - Gestione | Procedura di gestione delle fasi di Cartellazione con le transazioni di interrogazione dati ed elaborazioni batch per acquisizione e monitoraggio flussi Cartelle | Cartellazione.doc |
| Ruoli | Compensazioni 28-Ter | Procedura per la compensazione volontaria Ruoli con crediti di imposta (Art. 28-ter D.P.R. 602/1973) | Compensazione 28-TER.doc |
| Ruoli | Compensazioni Art. 31 | Procedura per la gestione dell'autocompensazione in presenza di debiti su Ruoli definitivi ai sensi dell'Art. 31 del D.L. 78/2010 | Compensazione art 31 DL 78-2010.doc |
| Ruoli | Condono ex-Art. 12 Legge 289/2002 | Procedura per la gestione completa di tutte le fasi di definizione e monitoraggio del Condono emesso ai sensi dell'Art. 12 della Legge 289/2002 | Condono\_art12.doc |
| Ruoli | Condono Sanzioni Amministrative CdS | Procedura per la gestione della Definizione agevolata di Sanzioni Amministrative Codice della Strada (ex-Art. 15, commi 8-quinquiedecies e ss., D.L. 78/2009) | Condono\_CdS\_2009.doc |
| Ruoli | Condono Enti Territoriali | Procedura per la gestione del Pagamento agevolato dei carichi iscritti a ruolo da Enti territoriali | Condono\_EntiTerritoriali.doc |
| Ruoli | Contenzioso - Accesso ad A.T. - ARCO | Contenzioso - Accesso alle informazioni disponibili presso il sistema informativo (Anagrafe Tributaria) del Ministero delle Finanze (Sistema ARCO) | Contenzioso\_AccessoAT.doc |
| Ruoli | Contenzioso - Comunicazioni di Inesigibilità | Procedura di emissione e gestione delle Comunicazioni di Inesigibilità da eventi (puntuali) o da elaborazioni massive (batch) | Contenzioso\_ComunicazioniInesigibilita.doc |
| Ruoli | Contenzioso - Coobbligati in via esecutiva | Procedura di gestione dei contribuenti Coobbligati in via esecutiva | Contenzioso\_Coobbligati.doc |
| Ruoli | Contenzioso - Fermo Telematico | Procedura per la gestione dell'accesso telematico al Pubblico Registro Automobilistico e per l'iscrizione telematica del Fermo Amministrativo | Contenzioso\_FermoTelematico.doc |
| Ruoli | Contenzioso - Fogli di Lavoro | Procedura per la gestione del conferimento ad uffici / operatori di attività delle procedure cautelari e/o esecutive tramite i Fogli di Lavoro | Contenzioso\_FogliDiLavoro.doc |
| Ruoli | Contenzioso - Ipoteche tramite Gestore | Procedura di interfaccia al sistema GESTORE per la emissione e gestione delle Ipoteche Immobiliari con l'internalizzazione del servizio | Contenzioso\_Ipoteche\_Gestore.doc |
| Ruoli | Contenzioso - Procedure esecutive | Procedura di gestione delle attività cautelari ed esecutive - descrizione delle funzionalità di base e dell'impianto tabellare per la parametrizzazione e la definizione dell'Iter Procedurale | Contenzioso\_ProcedureEsecutive.doc |
| Ruoli | Contenzioso - Profilo procedurale | Gestione del Profilo Procedurale dei Contribuenti attivato ed aggiornato in fase di accesso ad A.T. ed in occasione delle attività esecutive svolte | Contenzioso\_ProfiloProcedurale.doc |
| Ruoli | Contenzioso - Rilevazione Morosità | Procedura per l'aggiornamento e la gestione della morosità Contribuenti con la valorizzazione dell'archivio "Quadro Morosi" | Contenzioso\_RilevazioneMorosità.doc |
| Ruoli | Contenzioso - Solleciti di Pagamento - Diffide | Procedura per l'emissione e la gestione dei Solleciti di Pagamento, delle Diffide, degli Avvisi di Intimazione e degli Avvisi di Mora per cartelle Ante Riforma | Contenzioso\_SollecitiAvvisi.doc |
| Ruoli | Contenzioso - Terzo@web | Procedura di interfaccia al package ancillare Terzo@web per la gestione delle attività esecutive presso Terzi | Contenzioso\_Terzo@Web.doc |
