

CONSIP S.p.A.

ALLEGATO 8 AL CAPITOLATO TECNICO

Documentazione tecnica relativa all'applicazione ASG/LCM

**Capitolato relativo all'affidamento dei servizi di Sviluppo,
Manutenzione, Assistenza e Servizi intellettivi su aree gestionali del
Sistema Informativo del Ministero dell'Economia e delle Finanze**

***GUIDA ALLA GESTIONE
DELL'AMBIENTE A SUPPORTO DEL C.V.S. CON IL ASG /LCM***

1. INTRODUZIONE	1
2. GUIDA ALLA LETTURA DEL MANUALE	3
3. CARATTERISTICHE GENERALI DEL PRODOTTO ASG/LCM.....	5
3.1. LA GESTIONE DELLE APPLICAZIONI E LA STRUTTURA DEL DATABASE DEL PRODOTTO	5
3.1.1 <i>System</i>	6
3.1.2 <i>Configuration</i>	6
3.1.3 <i>Type</i>	7
3.1.4 <i>Member</i>	7
3.2. GESTIONE DELLE MODIFICHE AL SOFTWARE: EXPORT, IMPORT E HOST LIBRARY	8
3.2.1 <i>Version</i>	8
3.3. GESTIONE DEI PACCHETTI	9
3.3.1 <i>Restricted Package Option</i>	9
3.4. GESTIONE DELLE DIPENDENZE.....	10
3.4.1 <i>Le relazioni di reference, cross reference, impact</i>	10
3.4.2 <i>Le relazioni di build</i>	11
3.4.3 <i>Il popolamento delle relazioni e il processo di Analyze</i>	12
3.4.4 <i>L'implementazione delle relazioni di Build e la funzione di Controlled Build</i>	13
3.5. PERSONALIZZAZIONE DEL ASG/LCM NEL SISTEMA INFORMATIVO INTEGRATO DELLA RAGIONERIA GENERALE DELLO STATO E DELLA CORTE DEI CONTI.....	13
4. MAPPA DELL'AMBIENTE.....	16
4.1. IL PRODOTTO ASG/LCM.....	16
4.1.1 <i>Le librerie del prodotto</i>	16
4.1.2 <i>Gli archivi di sistema</i>	16
4.1.3 <i>La personalizzazione "nativa" del prodotto</i>	17
4.1.4 <i>L'interfaccia nativa del prodotto</i>	18
4.2. PERSONALIZZAZIONE RGS DEL PRODOTTO ED IL SOFTWARE LCMSW 1.0.....	18
4.2.1 <i>Librerie</i>	18
4.2.2 <i>I parametri del prodotto</i>	19
4.2.3 <i>La base dati del prodotto</i>	20
4.2.4 <i>L'attivazione sotto ISPF del ASG/LCM personalizzato per l'RGS/CDC</i> ..	20
4.3. LA STRUTTURA DEL DATABASE RGSCDC	21
4.3.1 <i>Aree operative e system</i>	21
4.3.2 <i>Ambienti e configurationa</i>	21
4.3.3 <i>Tipologie e type</i>	22
4.3.4 <i>Dependency relationship</i>	22
4.3.5 <i>Build relationship</i>	22
4.3.6 <i>Host library</i>	27
4.4. SICUREZZA.....	28
4.4.1 <i>Utenze, gruppi di utenze e funzioni consentite per i vari gruppi</i>	28
4.4.2 <i>Profili RACF per la protezione del database RGSCDC</i>	29
4.4.3 <i>Profili RACF per la protezione delle host library</i>	30
4.4.4 <i>L'utenza STCTRAS per la gestione dei trasferimenti</i>	38
4.5 MODIFICA AL PARAMETRO MAXIMUM NUMBER OF CCC USER	39
5. LA GESTIONE DEGLI AMBIENTI A SUPPORTO DEL C.V.S. CON IL ASG/LCM	41
5.1. PROCEDURE DI GESTIONE ORDINARIA DI LIBRERIE E ARCHIVI DI AMBIENTE	41

5.1.1	<i>Il back-up del database RGSCDC e degli archivi del prodotto.....</i>	41
5.1.2	<i>Pulizia del file di log.....</i>	41
5.1.3	<i>I back-up delle export e delle target host library</i>	41
5.1.4	<i>Procedure di pulizia delle build host library.....</i>	42
5.1.5	<i>I back-up delle librerie del ASG/LCM e dell'LCMSW 1.0.....</i>	42
5.1.6	<i>I back-up della base dati DB2</i>	42
5.2.	PROCEDURE DI GESTIONE STRAORDINARIA DI LIBRERIE E ARCHIVI DI AMBIENTE	42
5.3.	PROCEDURA DI PARTENZA E DI CHIUSURA DEL ASG/LCM	42
5.4.	LA SCHEDULAZIONE BATCH DEI TRASFERIMENTI.....	43
5.4.1	<i>Le librerie di appoggio dei trasferimenti.....</i>	43
5.4.2	<i>La struttura dei job che implementano il trasferimento di un pacchetto...</i>	45
5.4.3	<i>L'archivio di log dei trasferimenti e lo storico</i>	50
5.4.4	<i>Il log degli eventi</i>	52
5.4.5	<i>Lo stato del trasferimento.....</i>	54
5.4.6	<i>Il protocollo di attivazione del job che completa la migrate di un pacchetto sulla partizione di arrivo</i>	55
5.4.7	<i>Lo schedulatore dei trasferimenti</i>	57
5.4.7	<i>Il job HTRASCCC di schedulazione dei trasferimenti nel batch standard</i>	58
5.4.9	<i>La procedura HTRASURG di attivazione dei trasferimenti urgenti.....</i>	59
5.4.10	<i>L'esito del trasferimento e l'individuazione del punto di interruzione.....</i>	60
5.4.11	<i>Il restart del trasferimento</i>	61
5.4.12	<i>Cancellazione della richiesta di Migrate di un pacchetto</i>	68
5.5.	LA GESTIONE DEGLI UTENTI NEL ASG/LCM.....	68
5.5.1	<i>La definizione/cancellazione degli utenti al ASG/LCM.....</i>	69
5.6	LA CANCELLAZIONE DI UN OGGETTO DAL DATABASE RGSCDC.....	69
5.7	MIGLIORAMENTO DELLE PRESTAZIONI DELL'ANALYZE.....	70
6.	LA MANUTENZIONE DEGLI AMBIENTI A SUPPORTO DEL C.V.S. CON IL ASG/LCM.....	71
6.1.	DEFINIZIONE DI UNA NUOVA AREA.....	71
6.1.1.	<i>La nuova area e gli standard di nomenclatura degli oggetti nella base dati DB2</i>	74
6.2.	DEFINIZIONE DI UNA NUOVA TIPOLOGIA DI OGGETTI.....	75
6.3.	GESTIONE DEI SOTTOTIPI NELL'AMBITO DEI SOURCE	75
6.4	GESTIONE DELL'EXCLUSION LIST	77
7.	VARIAZIONI DI GESTIONE ORDINARIA DEGLI AMBIENTI GESTITI DAI PRODOTTI PROPRIETARI.....	79
7.1.	GESTIONE OGGETTI DL/I	79
7.2.	FUNZIONI ACCESSORIE APPLICATIVE.....	80
	BIBLIOGRAFIA	82
	GLOSSARIO.....	83
	NOME OGGETTO.....	93

1. INTRODUZIONE

Il presente documento costituisce la guida operativa alla gestione ordinaria e straordinaria degli ambienti a supporto del Ciclo di Vita del Software (C.V.S.) ¹, con il prodotto di mercato ASG/Life Cycle Manager ² della Allen System Group, scelto da Consip S.p.A. quale strumento di change and configuration management per l'ambiente centrale MVS.

Il punto di vista del presente manuale è duplice; le attività di gestione ordinaria e straordinaria sono descritte:

- nella sez. 5., dal punto di vista della **conduzione tecnica**, intesa come *personale tecnico sistemista preposto alla conduzione del centro*;
- nella sez. 6., dal punto di vista del **supporto sistemistico**, inteso come *personale tecnico sistemista responsabile delle attività di manutenzione adeguativa ed evolutiva necessarie a fronte di evoluzioni del software di sistema, di creazione di nuove aree di applicazione, di modifiche all'architettura tecnica delle applicazioni*

nell' ambito del sistema centrale MVS del Sistema Informativo integrato del Dipartimento della Ragioneria Generale dello Stato e della Corte dei Conti ³.

Nel manuale [1] è trattato invece il punto di vista degli **utenti** del prodotto ossia del personale tecnico (prevalentemente applicativo) *che*:

- *implementa e mantiene software per un'area operativa, per conto della quale sia abilitato ad operare*(**programmatori**);
- *è responsabile della Conduzione Funzionale o dei singoli obiettivi di sviluppo e MEV relativi ad un area operativa* (**Capi Progetto Consip**)

Questo manuale integra la documentazione tecnica del prodotto ASG/Life Cycle Manager (cfr. [2] e [3]), descrivendo le modalità operative specifiche del centro RGS-Cdc; per tutto ciò che riguarda la descrizione completa ed esaustiva delle funzionalità del prodotto, si rimanda alla sopra citata manualistica.

L'esposizione si articola come segue:

- breve guida alla lettura del manuale;
- descrizione delle principali caratteristiche del prodotto ASG/LCM e delle personalizzazioni fatte al prodotto per gestire le modalità operative e i requisiti specifici dell'ambiente RGS-Cdc non forniti dal prodotto nella sua forma "nativa";

¹ Relativamente alle fasi di realizzazione, collaudo, avviamento in esercizio e manutenzione.

² Al prodotto nel seguito ci riferiremo anche con le sigle abbreviate ASG/LCM o solamente CCC.

³ Per brevità nel seguito parleremo di progetto RGS-Cdc.

- la mappa dell'ambiente in termini di librerie e dataset del prodotto ASG/LCM e del software LCMSW di personalizzazione, di struttura del database del prodotto e delle librerie host di lavoro; di implementazione della sicurezza attraverso RACF e attraverso il ASG/LCM stesso;
- la gestione degli ambienti a supporto del C.V.S. con il ASG/LCM dal punto di vista del personale di conduzione tecnica;
- la gestione degli ambienti a supporto del C.V.S. con il ASG/LCM dal punto di vista del personale di supporto sistemistico;
- le variazioni alla gestione ordinaria introdotte dalle modifiche ai prodotti proprietari (cfr. [4]).

Chiudono il manuale, la bibliografia, il glossario e in Allegato, la base dati DB2 che integra il ASG/LCM e la modulistica.

2. GUIDA ALLA LETTURA DEL MANUALE

Il presente manuale è costituito da alcune sezioni, destinate ad una lettura *una tantum*, e da altre, destinate ad una consultazione tipicamente operativa.

In particolare sono pensate per una lettura *una tantum* le sezioni:

3. *Caratteristiche generali del prodotto:* costituisce una breve panoramica sulla filosofia del prodotto ASG/Library Configuration Management e sulle sue principali componenti, in modo da consentire un approccio graduale con il nuovo ambiente e la nuova terminologia del prodotto. La panoramica non è ovviamente completa, ma si limita a descrivere quelle caratteristiche o funzionalità che sono state sfruttate maggiormente nel progetto RGS-Cdc. L'ultima sezione è dedicata ad una breve descrizione della logica della personalizzazione del prodotto nell'ambito del progetto.
4. *La mappa dell'ambiente:* viene descritto l'ambiente da gestire in termini di librerie e dataset del ASG/LCM, librerie e base dati del software LCMSW, struttura e caratteristiche del database ASG/LCM RGSCDC e delle host library ad esso collegate, implementazione della sicurezza sia attraverso il RACF sia attraverso la sicurezza interna del ASG/LCM.

Le sezioni invece pensate per una consultazione di tipo operativo ⁴sono:

5. *La gestione degli ambienti a supporto del C.V.S con il ASG/LCM:* in cui sono descritte esaurientemente o attraverso riferimenti alla manualistica originale del prodotto e a procedure operative del centro le modalità di gestione ordinaria e straordinaria delle librerie e dataset di ambiente, i trasferimenti in termini di struttura, archivi, schedulazione, gestione degli errori e del restart, la gestione delle utenze, le modalità di cancellazione degli oggetti obsoleti.
6. *La manutenzione degli ambienti a supporto del C.V.S. con il ASG/LCM :* in cui sono descritte modalità o metodi di evoluzione dell'ambiente in termini di modalità di definizione di nuove aree, problematiche da valutare e definire per l'introduzione di nuove tipologie di oggetti, modalità di gestione dei sottotipi dei programmi soprattutto

⁴ Queste sezioni sono caratterizzate da un alto livello di strutturazione per consentire un accesso diretto all'informazione cercata, attraverso la consultazione dell'indice.

indirizzati alla comprensione del meccanismo che guida la loro compilazione.

7. Variazioni di gestione ordinaria degli ambienti gestiti dai prodotti proprietari: in cui sono descritte le variazioni nelle modalità operative dovute all'integrazione dei prodotti proprietari con il ASG/LCM

Chiudono il documento la Bibliografia, il Glossario dei termini, e, in Allegato:

- la base dati DB2 che integra il prodotto LCMSW
- la modulistica.

Ricordiamo inoltre, per una più semplice lettura, che le figure presenti nel manuale sono identificate da un codice costituito dal numero della sezione in cui si trova la figura, seguito da un progressivo alfabetico di figura all'interno della sezione (per esempio la terza figura della sez.4.2 sarà contrassegnata dal codice 4.2.c); in tal modo è più facile ritrovarle nel corpo del documento, quando, in una sezione si fa riferimento a figure presenti in sezioni differenti.

3. CARATTERISTICHE GENERALI DEL PRODOTTO ASG/LCM

Il ASG/Lyfe Cycle Manager è un prodotto di mercato di Change and Configuration Management in ambiente mainframe TSO/ISPF, che, attraverso una interfaccia a pannelli, controlla le modifiche apportate al software e regola il processo di sviluppo, facilitando il trasferimento controllato delle modifiche alle applicazioni nei differenti ambienti che implementano le differenti fasi del C.V.S.

Le funzioni del prodotto sono sostanzialmente divise in quattro componenti principali:

- la gestione delle applicazioni;
- la gestione delle modifiche al software;
- la gestione dei pacchetti di modifiche;
- la gestione delle relazioni e delle dipendenze fra gli oggetti software.

Nel seguito è fornita una succinta descrizione di queste componenti, rimandando alla manualistica del prodotto per un quadro completo ed esaustivo delle stesse (cfr.[2]).

3.1. LA GESTIONE DELLA APPLICAZIONI E LA STRUTTURA DEL DATABASE DEL PRODOTTO

L'informazione nel CCC è identificata ed organizzata come una gerarchia di strutture dati. Queste strutture dati sono:

- il database CCC
- le strutture dati SYSTEM
- le strutture dati CONFIGURATION
- le strutture dati TYPE
- le strutture dati MEMBER

La figura 3.1.a. illustra la struttura della gerarchia CCC:

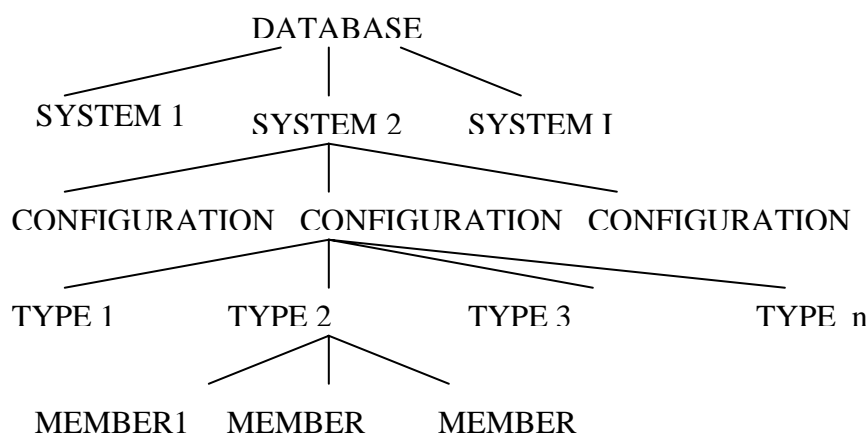


Fig. 3.1.a
La gerarchia del CCC

Il livello database del CCC è la radice della gerarchia e comprende tutta l'informazione controllata. Possono esserci molti database CCC su una macchina, ma un utente può operare su un solo database per volta e il software memorizzato in un database non è condivisibile con il software memorizzato in un database differente.

3.1.1 System

Il CCC consente di memorizzare applicazioni indipendenti nelle strutture dati SYSTEM (o sistemi). Per esempio, ogni system può contenere una distinta applicazione da svilupparsi o mantenere.