| Ruoli | Contenzioso - Visure Immobiliari Kastoffice | Procedura di gestione delle Visure Immobiliari tramite il Provider KASTOFFICE | Contenzioso\_VisuraImmobiliare\_KastOffice.doc |
| Ruoli | Contenzioso - Visure Immobiliari Ribes | Procedura di gestione delle Visure Immobiliari tramite il Provider RIBES | Contenzioso\_VisuraImmobiliare\_Ribes.doc |
| Ruoli | Contenzioso - Visure Immobiliari Sistemia | Procedura di gestione delle Visure Immobiliari tramite il Provider SISTEMIA | Contenzioso\_VisuraImmobiliare\_Sistemia.doc |
| Ruoli | Conto di Gestione | Procedura per l'emissione del Conto di Gestione degli Agente Contabili – Modello 21 per Province, Comuni, Comunità Montane, Unioni di Comuni e Città Metropolitane | Conto di gestione.doc |
| Ruoli | Conto Giudiziale Ruoli | Procedura per l'emissione e la gestione del Conto Giudiziale Ruoli | Conto\_giudiziale\_ruoli.doc |
| Ruoli | Controllo Decadenza ex-Art. 25 DPR 602 | Procedura per il controllo dei Termini di Decadenza della riscossione a mezzo ruolo ai sensi dell'Art. 25 del DPR 602 | Controllo Decadenza art 25 DPR 602.doc |
| Ruoli | Controllo di Processo acquisizione TP Provvedimenti | Procedura per la gestione del Data Entry Provvedimenti e la relativa acquisizione sugli archivi standard del Sistema SET | Controllo di Processo.doc |
| Ruoli | Definizione Agevolata pagamento carichi | Procedura per la gestione delle Riscossioni tramite Definizione Agevolata ai sensi della Legge 147/2013 commi 818-624 | Definizione agevolata carichi L. 147-2013.doc |
| Ruoli | Deleghe di Notifica | Procedura per la gestione del conferimento ed espletamento delle Deleghe di Notifica | DelegheNotifica.doc |
| Ruoli | Esiti e Stato della Riscossione | Procedura per la predisposizione dei flussi di Esito su Avvisi Bonari (GIA) e dei flussi dello Stato della Riscossione per la rendicontazione agli Enti | Esiti.doc |
| Ruoli | Estratto di Ruolo | Procedura per l'emissione parametrica degli Estratti di Ruolo con la situazione debitoria dei Contribuenti | Estratto del Debito.doc |
| Ruoli | Flussi RGS | Procedura per la predisposizione dei Flussi da trasmettere alla RGS per la rendicontazione delle entrate dello Stato | Flussi RGS.doc |
| Ruoli | Gestione Spese da richiedere solo agli Enti | Procedura per la richiesta e la gestione delle Spese di procedure Esecutive da richiedere soltanto agli Enti Impositori | GestionedelleSpesedaRichiedereSoloallEnteImpositore.doc |
| Ruoli | Notifica | Procedura per la gestione della Notifica delle Cartelle ed altre ttipologie di documento (transazioni TP e modalità batch di notifica R\_AR) | Notifica.doc |
| Ruoli | Prescrizione e Decadenza Termini | Procedura per le rilevazioni delle condizioni di prescrizione e decadenza su cartelle non notificate | Prescrizione e Decadenza.doc |
| Ruoli | Protocollazione Flussi Telematici | Procedura per la protocollazione, la gestione ed il monitoraggio dei Flussi Telematici interscambiati fra AdR ed Equitalia | Protocollazione.doc |
| Ruoli | Provvedimenti Ruoli - Parte 1° | Procedura di acquisizione e gestione Provvedimenti di Sgravio, Sospensione e Revoca, Maggior Rateazione e Revoca, Annullamento coobbligati | Provvedimenti\_I.doc |
| Ruoli | Provvedimenti Ruoli - Parte 2° | Procedura di acquisizione e gestione Provvedimenti - Elaborazioni batch e report (tabulati) prodotti per il monitoraggio | Provvedimenti\_II.doc |
| Ruoli | Rateazioni emesse dagli AdR | Procedura per la gestione delle Rateazioni concesse dagli Agenti della Riscossione (art. 19 D.P.R. 602/1973) | Rateazioni AdR.doc |