Un system dovrebbe consistere di tutti i componenti di un'applicazione che possano essere «congelati» come parte di un release. Per esempio, supponiamo che un grande sistema software consista di tre differenti sottoapplicazioni. Ognuna contiene i suoi propri programmi sorgente, moduli load, JCL, documentazione, etc.

I tre pezzi costituiscono un'unica versione rilasciata del sistema, che è l'applicazione software rilasciata. Nell'ambito del CCC, tutte e tre le sottoapplicazioni dovrebbero essere poste in un unico system.

Usando lo stesso esempio, se i sistemi software rilasciati consistono di differenti combinazioni di queste sottoapplicazioni, è importante tenere traccia e congelare un release di ogni sottoapplicazioni indipendentemente. In tal caso, ogni sottoapplicazione dovrebbe costituire un particolare system, avendo così tre system in totale.

3.1.2 Configuration

Le CONFIGURATION (o configurazioni o, nel nostro caso, ambienti) si collocano nella gerarchia, immediatamente al di sotto dei

system e possono essere usate per contenere al loro interno tutti gli oggetti di un'applicazione (system):

- in una loro specifica versione, quale potrebbe essere per esempio quella associata ad una determinata piattaforma hardware e software,
- in una specifica fase del ciclo di vita del software (i.e. sviluppo, test, collaudo, esercizio, manutenzione),

per generare, identificare, tener traccia e controllare versioni differenti di SYSTEM.

Le CONFIGURATION permettono agli utenti di organizzare e gestire tutte le componenti del release o della base di partenza di un'applicazione come fossero una singola entità, così che le relazioni tra componenti siano sempre conservate ed identificabili.

L'architettura del prodotto fa sì che le configurazioni debbano essere create come gerarchicamente dipendenti da una configurazione di partenza («baseline») costituita da una vista onnicomprensiva del software dell'applicazione, ottenuta tramite un caricamento iniziale di una versione consolidata di tale software.

In ciascuna configurazione create a partire da una medesima baseline, non si ha solo la visibilità degli oggetti in essa modificati rispetto alla baseline, ma si ha la visibilità completa di tutti gli oggetti sw che compongono l'applicazione(system) e delle relazioni di dipendenza reciproca.

Una volta che, a partire da una configurazione di baseline, sono derivate altre configurazioni nelle quali implementare sviluppi e manutenzioni, la baseline risulta "congelata" e non può più essere modificata.

3.1.3 Type

Le strutture dati TYPE sono le strutture utilizzate nel costruire l'architettura del software applicativo in una configurazione (sorgenti, copy, job, mappe, etc.) e sono paragonabili ai TYPE definiti come terzo qualificatore dei data-set partitioned MVS.

3.1.4 Member

La funzione delle strutture dati MEMBER (o membri) è di memorizzare informazioni e sono paragonabili ai membri delle librerie MVS. Ogni tipo di informazione leggibile dalla macchina può essere memorizzata nei membri. Non c'è un limite pratico al numero di membri posto in un TYPE.

I membri contengono le informazioni realmente modificabili. Le modifiche sono eseguite esportando un membro dal CCC sull'host, dove è modificato, e importandolo poi indietro nel CCC.

Si usa il termine Configuration Item (CI), o semplicemente item, per riferirsi alla combinazione TYPE e MEMBER. Un CI è l'unità base di memorizzazione dell'informazione nel database.

3.2. GESTIONE DELLE MODIFICHE AL SOFTWARE: EXPORT, IMPORT E HOST LIBRARY

La gestione delle modifiche consiste in primo luogo:

- nella cattura delle variazioni alle informazioni memorizzate nel database (Configuration item),
- nel controllo dell'accesso a tali informazioni,
- nella tracciabilità delle modifiche in termini di chi le ha effettuate, quando e perché.

La filosofia del ASG/LCM prevede che la fase di editing dei configuration item avvenga al di fuori del database stesso, su dataset esterni che vengono detti (export) host dataset.

La funzione che estrae gli oggetti dal database verso le host library si definisce EXPORT, mentre IMPORT è la funzione inversa, che consente quindi di riportare gli oggetti modificati all'interno del database.

L'Export di un oggetto può essere di due tipi:

- *for update* che lascia l'oggetto esportato in stato di EXPORTED e nessun altro può esportarlo nuovamente per UPDATE, così come nessuno può effettuarne l'IMPORT se non l'utente stesso che ne ha fatto l'EXPORT;
- *for read* che lascia l'oggetto in stato NORMAL.

Anche l'Import può essere di due tipi:

- *Intermediate* che lascia l'oggetto importato in stato di EXPORTED;
- *Complete* che riporta l'oggetto in stato NORMAL.

Al momento dell'import il prodotto richiede di specificare un *change name* e un eventuale testo esplicativo da associare alla nuova versione salvata dell'oggetto.

3.2.1 Version

Ogniqualvolta, attraverso l'import, sono effettuate delle modifiche ad un membro, il ASG/LCM intercetta tutte le differenze rispetto alla precedente versione del membro e memorizza la nuova versione, sotto forma di variazioni rispetto alla versione precedente. La data, l'ora e

l'autore della modifica sono automaticamente registrati con la nuova versione.

L'utente ha l'opportunità di inserire un CHANGE NAME ed una breve descrizione del perché sia stata effettuata la modifica. La documentazione di una modifica è associata direttamente all'informazione che si sta modificando e non può essere persa o messa fuori posto. Come risultato, i programmatori sono sollevati da un tedioso compito di registrazione e la reportistica diventa più flessibile.

Il numero di versioni di un configuration item può essere infinito ed il prodotto non fornisce alcuno strumento parametrico ed automatico per limitarne o controllarne il numero.

3.3. GESTIONE DEI PACCHETTI

Il pacchetto di modifiche, o semplicemente pacchetto, è il concetto centrale nella strategia di gestione a pacchetti del CCC. Un pacchetto è un gruppo di membri all'interno del CCC migrati insieme da una configurazione CCC ad un'altra e, se specificato, alle corrispondenti *host library*.

I pacchetti sono definiti specificando dei criteri di selezione, consistenti di nomi di membri e change names, dove un change name è un nome associato ad una particolare versione di un membro.

Ad un pacchetto possono essere associati dei *migration path*. Un *migration path* permette di definire la sequenza di configurazioni attraverso cui il pacchetto deve essere migrato.

E' inoltre possibile indicare requisiti di approvazione da parte di singoli utenti o di gruppi di utenti per specifici migration path, che se non soddisfatti, impediscono la migrazione del pacchetto lungo il path indicato.

La migrazione di un pacchetto è inoltre reversibile (fall back).

3.3.1 Restricted Package Option

La Restricted Package Option (RPO) è una specifica opzione a livello di intero prodotto, che se attivata, definisce più chiaramente come possono essere usati i pacchetti, fornendo maggiore controllo sull'aggiornamento delle CONFIGURATION dei SYSTEM.

- Ad un pacchetto viene associato uno stato, determinato dal nome di una CONFIGURATION. Alla creazione del pacchetto deve essergli assegnato uno stato iniziale, correlato all'insieme di stadi del ciclo di vita definito per quel pacchetto tramite il *migration path*. Al pacchetto sarà quindi assegnata come stato la CONFIGURATION iniziale del *migration path*.

Lo stato del pacchetto cambierà ogni volta che esso sarà migrato. Non sarà possibile migrare un pacchetto che si trova in un determinato stato verso una CONFIGURATION che non segua quello stato nel *migration path*, né sarà possibile effettuare delle modifiche ad un pacchetto in una CONFIGURATION che non sia quella indicata dal suo stato.

- È possibile definire delle *Import Rules* con cui controllare ulteriormente gli aggiornamenti agli item di una CONFIGURATION. In particolare, è possibile permettere od impedire qualunque import di item nella CONFIGURATION, o restringere selettivamente l'import ad alcuni TYPE.
- Con l'RPO attiva, è obbligatorio indicare al momento dell'import di un oggetto, anche il nome del pacchetto cui l'oggetto appartiene, oltre al change name da associare alla versione e non viene effettuato l'import se l'oggetto non risulta fra quelli assegnati al pacchetto.

3.4. GESTIONE DELLE DIPENDENZE

Ogni applicazione scritta con codice modulare ha una rete interna di dipendenze. Fondamentalmente, una tale rete consiste di relazioni tra componenti dell'applicazione.

Per esempio, una tipica applicazione Cobol include componenti come moduli source, COPY, moduli assembler, mappe CICS. Il Cobol può essere suddiviso in routines che eseguono call statiche risolte negli step di compilazione e link-edit, e routine che eseguono call dinamiche, risolte in fase di esecuzione.

Un esempio di rete di dipendenze tra questi componenti è illustrato in Fig.3.4.a. In questa figura, le frecce rappresentano le relazioni tra i componenti. Ci sono quattro differenti tipi di frecce, usate per rappresentare i quattro tipi base di relazioni disponibili.

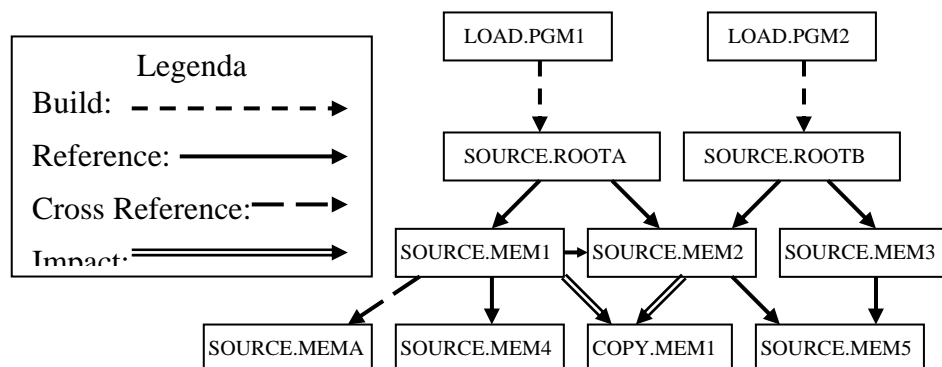


Fig. 3.4.a
Esempio di rete di relazioni

3.4.1 Le relazioni di reference, cross reference, impact

In una **relazione di reference**, un item invoca, chiama, referencia o comunque usa un secondo item. L'item che effettua l'invocazione è noto come componente primario, e l'item invocato è chiamato componente secondario. Entrambi i componenti nella relazione sono necessari per costruire il target. Quando si modifica un componente secondario, il componente primario non diventa obsoleto.

Un esempio è una routine Cobol fa un riferimento statico ad un sottoprogramma Cobol:

CALL '<nome-programma>'

Una **relazione di cross reference** è simile ad una relazione di reference in cui un item invoca, chiama, referencia o comunque usa un secondo item. D'altra parte, in una relazione di cross reference l'item riferito non è richiesto per costruire il target. Quando si modifica un componente secondario, il componente primario non diventa obsoleto. Un esempio è un main CICS cobol che richiama dinamicamente una routine CICS cobol:

EXEC CICS LINK PROGRAM('<nome-programma>')

Quando esiste una **relazione di impact** tra due item, una modifica al componente secondario rende obsoleto il componente primario. La relazione tra una routine source Cobol ed una COPY è un esempio di un particolare tipo di relazione di impact.

Esempi:

- un source Cobol copia una COPY: COPY <nome-copy>
- un source C include un file di include:
#INCLUDE <nome-include>

Le relazioni di reference e di impact sono denominate anche relazioni componente/componente.

La definizione delle relazioni è un'attività di personalizzazione del prodotto, che deve essere replicata per ogni system (leggi applicazione o area). In questa fase, quello che si definisce è una relazione generale che può intercorrere fra differenti type dello stesso o di differenti system definiti *cross-system* (i.e. un qualsiasi SOURCE dell'applicazione Bilancio può avere una relazione di IMPACT con una qualsiasi COPY dell'applicazione Bilancio o dell'applicazione Spese), senza doverla istanziare a livello di singoli membri di tali type.

3.4.2 Le relazioni di build

Una relazione di **build** (altrimenti nota come relazione target/componente) è definita come segue: ogni qualvolta uno o più item componenti sono messi insieme per formare un singolo item target, si dice che gli item sono coinvolti in una relazione di build.

Esempi:

- dei moduli oggetto (componenti di build) sono link-editati insieme per creare un modulo load (target);

- un programma sorgente (il componente di build) è compilato e sottoposto a link-edit per creare un modulo eseguibile (il target);
- un panel group e un panel SDF II sono elaborati per dar luogo alla copy della mappa (MAPDSECT) e al MAPSET.

La relazione di build, che genera un target a partire da uno o più componenti primari della build, è fisicamente implementata da una *build procedure*.

Questa è una procedura utente, specificata al momento della definizione della relazione di build (nella fase di personalizzazione del prodotto, iterata a livello di system), che contiene le istruzioni necessarie (per esempio jcl di compilazione e link-edit) per generare, a partire dai componenti primari (i.e. un sorgente COBOL), il target previsto (il modulo eseguibile).

3.4.3 Il popolamento delle relazioni e il processo di Analyze

L'analyze è una funzione che l'utente può attivare interattivamente e che gli consente di popolare le relazioni sopra descritte con le occorrenze relative ai singoli item dell'applicazione.

L'analyze può partire o da un insieme di item modificati o da un target.

Analyze a partire da un insieme di item modificati

Il processo di analisi serve a determinare l'impatto dei configuration item modificati per un pacchetto sugli oggetti target del system.

L'analyze parte da tutti i configuration item per i quali esiste una versione contrassegnata da un change name che è parte del pacchetto.

Analyze a partire da un target

Il processo di analisi serve a determinare l'insieme di configuration item che concorrono a costituire un target, partendo dai componenti primari della relazione di build per giungere a tutti i componenti a questi correlati direttamente o indirettamente attraverso relazioni di impact e reference.

L'output dell'analisi è memorizzato nel database CCC associato alla work-area indicata dall'utente all'atto dell'attivazione della funzione.

3.4.3.1 *Le dependency work area*

Una dependency work area (o più semplicemente una work-area) è lo strumento che il prodotto utilizza per raggruppare e memorizzare le informazioni relative alle dipendenze fra CI.

La work area fornisce un ambiente comune a più utenti consentendo di specificare una volta per tutte una lista di change name che devono guidare il processo di analyze e le configurazioni dei cross system in cui effettuare la ricerca degli eventuali oggetti coinvolti nella rete di relazioni, ma facenti parte di applicazioni differenti.

La work area può essere:

- specifica per un utente (che indicherà la sua userid come nome della work area) nel caso in cui non si vogliano condividere con alcuno i risultati del proprio processo di analyze;
- specifica di configurazione (per consentire a tutti gli utenti che operano in un dato ambiente di condividere i risultati del loro processo di analyze);
- specifica di pacchetto (in questo caso deve essere omonima rispetto al pacchetto) per indirizzare il processo di analyze esclusivamente sugli oggetti modificati direttamente o indirettamente nell'ambito di un pacchetto di modifiche. In questo caso infatti la lista di change name che deve guidare il processo di analyze è automaticamente valorizzata dal prodotto con i change name associati al pacchetto.

3.4.4 L'implementazione delle relazioni di Build e la funzione di Controlled Build

Il processo di Controlled BUILD (o più semplicemente di build) ha come oggetto uno o più target:

- forniti in input esplicitamente dall'utente;
- individuati dal processo di analyze eseguito a partire da un insieme di item modificati o direttamente da un insieme di target.

La funzione prende input dai dati dell'analyze associati alla work area specificata in input alla funzione e ne utilizza le informazioni per:

- individuare tutti i configuration item che sono necessari per costruire un target (sorgenti, copy, INCLUDE, etc.)
- estrarli dal database del ASG/LCM appoggiandoli su host library utilizzate dal processo di BUILD.

Esegue quindi un job costruito a partire dalla build procedure associata al target type oggetto della build, al termine del quale il target item risulterà generato.

3.5. PERSONALIZZAZIONE DEL ASG/LCM NEL SISTEMA INFORMATIVO INTEGRATO DELLA RAGIONERIA GENERALE DELLO STATO E DELLA CORTE DEI CONTI

Il prodotto ASG/LCM con le sue funzionalità native descritte alle sezioni precedenti, non si adatta completamente alle richieste e alle modalità di lavoro specifiche del progetto RGS-Cdc.

In particolare:

- non è strutturato per operare contemporaneamente sui medesimi oggetti in due ambienti diversi (quali sono Sviluppo e Manutenzione) e quindi non effettua in maniera nativa nessun controllo reciproco su eventuali attività che si intersechino in tali ambienti⁵;
- ogni configurazione nell'ambito di un sistema ha la visibilità completa degli oggetti che appartengono al sistema, e non solo di quelli che sono oggetto della fase del C.V.S. identificata dalla configurazione in cui operano; i trasferimenti da una configurazione ad un'altra non muovono, bensì copiano gli oggetti del pacchetto dall'ambiente di partenza a quello di arrivo; quindi quando un oggetto è trasferito in esercizio proveniente da sviluppo e collaudo, la nuova copia dell'oggetto non è automaticamente aggiornata dal prodotto "nativo" nell'ambiente alternato di manutenzione, e viceversa per i trasferimenti da manutenzione verso esercizio;
- quando un pacchetto è trasferito da una configurazione ad un'altra (per esempio da sviluppo a collaudo), il ASG/LCM "nativo" consente la modifica degli oggetti appartenenti al pacchetto nella configurazione di partenza (leggi sviluppo) per conto di un altro pacchetto, senza attendere il completamento del ciclo di vita del pacchetto (ossia il trasferimento in esercizio) perché ciò avvenga;
- ad ogni tipo di oggetti (per esempio sorgenti) che intervengono come componenti primari in una relazione di build è associata una ed una sola build procedure atta ad implementare la relazione. Questo fa sì che, se si vogliono compilare in maniera differente sorgenti diversi (per esempio un cobol batch DL/I e un main CICS DB2), l'unica soluzione fornita dal prodotto nativo è quella di definire tipi differenti di oggetti (per esempio SBTDLI e SMNCXD2); questo rende:
 - impossibile realizzare elenchi completi di programmi per un'area; complesso reperire un oggetto nel database in quanto per operare su esso è necessario indicare il type di appartenenza che risulta di difficile memorizzazione;
 - rischiosa la creazione di nuovi programmi in quanto l'unico controllo di univocità nativo che il prodotto svolge è nell'ambito del `system.configuration.type`.

⁵ Per esempio verificare, quando si opera in sviluppo su un oggetto, che non ci siano su esso attività contemporanee di manutenzione ed avvisare, in caso negativo, gli utenti coinvolti.

Per rendere quindi il prodotto conforme alle esigenze e alle modalità operative del progetto sono state fatte delle personalizzazioni al codice del prodotto e degli sviluppi integrativi di software al contorno.

Questi interventi che denomineremo nel seguito "personalizzazioni RGS al ASG/LCM" hanno consentito di:

- verificare la presenza di interventi di aggiornamento negli ambienti alternati rispetto a quelli in cui si opera (manutenzione se si opera in sviluppo o si trasferisce da sviluppo in collaudo e da collaudo in esercizio, sviluppo e collaudo se si opera in manutenzione o si trasferisce da manutenzione verso esercizio);
- controllare l'univocità degli oggetti che vengono creati in sviluppo o in manutenzione, nonché l'aderenza agli standard di nomenclatura specifici dell'area (system) di appartenenza dell'oggetto;
- dato un configuration item, modificato per conto di un pacchetto in un dato ambiente (sviluppo o manutenzione), impedire nel medesimo ambiente la modifica dell'oggetto per conto di altri pacchetti fino al corretto completamento del trasferimento del pacchetto in esercizio;
- consentire la compilazione di un programma personalizzata in funzione delle sue caratteristiche (cobol o assembler, batch o tp, DB2, DL/I, C/S, main o subroutine, etc) pur definendo tutti i programmi come facenti parte di un unico type (SOURCE);
- garantire, all'atto del trasferimento dei pacchetti, tutte quelle operazioni di allineamento dei sottosistemi, delle librerie, della documentazione, delle configurazioni ASG/LCM alternate sia sulla partizione MVS di partenza del trasferimento che su quella di arrivo;
- rendere maggiormente usabile il prodotto nel colloquio interattivo minimizzando il numero di dati da introdurre, preimpostando, proteggendo, inibendo quelle scelte non significative o pericolose per la modalità di utilizzo del prodotto nel progetto.

4. MAPPA DELL'AMBIENTE

Questa sezione dà una panoramica dell'ambiente del ASG/LCM rilasciato con il presente progetto.

4.1. IL PRODOTTO ASG/LCM

In questa sezione è descritto l'ambiente rilasciato del ASG/LCM in termini di:

- librerie del prodotto,
- archivi del prodotto stesso,
- personalizzazioni native del prodotto (per esempio RPO attiva, CDO disattivata, etc.),

4.1.1. Le librerie del prodotto

Si riportano di seguito le librerie prodotto ASG/LCM create in fase di installazione.

Sono tutte allocate sul volume CCCS01.

CCCLCMS.V30D.CCCCLIST
CCCLCMS.V30D.CCCEXLIB
CCCLCMS.V30D.CCCHHELP
CCCLCMS.V30D.CCCHLP01
CCCLCMS.V30D.CCCHLP02
CCCLCMS.V30D.CCCHLP03
CCCLCMS.V30D.CCCHLP04
CCCLCMS.V30D.CCCHLP05
CCCLCMS.V30D.CCCHLP06
CCCLCMS.V30D.CCCINSTL
CCCLCMS.V30D.CCCLOAD
CCCLCMS.V30D.CCCMACRO
CCCLCMS.V30D.CCCMLIB
CCCLCMS.V30D.CCCPARM
CCCLCMS.V30D.CCCPLIB
CCCLCMS.V30D.CCCREXX
CCCLCMS.V30D.CCCSLIB
CCCLCMS.V30D.CCCSYMBL
CCCLCMS.V30D.PROCLIB

4.1.2. Gli archivi di sistema

Gli archivi di sistema, sia VSAM che non VSAM, sono creati contestualmente alla creazione del database di CCC e hanno il prefix specificato nei parametri 'DATABASE VSAM QUALIFIER' e NON VSAM QUALIFIER' del pannello CCC.DB.1

I suddetti parametri sono impostati rispettivamente a 'H0VS' e 'H0TS'.

I non VSAM sono i seguenti

H0TS.RGSCDC.INDEX
H0TS.RGSCDC.INITIAL
H0TS.RGSCDC.LOG
H0TS.RGSCDC.LOOKAHED
H0TS.RGSCDC.DYYYYMMDD.THHMMSS

I VSAM sono i seguenti:

CCCLCMS.V30D.RGSCDC.CCCNOT
H0VS.RGSCDC.DATABASE che rappresenta fisicamente la base
dati di CCC.

4.1.3. La personalizzazione "nativa" del prodotto

L'installazione è stata fatta utilizzando i parametri di installazione
riportati in tabella

NOME	VALORE
Prefix:	CCCLCMS.V30D
Product tape volser	LCM30D
Tape drive device type	3480
SVC number	235
Enter Y if CCC to use SMS managed data sets	N
Target DASD unit type or Storage Class	3380
Target DASD volser or Management Class	CCCS01
System PROCLIB	CCCLCMS.V30D.PROCLIB
Data base manager started task name	DBSTC
IPC started task proc name	IPC
SYSOUT message class	X
Load module library name	CCCLCMS.V30D.CCCLOAD
Create new or delete and replace old loadlib?	Y
Load library or linklist?	N
SYSOUT printer output class	A
Authorized copy program name	IEBCOPY
Maximun number of CCC users	30-->99
Delete old target data sets?	N
Allocate new target data ses?t	Y
Date format	DMY
Memory work space increment in pages	256

Maximum increments	32
SMF record number	
SMF types	
ELCM started task proc name	
Notification prefix	CCCLCMS.V30D
Security Interface	RACF
Product or Trial Passwords	Y
CPU password	N

Per il significato di ogni singolo parametro di rimanda al manuale (cfr. [2]).

Tra i parametri specificati in fase di installazione vi è la Restricted Package Option settata a ON e la Concurrent Development Option disattivata.

4.1.4. L'interfaccia nativa del prodotto

L'interfaccia nativa del prodotto si attiva con l'opzione 9.C della procedura di logon ASYSCCC e per il suo utilizzo si rimanda al manuale ASG/LCM "USER'S GUIDE".

4.2. PERSONALIZZAZIONE RGS DEL PRODOTTO ED IL SOFTWARE LCMSW 1.0

4.2.1. Librerie

Per adeguare il prodotto ASG/LCM all'ambiente Rgs/Cdc sono stati personalizzati dei moduli del prodotto stesso, posti nelle librerie di cui alla sez. 4.2.1.1, e sono state effettuate delle integrazioni con moduli scritti ex novo, che costituiscono il software LCMSW 1.0, posto nelle librerie di cui alla sez. 4.2.1.2.

4.2.1.1. Librerie di personalizzazione RGS del prodotto

Clist TSO: CCCLCMS.V30D.CCCCLIST.USERAPPL
Macro: CCCLCMS.V30D.CCCMACRO.USERAPPL
Messaggi ISPF: CCCLCMS.V30D.CCCMLIB.USERAPPL
Pannelli ISPF: CCCLCMS.V30D.CCCPLIB.USERAPPL
Clist REXX: CCCLCMS.V30D.CCCREXX.USERAPPL
Skeleton ISPF: CCCLCMS.V30D.CCCSLIB.USERAPPL

4.2.1.2. Librerie del software LCMSW 1.0

Build procedure:	C0T4.LCMSW10.BLDPROC
Clist TSO	C0T4.LCMSW10.CLIST
JCL di compilazione e gestione del prodotto:	C0T4.LCMSW10.CNTL
Copy COBOL e Assembler, macro Assembler e DCLGEN della base dati:	C0T4.LCMSW10.COPY
DBRM:	C0T4.LCMSW10.DBRM
DDL, DCL e DML per la definizione della base dati e il caricamento iniziale delle tabelle di decodifica:	C0T4.LCMSW10.DDL
Eseguibili del prodotto:	C0T4.LCMSW10.LOAD
Messaggi ISPF:	C0T4.LCMSW10.MSG
Pannelli ISPF:	C0T4.LCMSW10.PANEL
Procedure di trattamento:	C0T4.LCMSW10.PROC
Scanner:	C0T4.LCMSW10.SCANNER
Skeleton ISPF:	C0T4.LCMSW10.SKELE
Sorgenti:	C0T4.LCMSW10.SOURCE
Schede di controllo:	C0T4.LCMSW10.SYSIN

Si noti che le librerie di LOAD, PROC e SYSIN devono risiedere su dischi comuni, in quanto devono essere visibili anche nelle partizioni di PLT2 ed Esercizio.

4.2.2. I parametri del prodotto

L'ambiente è descritto al prodotto, così come risulta nella personalizzazione RGS/Cdc, tramite una serie di parametri, che sono presenti nella clist LCCUTPAR, contenuta nella libreria C0T4.LCMSW10.CLIST, e nei seguenti membri della libreria C0T4.LCMSW10.SKELE:

LCSLMPAR
LCSLSPAR
LCSMCPAR
LCSMEPAR
LCSMIPAR
LCSMMPAR
LCSMSPAR
LCSPCPAR
LCSPEPAR
LCSPMPAR
LCSPSPAR
LCS0CPAR
LCS0EPAR

4.2.3 La base dati del prodotto

La base dati del prodotto è costituita da tabelle DB2, raggruppate nel database DDLCM000 definito sul sottosistema DB2 di sviluppo, e da archivi sequenziali definiti su dischi condivisi a tutte le partizioni. Per quanto riguarda la descrizione di tale base dati, si rimanda all'Allegato 1.

4.2.4 L'attivazione sotto ISPF del ASG/LCM personalizzato per l'RGSC/CDC

Per poter attivare la versione personalizzata del prodotto sotto ISPF, è necessario:

- utilizzare una procedura di logon che abbia concatenata in testa alla SYSPROC la libreria CCCLCMS.V30D.CCCCLIST.USERAPPL
- associare ad una voce del pannello omonimo della suddetta procedura, contenuto nella libreria ISP.PANCUST, il richiamo alla clist CCCINVK, con parametri CCCLCMS.V30D e H0T4.LCMSW, come in Fig. 4.2.4.b.

Prodotti di sviluppo e funzioni di utilita'			e
Option ==>			e
		More: +	e
8	SUPPORTI	Prodotti centralizzati	e
9	GESTIONE	Funzioni di utilita'	e
C	ASG/LCM	CCC Lyfe Cycle Manager	e
D	DB2	Relational database environment	e
F	APPL	Funzioni accessorie applicative	e
I	DFSORT	DFSORT release 13	e
M	SMARTTEST	Test - Debug application program	e
SD	SDF II	Screen Definition Facility II	e
S	SYSCONF	Informazioni schedulazione sistemi	e
V	ABEND-AID	Viewing Facility ABEND-AID	e
SO	SYSOUT	Visualizzazione sysout	e
	F1=Help	F3=Exit F10=Actions F12=Cancel	e

Fig. 4.2.4.a
Esempio di pannello con attivazione del prodotto ASG/LCM


```

&ZSEL = TRANS( TRUNC (&ZCMD, '.' )
      8, 'PANEL(WXXXX00)'
      9, 'PANEL(KFUGP02M) NEWPOOL'
      C, 'CMD(%CCCINVK CCCLCMS.V30D HOT4.LCMSW)'
      D, 'PANEL(ISRDB2)'
      F, 'PGM(HGXXX00) PARM(&ZALTTR) NEWAPPL(HGIS) NEWPOOL NOCHECK'
      I, 'CMD(%R13LDEF &ZCMD) NOCHECK NEWAPPL(ICE) PASSLIB'
      M, 'CMD(ISPFVIA) NEWAPPL(VIAC)'
      S, 'PANEL(KSXXX00)'
      SD, 'CMD(SDF2DRIV)'
      V, 'PGM(CWDDSUTL) NEWAPPL(AAUT) PARM(SPF)'
      SO, 'CMD(%XSYSOUT)'
      BM, 'CMD(%ISPFBOOK)'
      FM, 'CMD(%FAIDMVS &ZTRAIL) SCRNAME(COMPAPPL) NOCHECK'
      ' ' ' ' ' '
      *, '?' )

```

Fig. 4.2.4.b

Valorizzazione di ZSEL nella sezione PROC del pannello precedente, con l'attivazione del prodotto ASG/LCM

4.3. LA STRUTTURA DEL DATABASE RGSCDC

Come riportato al cap. 5 della specifica di analisi, le informazioni all'interno del DB di CCC sono organizzate in maniera gerarchica e si puntano specificando le seguenti informazioni:

nome Database
 nome System
 nome Configuration
 nome Type
 nome Member

4.3.1 Aree operative e system

I system corrispondono all'area applicativa e contengono al loro interno più configuration. Ogni system contiene tutto il software di competenza dell'area corrispondente.

Un nuovo system si crea tramite l'opzione CCC.1.4 e con il parametro C=copy si può utilizzare a modello il template 'xxx' già predisposto a contenere le caratteristiche comuni a tutti i system. Una volta creato il nuovo system dal template occorre personalizzarlo mediante le opzioni CCC.3.1 (system set up) e CCC.4.S (set up system).

Le opzioni CCC.3.1.1 (cfr par 4.3.4) e la CCC.3.1.2. (cfr 4.3.5.3) sono quelle da personalizzare per adeguare il template al nuovo system.

Stessa attività occorre fare con le opzioni CCC.4.S.1 e CCC.4.S.2.

Per una descrizione dettagliata sulla modalità di creazione di una nuova area si rimanda al paragrafo 6.1.

4.3.2 Ambienti e configurationa

I configuration corrispondono agli ambienti operativi, SVIL, MAN, COLL e ESER di pertinenza dell'area. L'unico configuration previsto

nel system template è BASELINE dove andrà inizialmente caricato il software dell'area. I configuration d'ambiente saranno creati in seguito prendendo a modello il configuration BASELINE.

4.3.3 Tipologie e type

I Type rappresentano le varie tipologie di software presente sul database RGSCDC. La lista dei type è riportata nella tabella (Fig 5.3.a) di pag 23 del Manuale d'uso. I type vengono 'ereditati' dal system template al momento della creazione di un nuovo system.

4.3.4 Dependency relationship

Le dependency relationship vengono gestite con l'opzione CCC.3.1.1. Una volta usato il template per la definizione di un nuovo sistema, le relazioni 'ereditate' dal template vanno aggiornate con quelle caratteristiche del sistema.

Alcune informazioni da specificare sono riportate nei sottoparagrafi seguenti; per un dettaglio maggiore si rimanda alla manualistica di ASG/LCM.

4.3.4.1 Sistemi condivisi (cross system)

Nella definizione delle dependency relationship di un system è necessario specificare eventuali system con i quali condivide oggetti.

4.3.4.2 Gli scanner

Con l'opzione 3.1.6.1 è possibile specificare gli scanner da utilizzare per i type coinvolti.