| Ruoli | Caricamento massivo Procedure Esecutive | Procedura per la gestione (da Data Entry) delle Procedura esecutive con modalità massiva (elaborazioni batch) | Recupero\_Procedure\_Esecutive.doc |
| Ruoli | Report importi Sgravi non rimborsati | Produzione report somme non rimborsate in seguito al trattamento di Sgravi per indebito | ReportSommeDaSgravioNonRimborsate.doc |
| Ruoli | Rettifiche Versamenti | Procedura di gestione delle rettifiche automatiche dei versamenti agli Enti beneficiari | Rettifiche\_automatiche.doc |
| Ruoli | Rilevazione dati statistici | Procedura di produzione report statistici per il controllo della riscossione di importi rilevanti - Compilazione dati schede 31-ADR | Rilevazione Dati - Schede EQ.doc |
| Ruoli | Riscossione - Domiciliazioni Bancarie | Procedura per la gestione delle Domiciliazioni Bancarie per la riscossione a mezzo ruolo | Riscossione - Domiciliazione Bancaria.doc |
| Ruoli | Riscossione | Procedura per la gestione delle varie modalità di riscossione (TP Sportello, batch da flussi RAV, etc..) | Riscossione.doc |
| Ruoli | Ristampa Cartelle | Procedura dipartimentale per la Ristampa Cartelle, Avvisi GIA e Avvisi di Intimazione | Ristampa\_CartelleAvvisi\_PC.doc |
| Ruoli | Riversamento Eccedenze di Pagamento | Procedura per il controllo e la gestione del Riversamento delle Eccedenze di Pagamento | Riversamento eccedenze di pagamento.doc |
| Ruoli | Rottamazione Ruoli | Procedura per la 'Rottamazione' ruoli (ai sensi della Legge 228/2012 art.1, commi 527 e 528) | Rottamazione.doc |
| Ruoli | Ruoli vistati | Procedura per l'acquisizione e la gestione dei Ruoli telematici vistati | RuoliVistati.doc |
| Ruoli | Sospensione Titoli di Pagamento P.A. | Procedura per la gestione dei Pignoramenti presso terzi ex-Art. 48-bis DPR 602/1973, con la Sospensione dei Titoli di Pagamento Pubblica Amministrazione | Sospensione Titoli di Pagamento PA.doc |
| Ruoli | Tabelle Ruoli | Gestione Tabelle generali per decodifica Ruoli Vistati (Tipo Modello, Tipo Imposta, Periodo versamento, Stato Ricorso, Classi Tributo) | TabelleRuoli.doc |
| Ruoli | Versamento agli Enti | Procedura per la gestione dei versamenti agli Enti beneficiari | Versamento.doc |
| Ruoli A.R. | Anagrafe Ruoli A.R. | Procedura di gestione dei dati Anagrafici per ruoli ante riforma | Ruoli\_Anagrafe.doc |
| Ruoli A.R. | Archiviazione Ruoli A.R. | Procedura di gestione dell'archiviazione dei dati delle Cartelle di ruolo e relativo ripristino in caso di necessità per attività sulla Cartella archiviata | Ruoli\_Archiviazione.doc |
| Ruoli A.R. | Cartellazione Ruoli A.R. | Procedura per la gestione della Cartellazione Ruoli ante riforma | Ruoli\_Cartellazione.doc |
| Ruoli A.R. | Versamento agli Enti Ruoli A.R. | Procedura per la gestione dei riversamenti agli Enti (impositori e/o beneficiari) | Ruoli\_ContabilitàEnti.doc |
| Ruoli A.R. | Gestione del Contenzioso Ruoli A.R. | Procedura per l'attivazione del Quadro Morosi e degli adempimenti relativi alla gestione del contenzioso | Ruoli\_Contenzioso.doc |
| Ruoli A.R. | Deleghe Attive Ruoli A.R. | Procedura per l'acquisizione e la gestione delle Deleghe Attive conferite agli AdR sui Ruoli Ante Riforma | Ruoli\_DelegheAttive.doc |
| Ruoli A.R. | Domande di Rimborso Ruoli A.R. | Procedura per l'emissione e stampa delle Domande di Rimborso su Ruoli Ante Riforma - Gestione delle liquidazioni sulle D.R. presentate | Ruoli\_DomandeRimborso.doc |
| Ruoli A.R. | Prenotazioni elaborazioni batch Ruoli A.R. | Procedura di gestione delle prenotazioni di elaborazioni batch di tipo vario (ad es. Report situazione contribuenti, rendicontazione delle riscossioni, report cartelle senza obbligo, etc..) | Ruoli\_PrenotazioniVarie.doc |