Essi sono:

SCANSRC

SDGIGRP

e devono essere assegnati con l'opzione 3.1.6.3 seguendo il seguente schema:

SOURCE SCANSRC

COPY SCANSRC

INCLUDE SCANSRC

DGIGRP SDGIGRP

PSBSRC SCANSRC

4.3.5 Build relationship

4.3.5.1 Build procedure

Il nome di una relazione di build è dato da

Bxxxxxxx

dove:

B carattere fisso che indica la tipologia della relazione (B → Build)

xxxxxxx type CCC del target generato dalla relazione di build (i.e. BLOAD è la relazione di build che da un SOURCE genera un LOAD)

Il nome di una *build procedure* associato ad una relazione di build è

Jxxxxxxx

dove:

J carattere fisso

xxxxxxx type CCC del target generato dalla build procedure (i.e. JLOAD è la build procedure che da un SOURCE genera un LOAD)

Le BUILD procedure sono memorizzate nel database ASG/LCM nel type dedicato BLDPROC, valorizzate già a livello di BASELINE del system template XXX, invarianti rispetto al SYSTEM e alla CONFIGURATION.

Con questi oggetti il ASG/LCM costruisce il job di BUILD (compilazione) facendone il file tailoring ISPF e facendolo precedere dai necessari step di estrazione (scompress) dei configuration item coinvolti nel processo di build.

Le build procedure Jxxxxxxx al loro interno contengono le seguenti istruzioni

```
)IM LCSx&AMB.&SUFFL
)SEL &AMB = E ! &AMB = C
)IM LCS0&AMB.TRA
)ENDSEL
```

dove:

&AMB	può assumere i valori:	S	SVIL
		C	COLL
		E	ESER
		M	MAN

x	può assumere i valori:	L	LOAD
		M	MAPSET
		P	PSBLOAD

&SUFFL può assumere i valori: 000
system-name (non usato)

istruzioni che generano il file tailoring di skeleton proprietari il cui nome è parametrizzato in funzione:

- della configurazione,
- del target da generare

- se necessario, del system.

I valori delle variabili ISPF che concorrono a generare lo skeleton che realmente implementa le BUILD Procedure sono valorizzati nei *parameter set* del prodotto (cfr. sez.4.3.5.2).

Gli skeleton richiamati dalle BUILD procedure sono appoggiati sulla libreria

HOT4.LCMSW.SKELE

concatenata in ISPSLIB nello skeleton DBOTM3 (cfr. personalizzazione RGS del prodotto alla sez.4.2).

In questo modo la manutenzione delle build procedure avviene al di fuori del prodotto ASG/LCM facilitandone la gestione e la centralizzazione.

Negli ambienti SVIL e MAN, la build procedure opera esclusivamente a livello di target item per il quale è stata attivata, effettuandone la compilazione ed altre operazioni specifiche per il target (ACBGEN per i PSBLOAD, Import nel database di MAPDSECT e COPY Fxxx1MXR per i MAPSET, BIND per I LOAD DB2, etc).

In COLL opera sia a livello di target item negli skeleton LCSxC000 (compilazione e preparazione di comandi specifici necessari nel processo di MIGRATE), sia a livello di intero pacchetto, nello skeleton LCS0CTRA, contenendo al suo interno il codice necessario per:

- individuare se è finita la fase di build degli oggetti del pacchetto, cioè se il job in corso di esecuzione è l'ultimo dei job di build lanciati a valle della fase di MIGRATE e di ANALYZE del pacchetto nell'ambiente di destinazione;
- effettuare operazioni a livello di intero pacchetto sulla partizione PLT1;
- eventualmente preparare e schedulare un job che effettui sulla partizione PLT2 operazioni a livello di intero pacchetto (per es. il BIND PACKAGE e BIND PLAN sul sottosistema DB2 di collaudo).

In ESER pure opera sia a livello di target item negli skeleton LCSxE000 (copie e preparazione di comandi specifici necessari al processo di MIGRATE)), sia a livello di intero pacchetto, nello skeleton LCS0ETRA) contenendo al suo interno il codice necessario per:

- individuare se è finita la fase di build degli oggetti del pacchetto, cioè se il job in corso di esecuzione è l'ultimo dei job di build

lanciati a valle della fase di MIGRATE e di ANALYZE del pacchetto nell'ambiente di destinazione;

- effettuare operazioni a livello di intero pacchetto sulla partizione PLT1 (allineamento delle target library di consultazione e riferimento, allineamento nel database ASG/LCM degli ambienti alternati rispetto all'ambiente di provenienza della MIGRATE, BIND e ACBGEN sui sottosistemi DB2 e IMS di provenienza e alternati,etc.).
- preparare e schedulare un job che effettui sulla partizione di esercizio operazioni a livello di intero pacchetto (per es. la copia degli oggetti target dalle librerie di appoggio su dischi condivisi sulle librerie target di esercizio, il BIND PACKAGE e BIND PLAN sul sottosistema DB2 di esercizio, etc.).

4.3.5.2 I parametri delle build procedure

Gli skeleton richiamati dalle Build procedure prevedono che siano impostate una serie di variabili ISPF che possono essere specifiche di:

1. di singolo ambiente (al di sopra delle aree)
2. di ambiente e tipologia (al di sopra delle aree)
3. di area
4. di area e ambiente
5. di area ,ambiente e tipologia
6. di pacchetto
7. di item

Scegliere la modalità più idonea per l'impostazione di queste variabili significa rendere estremamente più manutenibile LCMSW, riducendo al massimo i punti di intervento quando si deve modificare la valorizzazione di una variabile.

A tale scopo il ASG/LCM fornisce tre modalità native per valorizzare variabili ISPF che vengono messe a disposizione delle build procedure al momento del 'file tailoring'; queste modalità, che ben soddisfano la necessità di valorizzare variabili del tipo 3., 4., 5., 6.,7 sono di seguito descritte.

Il **system parameter set** consente di definire e valorizzare un insieme di variabili ISPF a livello di system (area) che, nell'ambito di questa, possono essere ulteriormente differenziate a livello di configuration e type.

Il **work-area parameter set** consente di definire e valorizzare un insieme di variabili ISPF specifiche a livello di work-area, disgiunte rispetto a quelle definite a livello di system parameter set. Con il termine disgiunte intendiamo che, laddove il nome di tali variabili coincida nei due parameter

set, prevale il valore assegnatole a livello di system parameter set.

Utilizzando, come nel caso della nostra progettazione, work-area di package, omonime rispetto a questi, la tecnica risulta estremamente utile per valorizzare variabili specifiche di pacchetto, come il nome del pacchetto, la sua configurazione iniziale, l'obiettivo contrattuale cui si riferisce il pacchetto etc.

I **parametri a livello di build procedure** sono delle variabili ISPF di sistema che il ASG/LCM rende disponibili all'atto della generazione (file tailoring) di un job di build per uno specifico target item. Fra queste quelle di maggior interesse nel nostro caso sono:

&TYPE che rappresenta il target type oggetto della build procedure
 &ITEM che rappresenta il nome membro nell'ambito del TYPE che si sta generando con l'esecuzione corrente della build procedure

4.3.5.3 System parameter set

Nella nostra installazione, per ciascun system, i system parameter set saranno i seguenti:

<i>target config.</i>	<i>LOAD</i>	<i>PSBLOAD</i>	<i>MAPSET</i>
SVIL	SVILLOA	SVILPSB	SVILMAP
COLL	COLLLOA	COLLPSB	COLLMAP
ESER	ESERPAR	ESERPAR	ESERPAR
MAN	MANLOA	MANPSB	MANMAP

4.3.5.4 Work-area parameter set

All'atto della creazione di una nuova work-area omonima rispetto al pacchetto su cui si sta operando, viene definito il corrispondente parameter set e viene valorizzato automaticamente con i parametri indicati nella tabella che segue

Parameter set di work area (PSW)		
Nome parametro	Descrizione contenuto	Valorizzazione del parametro
&RGSPKID	Nome del pacchetto cui si riferisce la work area	&workarea (valorizzato in automatico dalla personalizzazione utente ASG/LCM con il nome della workarea al momento della creazione della workarea)

&RGSCNFI	Configurazione iniziale del pacchetto	SVIL o MAN (valorizzato in automatico dalla personalizzazione utente ASG/LCM con la configurazione iniziale del pacchetto al momento della creazione della workarea)
&TESTPKG	Richiesta o meno build con strumenti di test a livello di pacchetto	(preimpostato a N in automatico dalla personalizzazione utente ASG/LCM e modificabile da parte dell'utente)
&OBIETT	Obiettivo per aggiornare program directory	a carico dell'utente con stringa di max 2 caratteri
&DATAVV	data avviamento obiettivo per aggiornare program directory	a carico dell'utente con stringa di formato aammgg
&DATARIL	data rilascio obiettivo per aggiornare program directory	a carico dell'utente con stringa di formato aammgg

4.3.5.5 *Gli skeleton LCSxyPAR di valorizzazione dei parametri a livello di type e/o configuration*

Per la valorizzazione dei parametri di tipo 2. (comuni a tutte le aree ma differenti per ambiente e tipologia), si utilizzano invece degli skeleton LCMSW richiamati all'inizio degli skeleton LCSxy000 di cui alla sez. 4.3.5.1

)IM LCSx&AMB..PAR

mentre, per quanto riguarda i parametri di tipo 1 (comuni a tutte le aree e specifici di configurazione), si utilizzano invece degli skeleton LCMSW richiamati all'inizio degli skeleton LCS0CTRA e LCS0ETRA di cui alla sez. 4.3.5.1

)IM LCS0&AMB.PAR

4.3.6 **Host library**

Le librerie previste da ASG/LCM possono essere ricondotte a tre tipologie:

- *Export host library*
- *Build host library*
- *Target library*

Le prime sono le librerie dove vengono appoggiati gli oggetti del database di CCC per effettuarne l'editing; le seconde sono le librerie di 'scompress' degli oggetti del database, necessarie alla compilazione, mentre le terze sono le librerie che contengono oggetti target (MAPSET, LOAD, PSBLOAD e DBRM). Quest'ultime possono essere aggiornate sia dalle procedure di BUILD del prodotto sia dalle procedure di trasferimento.

Nel paragrafo 5.4 del 'Manuale d'uso' si può trovare una descrizione dettagliata delle caratteristiche delle diverse Host library.

Di seguito si riporta la modalità di definizione delle librerie all'interno del database.

4.3.6.1 Export host library

Le export host library si definiscono a livello di system con l'opzione CCC.4.S.1. Le host library, suddivise per ambiente (Svil, Coll, Man, Eser), sono quelle indicate nella tabella (Fig. 4.4.3.a) di questo manuale.

4.3.6.2 Target library

Sono le librerie indicate nella tabella (Fig. 4.4.3.c) di questo manuale. Non si definiscono nel database di CCCLCM ma sono referenziate sia nelle procedure di Build che in quelle di trasferimento.

4.3.6.3 Build library

Si definiscono con l'opzione CCC.3.1.4 del prodotto, suddividendole per ambiente (Svil, Coll, Man, Eser). Sono quelle indicate nella tabella (Fig. 4.4.3.b) di questo manuale.

4.4. SICUREZZA

Per una descrizione dettagliata di come è implementato il sistema di sicurezza per ASG/LCM si rimanda al capitolo 6.10 della specifica di analisi.

4.4.1 Utenze, gruppi di utenze e funzioni consentite per i vari gruppi

I gruppi RACF definiti per ASG/LCM sono suddivisi per area e, all'interno di ogni area, suddivisi tra capo progetto e programmatore. Oltre ai gruppi suddivisi per area c'è il gruppo CCCDBA che contiene le chiavi dei gestori del prodotto ASG/LCM.

Lo standard di nomenclatura dei gruppi suddivisi per area è il seguente:

- CCCCPxxx per i Capo progetto dell'area xxx
- CCCUSxxx per i programmatori dell'area xxx.

Le utenze da definire nei gruppi sono le chiavi TSO con le quali l'utente esegue il logon, coincidenti anche con quelle tramite le quali si accede al prodotto ASG/LCM.

I gruppi di utenti CCC hanno lo stesso standard di nomenclatura di quelli RACF.

In base all'appartenenza ad un determinato gruppo è possibile accedere all'una o all'altra funzione del prodotto.

In particolare gruppi CCCUSxxx devono poter:

- creare pacchetti con la funzione di ASSIGN
- trasferire un pacchetto da SVIL a COLL (previa autorizzazione da parte di CCCCPxxx) e da MAN a ESER.
- eseguire il Fall Back da COLL a SVIL.

I gruppi CCCCPCxxx devono poter:

- autorizzare il trasferimento di un pacchetto da SVIL verso COLL
- autorizzare il trasferimento di un pacchetto da COLL verso ESER
- creare pacchetti con la funzione di MANAGE.

Il gruppo CCCDBA deve poter:

- inibire e ripristinare i trasferimenti da MAN verso ESER per un'intera area
- trasferire un pacchetto di modifiche da MAN verso ESER
- effettuare il ripristino veloce delle funzionalità in esercizio a valle di un trasferimento verso ESER.

Il gruppo CCCDBA ha in generale la possibilità di svolgere qualunque funzione prevista dal prodotto.

4.4.2 Profili RACF per la protezione del database RGSCDC

I profili RACF per la protezione degli oggetti del database RGSCDC devono essere definiti utilizzando la classe @CCCCLAS. La predisposizione della classe è stata fatta seguendo le indicazioni riportate sul manuale ASG/Life Cycle Manager 'Installation and Administration Guide'.

Si riporta di seguito lo schema dei profili da definire, dove con xxx si indica il system identification, che, come più volte indicato, coincide con l'area applicativa

<i>Profili RACF</i>	<i>UACC</i>	<i>Gruppo Utente</i>	<i>Accesso consentito</i>
RSGCDC.**	NONE	CCCDBA	ALTER
RGSCDC.xxx.**	NONE	CCCUSxxx CCCCPxxx	READ READ
RSGCDC.xxx.SVIL.**	NONE	CCCUSxxx CCCCPxxx	UPDATE ALTER
RGSCDC.xxx.SVIL.BLDPROC	NONE	CCCUSxxx CCCCPxxx	READ READ
RSGCDC.xxx.MAN.**	NONE	CCCUSxxx CCCCPxxx	UPDATE ALTER
RSGCDC.xxx.MAN.DCLGEN.**	NONE	CCCUSxxx CCCCPxxx	READ READ
RGSCDC.xxx.MAN.BLDPROC	NONE	CCCUSxxx CCCCPxxx	READ READ

Fig 4.4.2 a

I job predisposti per la creazione dei profili si trovano sulla libreria 'H0T2.CCCLCM.RACF'

Lo schema indica che gli utenti appartenenti al gruppo CCCDBA possono effettuare qualsiasi operazione su qualsiasi oggetto della base dati RGSCDC.

Gli utenti del gruppo CCCUSxxx possono leggere gli oggetti della propria area in tutte le configuration (SVIL, COLL, ESER, MAN); possono aggiornare (operazioni di ADD, MODIFY, COPY) gli oggetti della propria area nelle configuration SVIL e MAN.

Le DCLGEN nella configurazione MAN le possono accedere solo in lettura; così come le BLDPROC nelle configurazioni SVIL e MAN.

Non possono eseguire cancellazioni di oggetti.

Gli utenti del gruppo CCCCCPxxx possono leggere gli oggetti della propria area in tutte le configuration (SVIL, COLL, ESER, MAN); possono aggiornare gli oggetti come gli utenti di CCCUSxxx e eseguire cancellazioni, ma solo nelle configurazioni SVIL e MAN. Possono accedere solo in lettura alle DCLGEN della configurazione MAN così come alle BLDPROC nelle configurazioni SVIL e MAN.

4.4.3 Profili RACF per la protezione delle host library

Le host library del prodotto, descritte nel paragrafo 4.3.6, sono sottoposte alla protezione RACF.

Di seguito viene fornito, per ogni tipologia, l'elenco delle host library suddiviso per le tre partizioni: PLT1, PLT2 e ESERCIZIO.

Tutte le definizioni a RACF sono state fatte mediante job presenti nelle librerie elencate di seguito:

H0T2.CCCLCM.RACF	partizione PLT1
H0T2.CCCLCM.RACF	partizione PLT2
H0TP.CCCLCM.RACF	partizione ESER

PARTIZIONE PLT1

EXPORT HOST LIBRARY

La tabella è suddivisa tra i quattro ambienti previsti da ASG/LCM (SVIL, MAN, COL, ESER) e riporta il nome del dataset, la sua descrizione e gli utenti che devono accedervi. L'ultima colonna indica se si tratta di una libreria nuova o già presente sulla partizione. Per tutte le librerie è previsto l'accesso in ALTER del gruppo CCCDBA.

NOME DATASET	TIPO	ACCESSI	NEW/OLD
Ambiente sviluppo			
H0T2.&AREA.DCLGEN	Libreria Dclgen suddivise per area	Utenti dell'area in UPDATE	NEW
H0T2.&AREA.SOURCE	Librerie sorgenti suddivise per area	Utenti dell'area in UPDATE	NEW
H0T2.&AREA.COPY	Librerie copy suddivise per area	Utenti dell'area in UPDATE	NEW
H0T2.&AREA.INCLUDE	Librerie include suddivise per area	Utenti dell'area in UPDATE	NEW
H0T2.&AREA.MAPDSECT	Librerie Dsect delle mappe suddivise per area	Utenti dell'area in UPDATE	NEW
H0T2.COMMON.DGIGRP	Libreria dei panel group di SDF2	Tutti gli utenti in UPDATE	NEW
H0T2.COMMON.DGIPNL	Libreria dei pannelli di SDF2	Tutti gli utenti in UPDATE	NEW
H0T2.&AREA.PSBSRC	Librerie PSB suddivise per area	Utenti dell'area in UPDATE	NEW
Ambiente correzione			
H0T3.&AREA.SOURCE	Librerie sorgenti suddivise per area	Utenti dell'area in UPDATE	NEW
H0T3.&AREA.COPY	Librerie copy suddivise per area	Utenti dell'area in UPDATE	NEW
H0T3.&AREA.INCLUDE	Librerie include suddivise per area	Utenti dell'area in UPDATE	NEW
H0T3.&AREA.MAPDSECT	Librerie Dsect delle mappe suddivise per area	Utenti dell'area in UPDATE	NEW
H0T3.COMMON.DGIGRP	Libreria dei panel group di SDF2	Tutti gli utenti in UPDATE	NEW
H0T3.COMMON.DGIPNL	Libreria dei pannelli di SDF2	Tutti gli utenti in UPDATE	NEW
H0T3.&AREA.PSBSRC	Librerie PSB suddivise per area	Utenti dell'area in UPDATE	NEW
Ambiente collaudo			
H0TL.COMMON.SOURCE	Librerie sorgenti	CCCDDBA in ALTER	NEW
H0TL.COMMON.COPY	Librerie copy, dclgen, include e dsect	CCCDDBA in ALTER	NEW
H0TL.COMMON.DGIGRP	Libreria dei panel group di SDF2	CCCDDBA in ALTER	NEW
H0TL.COMMON.DGIPNL	Libreria dei pannelli di SDF2	CCCDDBA in ALTER	NEW
H0TL.COMMON.PSBSRC	Librerie PSB suddivise per area	CCCDDBA in ALTER	NEW
Ambiente consultazione e riferimento			
H0T5.COMMON.SOURCE	Librerie sorgenti	CCCDDBA in ALTER	NEW
H0T5.COMMON.COPY	Librerie copy	CCCDDBA in ALTER	NEW
H0T5.COMMON.DCLGEN	Libreria DCLGEN	CCCDDBA in ALTER	NEW
H0T5.COMMON.INCLUDE	Librerie INCLUDE	CCCDDBA in ALTER	NEW
H0T5.COMMON.MAPDSECT	Librerie Dsect delle mappe	CCCDDBA in ALTER	NEW
H0T5.COMMON.DGIGRP	Libreria dei panel group di SDF2	CCCDDBA in ALTER	NEW
H0T5.COMMON.DGIPNL	Libreria dei pannelli di SDF2	CCCDDBA in ALTER	NEW
H0T5.COMMON.PSBSRC	Librerie PSB suddivise per area	CCCDDBA in ALTER	NEW

Fig 4.4.3.a

Per questi data set occorre definire i seguenti profili RACF

H0T2.&AREA*.* UACC=NONE - il gruppo corrispondente all'area in UPDATE e i gruppi SISTEMA e CCCDBA in ALTER.

H0T2.COMMON* UACC=UPDATE - CCCDBA e SISTEMA in ALTER

H0T3.&AREA*.* UACC=NONE - il gruppo corrispondente all'area in UPDATE e i gruppi SISTEMA e CCCDBA in ALTER.

H0T3.COMMON* UACC=UPDATE - CCCDBA e SISTEMA in ALTER

H0TL.COMMON* UACC=READ - CCCDBA e SISTEMA in ALTER

H0T5.COMMON* UACC=READ - CCCDBA e SISTEMA in ALTER

BUILD HOST LIBRARY

Di seguito vengono riportate le build host library suddivise per ambiente con nome dataset, la sua descrizione e l'accesso previsto per gli utenti. L'ultima colonna indica se si tratta di una libreria nuova o già presente sulla partizione.

NOME DATASET	TIPO	ACCESSI	NEW/OLD
Ambiente sviluppo			
H0T2.BUILD.SOURCE	Libreria dei sorgenti	Tutti gli utenti in UPDATE	NEW
H0T2.BUILD.COPY	Libreria delle copy, dclgen, include e mapdsect	Tutti gli utenti in UPDATE	NEW
H0T2.BUILD.DGIGRP	Libreria dei panel group SDF2	Tutti gli utenti in UPDATE	NEW
H0T2.BUILD.DGIPNL	Libreria dei pannelli SDF2	Tutti gli utenti in UPDATE	NEW
H0T2.BUILD.PSBSRC	Libreria dei PSB	Tutti gli utenti in UPDATE	NEW
Ambiente manutenzione			
H0T3.BUILD.SOURCE	Libreria dei sorgenti	Tutti gli utenti in UPDATE	NEW
H0T3.BUILD.COPY	Libreria delle copy, dclgen, include e mapdsect	Tutti gli utenti in UPDATE	NEW
H0T3.BUILD.DGIGRP	Libreria dei panel group SDF2	Tutti gli utenti in UPDATE	NEW
H0T3.BUILD.DGIPNL	Libreria dei pannelli SDF2	Tutti gli utenti in UPDATE	NEW
H0T3.BUILD.PSBSRC	Libreria dei PSB	Tutti gli utenti in UPDATE	NEW
Ambiente collaudo			
H0TL.BUILD.SOURCE	Libreria dei sorgenti	Tutti gli utenti in UPDATE	NEW
H0TL.BUILD.COPY	Libreria delle copy, dclgen, include e mapdsect	Tutti gli utenti in UPDATE	NEW
H0TL.BUILD.PSBSRC	Libreria dei PSB	Tutti gli utenti in UPDATE	NEW
Ambiente esercizio			
H0T5.BUILD.SOURCE	Libreria dei sorgenti	Tutti gli utenti in UPDATE	NEW
H0T5.BUILD.COPY	Libreria delle copy, dclgen, include e mapdsect	Tutti gli utenti in UPDATE	NEW
H0T5.BUILD.PSBSRC	Libreria dei PSB	Tutti gli utenti in UPDATE	NEW
H0T5.BUILD.DGIPNL	Libreria dei pannelli SDF2	Tutti gli utenti in UPDATE	NEW
H0T5.BUILD.DGIGRP	Libreria dei Panel Group	Tutti gli utenti in UPDATE	NEW

Fig 4.4.3.b

Un batch le cancella e le ricrea giornalmente (cfr. sez. 5.1.3).

Una volta preparato il job occorre verificare che l'utenza associata abbia autorizzazione in ALTER a tutti i profili riportati di seguito.

I profili RACF da definire devono avere le seguenti caratteristiche.

H0T2.BUILD* UACC=UPDATE CCCDBA e SISTEMA
IN ALTER

H0T3.BUILD* UACC=UPDATE CCCDBA e SISTEMA
in ALTER

H0TL.BUILD* UACC=UPDATE CCCDBA e SISTEMA
in ALTER

H0T5.BUILD* UACC=UPDATE CCCDBA e SISTEMA
in ALTER

TARGET HOST LIBRARY

Si illustrano di seguito le target host library suddivise per ambiente, in tabella viene riportato il nome del dataset, la sua descrizione e il tipo di accesso per i vari utenti. L'ultima colonna indica se si tratta di una libreria nuova o già presente sulla partizione.

NOME DATASET	TIPO	ACCESSI	NEW/OLD
Ambiente sviluppo			
H0T2.&AREA.DBRM	Libreria dei DBRM	utenti dell'area in UPDATE	NEW
H0T2.MAPPE.LOAD	Libreria dei load delle mappe	Tutti gli utenti in UPDATE	NEW
WRS.PGM.LOAD1 WRS.PGM.LOAD2 WRS.PGM.LOAD3 WRS.PGM.LOAD4 WRS.PGM.LOAD5	Libreria dei load dei programmi	Tutti gli utenti in UPDATE	OLD
WRS.PSBLIB.LOAD	Libreria load dei PSB	Tutti gli utenti in UPDATE	OLD
Ambiente manutenzione			
H0T3.&AREA.DBRM	Libreria dei DBRM	utenti dell'area in UPDATE	NEW
H0T3.MAPPE.LOAD	Libreria dei load delle mappe	Tutti gli utenti in UPDATE	NEW
WRG.PGM.LOAD1	Libreria dei load dei programmi	Tutti gli utenti in UPDATE	OLD
WRG.PSBLIB.LOAD	Libreria load dei PSB	Tutti gli utenti in UPDATE	OLD
Ambiente collaudo			
H0TL.COMMON.DBRM	Libreria dei DBRM	Accesso in ALTER a CCCDBA	NEW
H0TL.RGSLIB	Libreria dei load dei programmi	Accesso in ALTER a CCCDBA	NEW
H0TL.MAPPE.LOAD	Libreria dei load delle mappe	Accesso in ALTER a CCCDBA	NEW
H0TL.PSBLIB.LOAD	Libreria load dei PSB	Accesso in ALTER a CCCDBA	NEW
Ambiente consultazione e riferimento			
H0T5.COMMON.DBRM	Libreria dei DBRM	Accesso in ALTER a CCCDBA	NEW
H0T5.RGSLIB H0T5.RLT.LOAD	Libreria dei load dei programmi (batch e TP)	Accesso in ALTER a CCCDBA	OLD
H0T5.MAPPE.LOAD	Libreria dei load delle mappe	Accesso in ALTER a CCCDBA	OLD
H0T5.PSBLIB.LOAD	Libreria load dei PSB	Accesso in ALTER a CCCDBA	OLD

Fig 4.4.3.c

Le compilazioni da parte degli utenti si effettuano solamente negli ambienti di sviluppo e correzione, mentre nell'ambiente di collaudo è il job dei trasferimenti che attiva la compilazione del sorgente dopo averlo trasferito.

I profili RACF necessari sono i seguenti:

H0T2.&AREA*.*	cfr. quanto indicato per le 'export host library'
WRS.PGM.LOAD*	UACC=UPDATE - CCCDBA, SISTEMA e BATCH in ALTER
H0T2.MAPPE.LOAD*	UACC=UPDATE - CCCDBA e SISTEMA in ALTER
WRS.PSBLIB.LOAD*	UACC=UPDATE - CCCDBA, SISTEMA e BATCH in ALTER
H0T3.&AREA*.*	cfr. quanto indicato per le 'export host library'.
WRG.PGM.LOAD*	UACC=UPDATE - CCCDBA, SISTEMA e BATCH in ALTER
H0T3.MAPPE.LOAD*	UACC=UPDATE - CCDBA e SISTEMA in ALTER
H0TL.COMMON*	cfr. quanto indicato per le 'export host library'
H0TL.MAPPE.LOAD*	UACC=READ - CCCDBA e SISTEMA in ALTER
H0TL.PSBLIB.LOAD*	UACC=READ - CCCDBA e SISTEMA in ALTER
H0TL.RGSLIB*	UACC=READ - CCCDBA e SISTEMA in ALTER
H0T5.COMMON*	cfr. quanto indicato per le 'export host library'
H0T5.MAPPE.LOAD*	UACC=READ - CCCDBA, SISTEMA, BATCH e DATABASE in ALTER
H0T5.PSBLIB.LOAD*	UACC=READ - CCCDBA, SISTEMA, BATCH e DATABASE in ALTER
H0T5.RGSLIB*	UACC=READ - CCCDBA, SISTEMA, BATCH e DATABASE in ALTER
H0T5.RLT.LOAD*	UACC=READ - CCCDBA, SISTEMA, BATCH e DATABASE in ALTER

Attenzione: la discordanza tra quanto specificato nella tabella e i profili da definire è dovuta al fatto che gran parte dei dataset target sono già definiti sulla partizione PLT1 e per essi è già definito un profilo RACF. Per non andare in contrasto con il profilo già definito (nella maggior parte dei casi più generico), le autorizzazioni risultano

un 'merge' che renda compatibile la protezione del dataset sia nel nuovo che nel vecchio contesto.

LIBRERIE IN FINESTRA PER I TRASFERIMENTI

Le procedure dei trasferimenti utilizzano librerie allocate su dischi condivisi tra più partizioni per la movimentazione degli oggetti load. Di seguito si riporta il profilo RACF necessario.

H0T4.*.COMMAND* UACC=READ - CCCDBA e SISTEMA
in ALTER

H0T4.TRASF* UACC=UPDATE - CCCDBA e
SISTEMA in ALTER

PARTIZIONE PLT2

In questa partizione occorre definire il gruppo RACF CCCDBA dove vanno inserite le chiavi che devono avere accesso in ALTER a tutti i dataset di questa partizione. In particolare il gruppo deve contenere l'utenza dei trasferimenti STCTRAS.

EXPORT HOST LIBRARY

Non esistono in quanto l'editing per gli oggetti del DB avviene solo nel sistema PLT1.

BUILD HOST LIBRARY

Non esistono, in quanto le compilazioni che prevedono lo scompres degli oggetti del DB si effettuano solo nel sistema PLT1

TARGET HOST LIBRARY

Sono le librerie che contengono oggetti target (MAPSET, LOAD, PSBLOAD e DBRM).

Librerie condivise con PLT1		Accessi	OLD/NEW
H0TL.COMMON.DBRM	Libreria dei DBRM	Read	NEW
H0TL.RGSLIB	Libreria dei load dei programmi	Read	NEW
H0TL.MAPPE.LOAD	Libreria dei load delle mappe	Read	NEW
H0TL.PSBLIB.LOAD	Libreria load dei PSB	Read	NEW
H0TL.ACBLIB.LOAD	Libreria load degli ACB	CCCDBA in update	
H0T5.COMMON.DBRM	Libreria dei DBRM	Read	NEW
H0T5.RGSLIB	Libreria dei load dei programmi (batch e TP)	Read	OLD
H0T5.RLT.LOAD			
H0T5.MAPPE.LOAD	Libreria dei load delle mappe	Read	OLD
H0T5.PSBLIB.LOAD	Libreria load dei PSB	Read	OLD
H0T5.ACBLIB.LOAD	Libreria load degli ACB	CCCDBA in update	

Fig 4.4.3.d

Dato che tutte le librerie sono shared tra le partizioni PLT1 e PLT2 occorre verificare la loro protezione RACF anche negli altri sistemi. Non vi è bisogno di dare autorizzazioni in ALTER poiché le movimentazioni su questi dataset vengono fatte sul sistema PLT1.

I profili RACF necessari sono i seguenti

H0TL.COMMON*	UACC=READ
H0TL.RGSLIB *	UACC=READ
H0TL.ACBLIB*	UACC=READ - CCCDBA in UPDATE
H0TL.MAPPE.LOAD*	UACC=READ
H0TL.PSBLIB.LOAD*	UACC=READ
H0T5.COMMON*	UACC=READ
H0T5.RGSLIB*	UACC=READ
H0T5.ACBLIB*	UACC=READ
H0T5.MAPPE.LOAD*	UACC=READ
H0T5.PSBLIB.LOAD*	UACC=READ
H0T5.RLT.LOAD*	UACC=READ
H0T4.*.COMMAND*	UACC=READ
H0T4.TRASF*	UACC=READ - CCCDBA in UPDATE

PARTIZIONE ESERCIZIO

In questa partizione occorre definire il gruppo RACF CCCDBA dove vanno inserite le chiavi che devono avere accesso in ALTER a tutti i dataset di questa partizione. In particolare il gruppo deve contenere l'utenza dei trasferimenti STCTRAS.

EXPORT HOST LIBRARY

Non esistono in quanto l'editing per gli oggetti del DB avviene solo nel sistema PLT1

BUILD HOST LIBRARY

Non esistono, in quanto le compilazioni che prevedono lo scompres degli oggetti del DB si effettuano solo nel sistema PLT1

TARGET HOST LIBRARY

Sono le librerie che contengono oggetti target (MAPSET, LOAD, PSBLOAD e DBRM).