| Ruoli A.R. | Riscossione Ruoli A.R. | Procedura di gestione della Riscossione Ruoli Antye Riforma con Transazioni TP (Sportello on-line, back-office, etc..) ed elaborazioni batch | Ruoli\_Riscossione.doc |
| Ruoli A.R. | Sgravi per indebito Ruoli A.R. | Procedura per la gestione degli Sgravi per indebito conferiti agli AdR su Ruoli Ante Riforma e per il recupero delle spese esecutive | Ruoli\_Sgravi.doc |
| Ruoli A.R. | Sgravi Provvisori su Ruoli A.R. | Procedura per l'emissione e la gestione degli Sgravi Provvisori su Ruoli Ante Riforma per D.R. presentate | Ruoli\_SgraviProvvisori.doc |
| Ruoli A.R. | Deleghe Passive Ruoli A.R. (SINDEL) | Procedura per la gestione delle Deleghe Passive da flussi telematici con gestione certificazioni e protocolli telematici (Sistema SINDEL-2) | Ruoli\_Sindel2.doc |
| Ruoli A.R. | Tabelle Generali per Ruoli A.R. | Gestione delle Tabelle base e dei Parametri generali della Procedura Ruoli Ante Riforma per la decodifica e/o l'indicazione comportamentale dell'intero Package | Ruoli\_Tabelle.doc |
| Ruoli A.R. | Tolleranze e Sospensioni per Ruoli A.R. | Procedura per il caricamento e la gestione delle Tolleranze e delle Sospensioni (con relativa Revoca) | Ruoli\_TolleranzeSospensioni.doc |
| Sistema | Gestione della Sicurezza | Attivazione e gestione dei moduli di sicurezza per la procedura di gestione Antiriciclaggio | RM2SICUR.doc |
| Sistema | Schede Parametro | Manuale di gestione delle Schede Parametro per la prenotazione delle elaborazioni Batch del package SET | Mbase.doc |
| Versamenti Diretti | Conto Giudiziale Versamenti Diretti | Procedura per la compilazione (tutte le Sezioni) del Conto Giudiziale delle riscossioni a mezzo Versamento Diretto | Conto\_Giudiziale.doc |
| Versamenti Diretti | Fasi fine anno Versamenti Diretti | Specifiche tecnico - funzionali per le attività di chiusura di fine esercizio e per la riattivazione del nuovo anno | Fasi fine anno.doc |
| Versamenti Diretti | Gestione Flussi telematici Versamenti Diretti | Procedura di gestione e monitoraggio di tutti i Flussi telematici utilizzati dal package (ad es. anagrafica Conto Fiscale, Agevolazioni, Versamenti Unificati, etc..) | Flussi.doc |
| Versamenti Diretti | Rimborsi in Conto Fiscale (vers. Ante 2011) | Procedura per la gestione dei Rimborsi in Conto Fiscale trasmessi in via telematica e controllo di erogabilità - Versione per Rimborsi emessi ante 2011 | Rimborsi ante 2011.doc |
| Versamenti Diretti | Rimborsi in Conto Fiscale | Procedura per la gestione dei Rimborsi in Conto Fiscale trasmessi in via telematica e controllo di erogabilità - Richieste di Rimborso IVA - Cessioni del credito | Rimborsi.doc |
| Versamenti Diretti | Servizi di Cassa (SAC) | Gestione versamenti diretti per Servizi di Cassa (SAC) - riscossioni TP Sportello e procedure Batch di acquisizione flussi deleghe di versamento da Banche | Servizi\_Cassa.doc |
| Versamenti Diretti | Versamenti Unificati | Gestione versamenti unificati tramite F24 - riscossioni TO sportello ed elaborazioni Batch per acquisizione flussi di rendicontazione Deleghe versate in Banca | Versamenti\_Unificati.doc |

1. comunque per effettuare ciò in generale è necessario accedere anche ad altri archivi, eventualmente solo in lettura. [↑](#footnote-ref-1)
2. come per i TP di gestione, può comunque essere necessario accedere anche ad altri archivi, eventualmente solo in lettura. [↑](#footnote-ref-2)
3. implementate da section di Procedure Division. [↑](#footnote-ref-3)