Librerie ESERCIZIO		
OCC.COMMON.DBRM	Libreria dei DBRM	CCCDDBA in UPDATE
RGSLIB	Libreria dei load dei programmi	CCCDDBA in UPDATE
OCC.RLT.LOAD	(batch e TP)	
OCC.MAPPE.LOAD	Libreria dei load delle mappe	CCCDDBA in UPDATE
OCC.PSBLIB.LOAD	Libreria load dei PSB	CCCDDBA in UPDATE
OCC.ACBLIB.LOAD	Libreria Load degli ACB	CCCDDBA in UPDATE

Fig 4.4.3.e

OCC.COMMON.DBRM* UACC=READ - SISTEMA, CICS e DATABASE in ALTER /CCCDDBA, in UPDATE

RGSLIB UACC=READ - CCCDBA in UPDATE, SISTEMA, CICS e DATABASE in ALTER (importante che sia attiva la protezione RACF per un singolo qualificatore)

OCC.RLT.LOAD* UACC=READ - CCCDBA in UPDATE, SISTEMA, CICS e DATABASE in ALTER

OCC.MAPPE.LOAD* UACC=READ - CCCDBA in UPDATE, SISTEMA, CICS e DATABASE in ALTER

OCC.PSBLIB.LOAD* UACC=READ - CCCDBA in UPDATE, SISTEMA, CICS e DATABASE in ALTER

OCC.ACBLIB.LOAD* UACC=READ - CCCDBA in UPDATE, SISTEMA, CICS e DATABASE in ALTER

Per le librerie dei trasferimenti in finestra, analogamente al sistema di collaudo i profili da definire sono:

H0T4.*.COMMAND* UACC=READ

H0T4.TRASF* UACC=READ - CCCDBA in UPDATE

4.4.4 L'utenza STCTRAS per la gestione dei trasferimenti

I trasferimenti tra le varie partizioni avvengono utilizzando l'utenza STCTRAS definita su tutte le partizioni (PLT1, PLT2, ESER).

L'utenza oltre a far parte del gruppo CCCDBA deve avere, sulla partizione PLT1, il corrispondente profilo di dataset, STCTRAS.*, tale da consentire accessi in ALTER a CCCDBA.

L'utenza STCTRAS dovrà essere definita SURROGAT in tutte le partizioni nelle quali debbano essere sottomessi job di trasferimento oggetti.

Nel database di CCC l'utenza STCTRAS va definita nella classe DBA con password coincidente con l'utenza.

L'utenza STCTRAS costituirà la chiave tramite la quale utilizzare la funzione RSHELL del TCP/IP per l'allineamento della base dati di consultazione e riferimento nelle partizioni PLT2 e ESERCIZIO.

La protezione RACF da implementare per le tre partizioni è la seguente:

Su PLT1

- UACC (NONE) su STCTRAS.RHOSTS.DATA
- ACCESS (ALTER) al gruppo CCCDBA su STCTRAS.RHOSTS.DATA.

Su PLT2 ed ESER:

- creazione dell'utente STCRXPRO da assegnare come OWNER al Remote Server RXPROC,
- ACCESS (READ) allo user STCRXPRO su STCTRAS.RHOSTS.DATA e UACC(NONE) (questa è la protezione minima affinché il processo possa funzionare),
- creazione del profile STCTRAS.SUBMIT nella classe SURROGAT, con STCRXPRO e CCCDBA avente ACCESS(READ),

4.5 MODIFICA AL PARAMETRO MAXIMUM NUMBER OF CCC USER

Il parametro di installazione 'Maximum number of CCC User', indica quanti utenti possono accedere contemporaneamente al prodotto.

Il valore attuale è 99; nel caso fosse necessario modificarlo occorre intervenire sui 2 membri:

CCCLCMS.V30D.CCCPARM(CCCPARM)

CCCLCMS.V30D.CCCPARM(IPCPARM)

Il valore 99, presente all'interno di essi, dovrà essere sostituito con quello scelto, compreso tra 4 e 9999. Nelle figure sotto, è riportato il contenuto dei membri dove sono evidenziati i valori da sostituire.

* ----- *

```

* --- IPC INITIALIZATION PARAMETERS --- *
* ----- *
*
* IPC SVC = 235,
* IPC KEY = 3,
* IPC CSA = IPCCSA,
* MAXIMUM IPC NODES = 99
*
IEFSSN IPC3,IPCINIT,'235,3,IPCCSA,99,X"00020000"'

```

Fig 4.5.a IPCPARM

```

235 99
PLATINUM CCC

```

Fig 4.5.b CCCPARM

Alla partenza successiva, il prodotto opererà con il nuovo parametro.

5. LA GESTIONE DEGLI AMBIENTI A SUPPORTO DEL C.V.S. CON IL ASG/LCM

In questa sezione vengono descritte le operazioni ordinarie e straordinarie a carico del personale tecnico per la gestione dell'ambiente a supporto del C.V.S. con il ASG/LCM

5.1. PROCEDURE DI GESTIONE ORDINARIA DI LIBRERIE E ARCHIVI DI AMBIENTE

5.1.1 Il back-up del database RGSCDC e degli archivi del prodotto

Il back-up del database del prodotto e dei dataset di sistema è gestito tramite il job batch HSAVECCC che viene attivato nel batch serale. I file che vengono salvati sono i seguenti:

H0VS.RGSCDC.DATABASE,
H0TS.RGSCDC.INITIAL,
H0TS.RGSCDC.LOG,
H0TS.RGSCDC.Dyyymmgg.Thhmmss,
H0TS.RGSCDC.LOOKAHED

5.1.2 Pulizia del file di log

Il job batch HSAVECCC, oltre ad eseguire il back up dei file sopraindicati, esegue, dopo averli salvati, la cancellazione di tutti i data set di log creati a partire dall'ultimo salvataggio.

I dataset di log hanno il nome che segue lo standard indicato sotto dove yyymmgg e hhmmss indicano rispettivamente l'anno, il mese, il giorno e l'ora minuti e secondi, in cui il log è stato creato:

H0TS.RGSCDC.Dyyymmgg.Thhmmss

Il file di log viene creato al momento della partenza del prodotto o quando risulta saturo quello vecchio.

5.1.3 I back-up delle export e delle target host library

Le export e le target host library di cui alla sez.4.3.6.1 e 4.3.6.2 saranno sottoposte alle modalità standard di salvataggio del DFSMS per quelle i cui prefissi sono attualmente già stati migrati su dischi 3390, sotto la gestione del suddetto prodotto IBM.

Quelle invece i cui prefissi sono ancora associati a volumi di tipo 3380 al di fuori del controllo del DFSMS, saranno sottoposte ai salvataggi con il prodotto SAMS/DISK, secondo gli standard del centro.

5.1.4 Procedure di pulizia delle build host library

Sul sistema di sviluppo PLT1, al completamento del batch notturno dei trasferimenti, sarà attivato quotidianamente il batch standard HALLBLD che effettua lo svuotamento delle build host library di cui alla sez.4.3.6.3, tramite cancellazione e riallocazione delle medesime;

5.1.5 I back-up delle librerie del ASG/LCM e dell'LCMSW 1.0

Queste librerie sono attualmente fuori del controllo del DFSMS. Per esse, finché non vengono spostate su volumi gestiti da DFSMS, è previsto il salvataggio con SAMS/DISK.

5.1.6 I back-up della base dati DB2

Il database DDLCM00 contenente le tabelle DB2 della base dati del prodotto (vedi sez. 4.2.3, 5.4.3 e 5.4.4) saranno incluse nella sicurezza DB2 standard su PLT1 attualmente prevista solo per le tabelle di catalogo e per i prodotti, con le medesime modalità previste per questi ultimi.

5.2. PROCEDURE DI GESTIONE STRAORDINARIA DI LIBRERIE E ARCHIVI DI AMBIENTE

Il recovery del Data base RGSCDC, in caso di danneggiamento, va effettuato seguendo le modalità riportate nella documentazione [2] al paragrafo 3 - Administrating a CCC data base.

Particolare attenzione va posta al data set Journal Index, che costituisce l'indice per poter referenziare correttamente i journal da applicare al data base RGSCDC da restaurare.

Poiché alla corretta chiusura del Data Base il journal index viene cancellato, si riporta di seguito la modalità di creazione nel caso non fosse presente sul sistema.

- Verificare che sul sistema ci siano solo i journal da applicare al DB restorato.
- Eseguire a TSO il seguente comando:
EXEC 'CCCLCMS.V30D.CCCCLIST(CCCINDEX)' 'H0TS RGSCDC'

Per quanto riguarda le procedure di gestione straordinaria e ordinaria delle librerie e dei dataset di ambiente, si rimanda alle modalità standard del centro (cfr. sez. 5.1).

5.3. PROCEDURA DI PARTENZA E DI CHIUSURA DEL ASG/LCM

L'attivazione e chiusura del ASG/LCM sarà implementata conformemente alle indicazioni contenute nella documentazione nativa del prodotto (cfr.[2]) e alle modalità standard del centro di schedulazione dei sistemi.

La partenza avviene mediante l'utilizzo dell'OPC;

i comandi sono i seguenti:

S CCCIPC

S RGSCDC

F RGSCDC,updates

La chiusura avviene mediante l'utilizzo dell'OPC;

i comandi sono i seguenti:

F RGSCDC, noupdate

F RGSCDC, close

P CCCIPC

La started procedure RGSCDC contiene lo step che effettua la copia del log in linea dei trasferimenti sul log di back-up (cfr 5.4.3)

5.4. LA SCHEDULAZIONE BATCH DEI TRASFERIMENTI

Quando un utente completa la sua richiesta di Migrare di un pacchetto, attraverso l'opzione 4.5 del ASG/LCM, viene creato il job capostipite nel membro omonimo rispetto al pacchetto della libreria H0T4.TRASF.JOB (cfr. sez. 5.4.1.1) e viene inserito un record nell'archivio di LOG (cfr. sez.5.4.3).

Nel seguito della presente sezione sono contenute tutte le informazioni necessarie a comprendere e a gestire i nuovi trasferimenti verso collaudo ed esercizio.

Le prime sei sezioni di terzo livello sono descrittive, le successive sono operative.

5.4.1 Le librerie di appoggio dei trasferimenti

I trasferimenti dei pacchetti si appoggiano su una serie di librerie in ambiente condiviso che sono brevemente descritte nel seguito

5.4.1.1 La libreria di appoggio dei job capostipite

Questa libreria è H0T4.TRASF.JOB ed è la libreria sulla quale viene generato il job capostipite per la migrate di un pacchetto con un nome membro omonimo rispetto al pacchetto.

Tale libreria è in input al job HTRASCCC e alla procedura HTRASURG che si occupano rispettivamente della schedulazione ordinaria (cfr. sez. 5.4.8) e straordinaria (cfr,sez. 5.4.9) dei trasferimenti.

5.4.1.2. La libreria di appoggio dei job di fase comune su PLT1

La libreria H0T4.TRASF.PLT1 è la libreria sulla quale è generato il job che implementa la fase comune del trasferimento su PLT1 a valle della fase di build sempre con un nome membro omonimo rispetto al pacchetto.

Tale libreria è in input alla procedura HTRASPL1 per il restart dei trasferimenti interrotti in stato 4 e 5.

5.4.1.3. Le librerie di appoggio dei job di fase comune sulla partizione di arrivo

Le librerie H0T4.TRASF.PLT2 (per i trasferimenti provenienti da SVIL) e H0T4.TRASF.ESER (per i trasferimenti provenienti da COLL o MAN) sono le librerie sulle quali è generato il job che implementa la fase comune del trasferimento sulla partizione di arrivo sempre con un nome membro omonimo rispetto al pacchetto.

Tali librerie sono in input rispettivamente alle procedure HTRASPL2 e HTRASESE, presenti rispettivamente nell'ambiente di Collaudo e Esercizio, il restart dei trasferimenti interrotti in stato 6 e 7.

5.4.1.4. La libreria di appoggio dei comandi necessari alla Migrate del pacchetto

Ogni job di build sottomesso dal job capostipite genera comandi e dati che sono presi in input dalle fasi comuni su PLT1 e sulla partizione di arrivo al completamento della fase di build.

Tali comandi sono accodati in membri di una libreria partitioned H0T4.packid..COMMAND allocata dal job capostipite in ambiente condiviso e che non può essere cancellata fino al completamento corretto del trasferimento del pacchetto.

La cancellazione di questa libreria, per tutti i pacchetti il cui trasferimento si sia correttamente completato è effettuata dal job HTRASCCC di schedulazione ordinaria dei trasferimenti

5.4.1.5. La libreria di appoggio dei jcl di utilità per la gestione dei trasferimenti

La libreria H0T4.TRASF.CNTL è una libreria che contiene una serie di jcl necessari alla gestione ordinaria dei trasferimenti, soprattutto nel caso di trasferimenti che si concludono in maniera anomala e di richieste di cancellazione di pacchetti.

Il membro \$\$DOC in testa alla libreria contiene la descrizione del contenuto della libreria stessa.

5.4.2 La struttura dei job che implementano il trasferimento di un pacchetto

La migrate di un pacchetto è implementata da un numero $n+3$ di job, omonimi rispetto al pacchetto, che quindi si serializzano automaticamente in virtù della loro omonimia dove n è il numero di job di build attivati dal trasferimento.

Di questi, $n+2$ job sono eseguiti sulla partizione PLT1 e un job, l'ultimo, è eseguito sulla partizione di destinazione del trasferimento (PLT2 per i trasferimenti da SVIL a COLL e esercizio per i trasferimenti da COLL a ESER e da MAN a ESER).

In Fig. 5.4.2.a è schematizzata la struttura degli n job che implementano la Migrate di un pacchetto

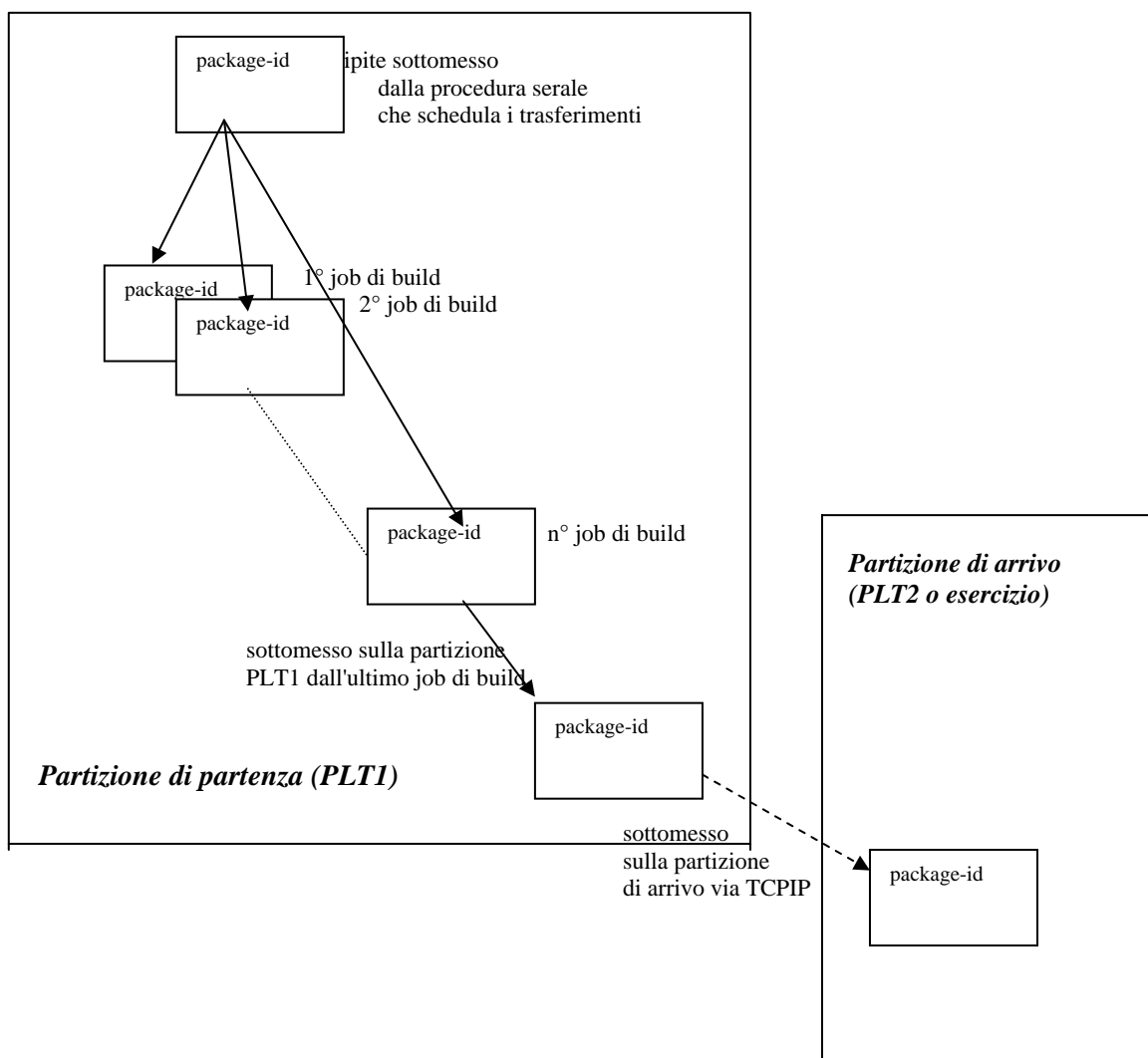


Fig.5.4.2.A
Struttura dei job che implementano la Migrate di un pacchetto

5.4.2.1 *Il job capostipite*

Il primo job sottomesso per la migrate di un pacchetto (job capostipite) effettua:

1. impostazione dello stato del pacchetto a 3 sul log dei trasferimenti;
2. operazioni preliminari, fra le quali l'allocazione della libreria `H0T4.&packid..COMMAND` su volume condiviso e l'inizializzazione di tutti i membri che su questa libreria verranno aggiornati e letti nel corso del trasferimento del pacchetto;
3. la generazione ⁶ di uno skeleton, che servirà, nel corso del job di cui alla sez.5.4.2.4 per i trasferimenti verso ESER, ad effettuare gli allineamenti delle configurazioni ASG/LCM alternate.
4. migrate del pacchetto con la promote sulle export host library degli ambienti di destinazione (funzione nativa del ASG/LCM);
5. analyze dei configuration item che costituiscono il pacchetto e dei configuration item indirettamente coinvolti dalle modifiche agli oggetti del pacchetto; questo passo, se l'analyze si conclude con esito corretto, provoca la sottomissione di tanti job di build (n) quanti sono i configuration item direttamente o indirettamente coinvolti dalle modifiche;
6. valorizzazione nel log dei trasferimenti del contatore dei job di build sottomessi dallo step di analyze ⁷ (campo num-job-subm di Fig.5.4.3.a);
7. generazione di un membro omonimo rispetto al pacchetto, contenente gli oggetti del pacchetto stesso, sul PO K0T2.CCC.BIG di interfaccia con il prodotto BIG

5.4.2.2 *I job di Build nel trasferimento verso COLL*

Ciascuno degli n job di build sottomessi dal job capostipite, nel trasferimento verso COLL effettua:

- la compilazione e il link-edit (con opzione NCAL se si tratta di un LOAD) dell'item specifico;
- la preparazione degli statement per il link-edit dell'oggetto accodati sul membro LKEDCMD del PO H0T4.&packid.COMMAND.;

⁶ Su una libreria temporanea di skeleton allocata e usata anche dal ASG/LCM nella fase di generazione dei job di build.

⁷ Questa informazione è ricavata elaborando l'archivio SYSTSPRT sul quale è scritto l'output dell'analyze (la personalizzazione RGS associa a SYSTSPRT un temporaneo elaborato negli step successivi).

- la preparazione dei comandi di ACBGEN, BIND PACKAGE e BIND PLAN in funzione del type dell'item oggetto della Build;
- l'aggiornamento del log degli eventi (cfr. sez.5.4.4) sia che il job si sia completato correttamente, sia che il job non si sia completato correttamente (abend);
- l'incremento del contatore dei job di build eseguiti sul log dei trasferimenti se le operazioni descritte ai punti precedenti, sono state implementate correttamente e il controllo se il job in esecuzione è l'ultimo dei job di build sottomessi. In questo caso il job effettua:
 1. la creazione di un job omonimo rispetto al pacchetto che viene copiato sulla libreria di sistema H0T4.TRASF.PLT1 e che, una volta sottomesso, implementerà la fase del trasferimento, comune a tutto il pacchetto, sulla partizione di partenza (ossia su PLT1);
 2. la creazione di un job omonimo rispetto al pacchetto che viene copiato sulla libreria di sistema H0T4.TRASF.PLT2 che, una volta sottomesso via TCPIP sulla partizione di arrivo, implementerà la fase conclusiva del trasferimento, comune a tutto il pacchetto;
 3. l'aggiornamento dello stato del trasferimento al valore 4 (finita fase di build) sul log dei trasferimenti;
 4. il submit sulla partizione PLT1 del job creato al punto 1.

5.4.2.3 *I job di Build nel trasferimento verso ESER*

Ciascuno degli n job di build sottomessi dal job capostipite, effettua nelle MIGRATE verso ESER:

- la copia degli oggetti target sulle librerie in ambiente condiviso e la preparazione di comandi di copia selettiva dei membri;
- la preparazione degli statement di pulizia delle librerie degli ambienti di provenienza dagli oggetti target del pacchetto;
- la preparazione degli statement di pulizia delle librerie in ambiente condiviso dagli oggetti target del pacchetto;
- la preparazione dei comandi di ACBGEN, BIND PACKAGE e BIND PLAN in funzione del type dell'item oggetto della Build;
- la preparazione dei comandi per l'aggiornamento della program directory;

- la preparazione dei comandi di INSERT SQL nella base dati di BOSS per far elaborare dal prodotto gli oggetti trasferiti;
- l'aggiornamento del log degli eventi (cfr. sez.5.4.4) sia che il job si sia completato correttamente, sia che il job non si sia completato correttamente (abend);
- l'incremento del contatore dei job di build eseguiti sul log dei trasferimenti se le operazioni descritte ai punti precedenti, sono state implementate correttamente e il controllo se il job in esecuzione è l'ultimo dei job di build sottomessi. In questo caso il job effettua:
 1. la creazione di un job omonimo rispetto al pacchetto che viene copiato sulla libreria di sistema HOT4.TRASF.PLT1 e che, una volta sottomesso, implementerà la fase del trasferimento, comune a tutto il pacchetto, su PLT1 (cfr. sez.5.4.2.4);
 2. la creazione di un job omonimo rispetto al pacchetto che viene copiato sulla libreria di sistema HOT4.TRASF.ESER che, una volta sottomesso via TCPIP sulla partizione di arrivo, implementerà la fase conclusiva del trasferimento, comune a tutto il pacchetto;
 3. l'aggiornamento dello stato del trasferimento al valore 4 (finita fase di build) sul log dei trasferimenti;
 4. il submit sulla partizione PLT1 del job creato al punto 1.

5.4.2.4 *Il job che implementa la fase comune sulla partizione PLT1 a valle della fase di Build*

Il job sottomesso al precedente punto 4. (job n+2-simo) svolge le seguenti funzioni:

- a. imposta a 5 lo stato del trasferimento sull'archivio di log dei trasferimenti;
- b. esegue, se il trasferimento è verso COLL, un passo di link-edit complessivo di tutti i LOAD ricompilati nei job di build per garantire che ciascuno contenga l'ultima versione degli eventuali LOAD richiamati staticamente;
- c. se il trasferimento è verso ESER, allinea l'ambiente di Consultazione e riferimento target (eseguibili e DBRM)⁸, aggiorna la Program Directory, allinea le configurazioni dei migration path alternati con gli oggetti modificati dal pacchetto, effettua BIND e/o ACBGEN di allineamento sui sottosistemi di sviluppo e manutenzione.

⁸ L'ambiente di Consultazione e Riferimento relativo ai simbolici è allineato direttamente dal ASG/LCM durante la MIGRATE (vedi job alla sez.5.4.2.1) attraverso la funzione di PROMOTE.

E' per questo motivo che prima di effettuare il trasferimento da COLL a ESER, occorre verificare che eventuali modifiche apportate alla base dati di sviluppo siano riportate nell'ambiente alternato (manutenzione) e nell'ambiente di ESERCIZIO dove verrà effettuato il BIND PLAN di allineamento.

- d. fino a quando è presente sul sistema l'ambiente SVEUR, all'interno del job ci saranno gli step necessari all'allineamento dell'ambiente di consultazione riferimento SVEUR a partire dall'ambiente di consultazione riferimento ASG/LCM. Questi step dovranno essere tolti al momento dello smantellamento dell'ambiente SVEUR.
- e. Fa partire, tramite TCPIP (cfr. sez.5.4.6), il job n+3-simo presente sulla libreria H0T4.TRASF.PLT2 o H0T4.TRASF.ESER sulla corretta partizione di arrivo
- f. Imposta a 6 lo stato del trasferimento sull'archivio di log dei trasferimenti.

5.4.2.5 Il job che implementa la fase comune sulla partizione di arrivo a valle della fase di Build

L'ultimo job sottomesso al precedente punto d. (job n+3-simo) sulla partizione di arrivo svolge le seguenti funzioni:

- a. imposta a 7 lo stato del trasferimento sull'archivio di log dei trasferimenti;
- b. esegue, se il trasferimento è verso ESER, la copia dei target item coinvolti nel pacchetto, dalle librerie in ambiente condiviso, sulle librerie in linea sul sistema di esercizio;
- c. esegue se necessario il BIND PACKAGE e BIND PLAN sul sottosistema DB2 della partizione di arrivo;
- d. esegue se necessario l'ACBGEN sul sottosistema IMS della partizione di arrivo;
- e. esegue l'inserimento nella base dati di BOSS (solo ESER) delle occorrenze necessarie a far elaborare i programmi trasferiti dal batch serale del prodotto;
- f. Imposta a 8 lo stato del trasferimento sull'archivio di log dei trasferimenti.

5.4.3 L'archivio di log dei trasferimenti e lo storico

Come si può evincere dalla descrizione della struttura del trasferimento di un pacchetto descritta alla sez. 5.4.2, l'archivio di log dei trasferimenti è il perno del meccanismo e consente di tenere sotto controllo lo stato di avanzamento della Migrate di un pacchetto e di individuarne le modalità di restart, nel caso in cui non termini correttamente.

Il log dei trasferimenti si articola logicamente in tre strutture:

1. *l'archivio in linea*: è un archivio sequenziale che deve essere acceduto esclusivamente via software, per garantire la corretta sequenzializzazione degli aggiornamenti che sono effettuati da partizioni differenti;
2. *l'archivio di consultazione giornaliera*: è anch'esso un archivio sequenziale H0T4.TRASF.LCALOGBK, giornalmente rinfrescato a partire dall'archivio in linea di cui al punto 1.; gli utenti possono consultarlo in browse per verificare l'esito della migrate del loro pacchetto;
3. *lo storico dei trasferimenti*: è una tabella DB2 D01.TDLCCLGT sulla quale vengono accodati i trasferimenti dei pacchetti che si sono correttamente conclusi (ossia che si trovano in stato 8). Questa tabella è alimentata dal batch di schedulazione dei trasferimenti giornaliero, che esegue una preventiva operazione di pulizia del log dei trasferimenti e dell'ambiente temporaneamente creato a corredo del trasferimento per tutti quei trasferimenti che si sono correttamente conclusi (stato 8 sull'archivio in linea), previa memorizzazione sullo storico del record che si sta cancellando dall'archivio in linea.

In Fig. 5.4.3.a e 5.4.3.b sono riportate le strutture rispettivamente degli archivi sequenziali di cui ai punti 1. e 2. e della tabella DB2 di cui al punto 3.

Elenco degli attributi dell'archivio di log dei trasferimenti						
Nome Colonna	Tipo dato	Null	Valori ammessi	Prim. key	For. Key	Tab. design.
package	X(8)					
nome_area	X(8)					
from_config	X(8)					
data_ins	X(8)		yyyymmdd			
ora_ins	X(6)		hhmmss			
stato_trasf	X(1)		1,2,3,4,5,6,7,8			
num_job_subm	9(5)					
num_job_ended	9(5)					

Fig. 5.4.3.a

Tracciato record dell'archivio sequenziale di log dei trasferimenti

Elenco degli attributi della tabella D01.TDLCLGT						
Nome Colonna	Tipo dato	Null	Valori ammessi	Prim. key	For. Key	Tab. design.
package	char(8)	NN				
nome_area	char(8)	NN				
from_config	char(8)	NN				
data_ins	char(8)	NN	yyyymmdd			
ora_ins	char(6)	NN	hhmmss			
stato_trasf	char(1)	NN	numerico			
num_job_subm	char(5)	NN	numerico			
num_job_ended	char(5)	NN	numerico			

Fig. 5.4.3.b

Tracciato della tabella DB2 D01.TDLCLGT
Storico del log dei trasferimenti

5.4.4 Il log degli eventi

Il log degli eventi è una tabella DB2 D01.TDLCLGE che risiede sul sottosistema DB2 di sviluppo (DB2S), consultabile da parte di tutti gli utenti, che la personalizzazione RGS del prodotto ASG/LCM aggiorna a fronte di:

- EXPORT per update degli oggetti in SVIL e MAN;
- completamento della fase di BUILD per il singolo configuration item a valle della MIGRATE da SVIL a COLL, da COLL e ESER e da MAN a ESER

Nel caso di EXPORT per UPDATE sono memorizzate nel log degli eventi le seguenti informazioni:

- Area (SYSTEM)
- Ambiente (CONFIGURATION a partire dalla quale è stato effettuato l'export)
- Tipo (TYPE)
- Oggetto (MEMBER)
- Data
- Ora
- Utente
- Export host library

Nel caso di completamento della fase di BUILD a seguito della MIGRATE saranno memorizzate nel log degli eventi le seguenti informazioni:

- Area (SYSTEM)
- Ambiente (CONFIGURATION di partenza)
- Tipo (TYPE)
- Oggetto (MEMBER)
- Data
- Ora

- Ambiente di arrivo
- Pacchetto
- Esito della fase di Build (OK, Abend)

La struttura della tabella di log degli eventi D01.TDLCLGE è riportata in Fig.5.4.4.a.

Elenco degli attributi della tabella D01. TDLCLGE						
Nome Colonna	Tipo dato	Null	Valori ammessi	Prim. key	For. Key	Tab. design.
tipo_evento	char(8)	NN	EXPORT MIGRATE			
tipo_oggetto	char(8)	NN				
nome_oggetto	char(8)	NN				
nome_area	char(8)	NN				
data_evento	Data	ND				
ora_evento	Time	ND				
from_config	char(8)	NN				
to_config	char(8)	ND	solo per MIGRATE			
package	char(8)	ND	solo per MIGRATE			
esito_build	char(30)	ND	solo per MIGRATE			
utente	char(8)	ND	solo per EXPORT			
exp_host_library	char(44)	ND	solo per EXPORT			

Fig. 5.4.4.a

Tracciato della tabella DB2 D01.TDLCLGE
Log degli eventi

5.4.5 Lo stato del trasferimento

Lo stato di avanzamento del trasferimento di un pacchetto è tenuto costantemente aggiornato dal software che lo implementa attraverso l'aggiornamento di un campo dell'archivio di log dei trasferimenti (cfr. Fig.5.4.3.a relativamente al campo *stato-trasf*) con valori che identificano la fase di elaborazione in cui si trova il trasferimento.

Questa informazione, unitamente ai contatori *num-job-subm* e *num-job-ended* sempre nell'archivio di log dei trasferimenti, consente di comprendere l'esito di un trasferimento e l'eventuale punto di interruzione.

Nella tabella di Fig. 5.4.5.a sono riportati i valori e i corrispondenti significati di questi campi.

<i>Stato-trasf</i>	<i>num-job-subm</i>	<i>num-job-ended</i>	<i>stato di avanzamento del pacchetto</i>
1	0	0	Effettuata richiesta di Migrare con l'opzione 4.5 del ASG/LCM e copiato job capostipite della migrate sulla libreria di sistema
2	0	0	La procedura che schedula i trasferimenti ha sottomesso il job capostipite sulla partizione PLT1
3	0	0	Il job capostipite è partito ma non si è ancora completato
3	n	0	Il job capostipite è partito e si è completato sottomettendo gli n job di build
3	n	N<n	La fase di build è in corso e si è completato correttamente l'm-simo job di build
3	n	N	La fase di build è in corso, l'n-simo job sta generando i jcl da eseguire a valle della fase di build su PLT1 e sulla partizione di arrivo
4	n	N	La fase di build è completata e i due job finali sono stati memorizzati sulle rispettive librerie ed ha sottomesso quello sulla partizione PLT1
5	n	N	Il job n+2-simo, sulla partizione PLT1 è partito.
6	n	N	il job n+2-simo sulla partizione PLT1 è finito correttamente e ha sottomesso via TCPIP il job n+3-simo sulla partizione di arrivo
7	n	N	il job n+3-simo è partito sulla partizione di arrivo
8	n	N	il job n+3-simo è finito correttamente sulla partizione di arrivo. Il trasferimento si può considerare correttamente concluso

Fig.5.4.5.a

5.4.6 Il protocollo di attivazione del job che completa la migrate di un pacchetto sulla partizione di arrivo

La migrate di un pacchetto si compone di due fasi successive: la prima eseguita nella partizione di PLT1, la seconda nella partizione di arrivo (PLT2 o Esercizio a seconda che la migrate abbia come Configuration di destinazione COLL o ESER).

Più specificatamente, come già spiegato in un paragrafo precedente, ogni migrate di un pacchetto è realizzata da una serie di $n+3$ job, omonimi rispetto al pacchetto. Di questi, i primi $n+2$ (serializzati grazie alla loro omonimia) sono eseguiti sulla partizione di PLT1. E' il job $n+2$ -esimo ad attivare nella partizione di arrivo il job conclusivo della catena, mediante, un servizio del TCP/IP.

Tale servizio si appoggia sul Remote Shell Protocol (RSH) del TCP/IP che permette di eseguire un comando TSO su un sistema remoto ricevendo il risultato sul sistema locale.

Il comando TSO RSH viene attivato in uno step di TSO batch dell'ultimo job della catena di job omonimi al pacchetto eseguiti su PLT1.

Tale comando permette l'attivazione del SUBMIT del job che completa la migrate nella partizione di arrivo.

Sia il job eseguito su PLT1 (host locale) che quello sottomesso su PLT2/ESER (host remoto) devono avere come OWNER RACF la userid STCTRAS.

Il formato usato per RSH è il seguente:

RSH *foreign-host comando*⁹,
dove:

- *foreign-host* è l'ip-address (o l'hostname) della partizione di destinazione,
- *comando* è il comando TSO da eseguire su tale partizione (nel nostro caso è: SUBMIT *nome-job*).

⁹ In generale, il formato del comando RSH è il seguente:

RSH -l *user/password foreign-host comando*

dove il parametro *user/password* specifica la coppia userid TSO e password dell'utenza definita su host remoto sotto la cui autorizzazione deve essere eseguito il comando indicato.

Nel caso dell'applicazione in questione, tale userid su sistema remoto è omonima a quella che sottomette il comando RSH su host locale (STCTRAS). Inoltre, è richiesto sia attivato il meccanismo che permette di non specificare la password.

Queste due condizioni consentono di semplificare il formato del comando RSH.

Per usare il comando RSH, è necessario che sia attivo un Remote Execution server su ognuna delle possibili partizioni remote (PLT2 ed Esercizio). Attraverso tale server, a fronte del lancio di un comando RSH su host locale, viene sottomesso un job (RSHDn) che, tramite uno step di TSO batch, esegue il comando TSO specificato.

Per configurare il Remote server sono stati effettuati i passi sotto elencati sia su PLT2 che su Esercizio:

1. sono state aggiornate le sezioni AUTOLOG e PORT del file TCPIP.PROFILE.TCPIP:
 - includendo in AUTOLOG il nome del membro contenente la procedura di Remote Execution (RXPROC, nel nostro caso),
 - aggiungendo nella sezione PORT il record:
514 TCP RXPROC ; shell.
2. è stata creata la procedura RXPROC di Remote Execution a partire dal sample TCPIP.SEZAINST(RXPROC). In questa procedura occorre personalizzare i parametri seguendo i commenti del sample. In particolare:
 - TSOPROC=TSOBATCH (nome della procedura di TSO batch TSOBATCH definita in SYS1.PROCLIB),
 - MSGCLASS=X (è il valore di MSGCLASS per i job RSHDn sottomessi dal server),
 - TSCLASS=W (è la classe di SYSOUT per la scheda di SYSTSPRT dei job RSHDn sottomessi dal server),
 - EXIT=RXUEXIT (è il nome della user exit routine che imposta CLASS=I per i job RSHDn. La classe I è dedicata all'esecuzione di tali job. Senza la personalizzazione realizzata dalla exit, i job verrebbero eseguiti nella classe di default, A). Il load di questa exit è stato copiato sulla libreria di STEPLIB del server: TCPIP.SEZATCP¹⁰.
3. per consentire l'uso 'protetto' del comando RSH senza password, è stato creato il data set STCTRAS.RHOSTS.DATA contenente il record:
MVSSVIL stctras.
dove:
 - MVSSVIL è l'hostname del sistema locale (PLT1, cioè) definito nel data set TCPIP.HOSTS.LOCAL,
 - stctras è la userid TSO (in minuscolo) che esegue su host locale il comando RSH.

Oltre a questo, sono state effettuate le definizioni RACF che consentono al Remote Execution server di sottomettere job, aventi USER=STCTRAS nella scheda job senza fornire la password (vedi cap. 4.4.4).

In tal modo, solo l'utenza STCTRAS dall'host locale PLT1 (e solo da questo) potrà eseguire un comando TSO su PLT2 o ESER,

¹⁰ Il sorgente e l'eseguibile della exit sono presenti, rispettivamente, sulle librerie C0T4.LCMSW10.SOURCE e C0T4.LCMSW10.LOAD mentre il job di compilazione relativo (\$LCJASME) è memorizzato su C0T4.LCMSW10.CNTL.

grazie a RSH, qualificandosi come STCTRAS su tali sistemi, senza fornire la password corrispondente.

5.4.7 Lo schedulatore dei trasferimenti

Lo schedulatore dei trasferimenti LCPLGSCH è un programma di LCMSW v.1.0 che, a partire dal log dei trasferimenti (scheda DD LCALOG), in base all'input fornitogli nella scheda DD LCASYSIN, sottomette i job omonimi rispetto ai pacchetti che soddisfano le condizioni in input a partire dalla libreria fornitagli nella scheda DD LCAJOB. Lo stato dei pacchetti sottomessi viene portato a 2.

L'input fornito in LCASYSIN deve essere fornito secondo la sintassi indicata:

- **AMB=SVIL|MAN|COLL**, parametro obbligatorio.
Può assumere uno ed uno solo fra i valori indicati.
Serve ad indicare l'ambiente di provenienza dei pacchetti individuati dai parametri successivi.

Di seguito, senza spazi intermedi e con la parola chiave (INCL, ESCL, PACKID o TUTTO) sulla stessa riga, uno ed uno solo tra i seguenti parametri, mutuamente esclusivi,:

- **INCL** (con sintassi **INCL=area1,area2, ... ,areaN**), con cui indicare le sole aree per i cui pacchetti si vuole che siano effettuati i trasferimenti.
In altre parole, specificando questo parametro saranno sottomessi solo i *job* di trasferimento da AMB registrati nel *log* per una delle aree elencate.
La lista di aree può essere riportata su più schede di input, andando a capo dopo una virgola e continuando nella scheda successiva ad una qualunque colonna.
Qualunque informazione scritta dopo uno spazio sarà ignorata.
- **ESCL** (con sintassi **ESCL=area1,area2, ... ,areaN**), con cui indicare le aree i cui pacchetti non dovranno essere trasferiti.
In altre parole, specificando questo parametro saranno sottomessi tutti i *job* di trasferimento da AMB registrati nel *log* ad eccezione di quelli associati ad una delle aree elencate.
La lista di aree può essere riportata su più schede di input, andando a capo dopo una virgola e continuando nella scheda successiva ad una qualunque colonna.
Qualunque informazione scritta dopo uno spazio sarà ignorata.
- **PACKID** (con sintassi **PACKID=packid1,packid2, ... ,packidN**), con cui indicare i soli pacchetti che si vogliono trasferire.

In altre parole, specificando questo parametro saranno sottomessi solo i *job* di trasferimento da AMB registrati nel *log* che siano relativi ai pacchetti elencati.

La lista di pacchetti può essere riportata su più schede di input, andando a capo dopo una virgola e continuando nella scheda successiva ad una qualunque colonna.

Qualunque informazione scritta dopo uno spazio sarà ignorata.

– **TUTTO.**

Specificando questo parametro saranno sottomessi tutti i *job* di trasferimento da AMB.

5.4.7 Il job HTRASCCC di schedulazione dei trasferimenti nel batch standard

I trasferimenti di pacchetti nel batch standard sono attivati dal job HTRASCCC schedato sul sistema PLT1¹¹.

Il job:

- richiede che siano attivi tutti i sottosistemi DB2 (DB2S, DB2M su PLT1, DB2L su PLT2 e DB2P su esercizio),
- è eseguito in classe 9,
- è sottomesso con il parametro USER=STCTRAS sulla scheda job, senza che sia indicata la password (cfr.4.4.4).

Il job prevede i seguenti passi logici:

1. esecuzione della pulizia preventiva del log dei trasferimenti e dell'ambiente al contorno per tutti quei trasferimenti che nel log risultano correttamente completati (stato 8). La pulizia prevede:
 - la cancellazione dei job relativi al pacchetto dalle librerie di cui alla sez.5.4.1.1, 5.4.1.2, 5.4.1.3
 - la cancellazione dei target che costituiscono il pacchetto dalle librerie in ambiente condiviso, per i trasferimenti verso ESER;
 - l'inserimento nello storico dei trasferimenti dell'occorrenza relativa al trasferimento cancellato;
 - la cancellazione dell'occorrenza relativa al pacchetto dal log dei trasferimenti;
 - la cancellazione della libreria H0T4.&packid..COMMAND di cui alla sez. 5.4.1.4.
2. esecuzione dello schedatore (LCPLGSCH) che sottomette tutti i pacchetti che trova in stato 1 nel log dei trasferimenti e che provengono da MAN (AMB=MAN,TUTTO);

¹¹ Nel periodo transitorio della messa in linea (cfr. [5]) questo batch segue i batch HCLEDAIL e HBINDPGM sulla medesima partizione, che completano i trasferimenti implementati con THOST e PROGRAM DIRECTORY. A seguire la messa in linea definitiva questi vincoli potranno essere rimossi.

3. esecuzione dello schedulatore (LCPLGSCH) che sottomette tutti i pacchetti che trova in stato 1 nel log dei trasferimenti e che provengono da SVIL (AMB=SVIL,TUTTO).
4. esecuzione dello schedulatore (LCPLGSCH) che sottomette tutti i pacchetti che trova in stato 1 nel log dei trasferimenti e che provengono da COLL (AMB=COLL,TUTTO);

I job sottomessi ai passi 2., 3., 4. sono i job capostipite di cui alla sez. 5.4.2.1.

5.4.8.1 Variazioni alla schedulazione dei trasferimenti

Si può verificare che, a fronte di particolari esigenze applicative o della conduzione sia necessario modificare la schedulazione dei trasferimenti come descritta ai punti 2., 3. 4. della sez. 5.4.8. Se, per esempio, si vogliono inibire i trasferimenti da MAN per l'area SPE, è possibile procedere come segue:

- predisporre su NIP.SYSIN un membro, per esempio di nome HMANXXX, contenente le condizioni di selezione per lo schedulatore
AMB=MAN,ESCL=SPE;
- attraverso l'opzione DAJ
3 TRASFERIMENTO DA AMBIENTE DI PRODUZIONE AD
 AMBIENTE DI TEST

portare il job HTRASCCC sulla libreria di sviluppo NIP.LIBJOB e modificarlo indicando nel richiamo della procedura HTRASCCC il parametro TRASMAN=HMANXXX;

- richiedere il trasferimento urgente del job HTRASCCC con l'opzione
C Trasferimenti urgenti job e procedure in
 esercizio
del menù 9.9.

5.4.9 La procedura HTRASURG di attivazione dei trasferimenti urgenti

La procedura HTRASURG è una started procedure presente in SYS1.STARTLIB che attiva lo schedulatore dei trasferimenti di cui alla sez. 5.4.9 che schedula i pacchetti che soddisfano la condizione fornita in input (cfr. sez. 4.4.x per quanto concerne i profili di sicurezza richiesti dalla procedura).

La sintassi di attivazione da console è:

S HTRASURG,MEMBER=XXXXXXXX

dove XXXXXXXX è un membro della libreria NIP.SYSIN che contiene l'input allo schedulatore dei trasferimenti secondo la sintassi espressa alla sez.5.4.7.

Per esempio se è richiesto il trasferimento urgente del pacchetto C040303A dall'ambiente MAN, si dovrà:

- predisporre in NIP.SYSIN un membro (per esempio HPIPP0) con i comandi per lo schedulatore:
AMB=MAN,PACKID=C040303A
- chiedere alla sala macchine del PLT1 di sottomettere ovvero sottomettere direttamente attraverso la console SDSF il comando:
S HTRASURG,MEMBER=HPIPP0.

Come per il batch schedulato di cui alla sez. 5.4.9, la procedura per funzionare necessita, al momento dell'attivazione:

- che siano attivi i sottosistemi DB2 DB2S e DB2M su PLT1 e il sottosistema DB2 della partizione di destinazione del trasferimento (DB2L se AMB=SVIL, DB2P se AMB=MAN|COLL)
- che sia aperto un initiator in classe 9 sia sulla partizione PLT1, sia sulla partizione di destinazione del trasferimento;
- che sia attivato correttamente secondo le modalità descritte alla sez.5.4.6 il meccanismo TCPIP che consente di sottomettere a partire dalla partizione PLT1 un job sulla partizione di destinazione.

5.4.10 L'esito del trasferimento e l'individuazione del punto di interruzione

Il giorno successivo alla richiesta di migrate del pacchetto, si può verificare l'esito dei trasferimenti, andando in browse sull'archivio di consultazione giornaliera H0T4.TRASF.LCLOGBK e controllando il valore del campo stato-trasf (offset 39 - cfr. sez.5.4.3 per il tracciato completo dell'archivio).

Se stato-trasf = 8, il trasferimento si è correttamente concluso.

Se stato-trasf = 1, il trasferimento non è stato schedulato nel batch serale.

Se $1 < \text{stato-trasf} < 8$, il trasferimento si è interrotto in maniera anomala.

Tuttavia, nel caso in cui si tratti di un trasferimento proveniente da SVIL e il trasferimento si sia interrotto con una situazione analoga a quella ombreggiata nella tabella di Fig. 8.4.a (stato_trasf = 3, num-job-ended = m < num job started = n), la causa dell'interruzione è da ricercare nei job di build che effettuano la compilazione degli oggetti, alcuni dei quali non si sono completati correttamente.

Un restart del trasferimento potrebbe portare al ripetersi dell'errore, se il problema è nella compilazione.

Per individuare il/i job di build andato/i male, fra gli n+1 sottomessi per il pacchetto, si dovrà procedere come segue:

- individuare i configuration item la cui build è andata in errore, acquisendo dal log degli eventi, la lista dei record relativi alla migrate del pacchetto in esame, ordinata per data e ora ascendente;
- dalla lista verificare quali degli n job di build sottomessi sono andati in errore, visto che il log degli eventi è scritto rispettando l'ordine con il quale i job sono sottomessi e quindi eseguiti;
- supponiamo che siano andati in errore il 3° e il 5° degli n job di build (il configuration item coinvolto lo si individua già dal log degli eventi); ottenendo, attraverso SDSF o l'opzione 3.8 dell'ISPF su PLT1, l'elenco dei job omonimi rispetto al pacchetto che sono in coda di output, si dovrà andare a controllare il 4° e il 6° in ordine di esecuzione.

5.4.11 Il restart del trasferimento

5.4.11.1 Trasferimento in stato 2

Se il trasferimento di un pacchetto è in stato 2, vuol dire che il job relativo è stato sottomesso, ma non è partito.

Possibili motivi:

1. non c'erano initiator in classe 9 (in questo caso dovrebbero essere tutti i pacchetti in stato 2);
2. il job è stato cancellato;
3. il pacchetto aveva la scheda job codificata male ed è stato sottomesso in una classe per la quale non c'erano initiator disponibili, oppure è andato in jcl error (controllare sulla syslog e sulla libreria H0T4.TRASF.JOB).

Un trasferimento in queste condizioni è di fatto un trasferimento non partito.

Se il problema si è verificato puntualmente (caso 2. e 3.) la prassi per il restart è la seguente:

- cancellare il record dal log utilizzando il jcl HCANC2 sulla libreria H0T4.TRASF.CNTL avendo cura di indicare correttamente il nome del pacchetto (il jcl opera su un solo pacchetto per volta);
- cancellare il membro il cui nome coincide con quello del pacchetto dalla libreria H0T4.TRASF.JOB.
- contattare l'utente che ha chiesto il restart del pacchetto ovvero i referenti dell'area proprietaria del pacchetto e dir loro di effettuare nuovamente la richiesta di migrate del pacchetto secondo le modalità indicate in [1], codificando correttamente la scheda job.

Se il problema è stato generale (caso 1) si devono riportare tutti i pacchetti coinvolti in stato 1 sottomettendo per ciascuno di essi il job HFORCE con il parametro stato=1, presente sulla libreria H0T4.TRASF.CNTL in modo tale che vengano schedulati nel successivo batch standard.

5.4.11.2 Trasferimento in stato 3

Come evidenziato nella tabella in Fig. 5.4.10.a questo caso si articola in tre sottocasi distinti in base ai valori dei contatori del numero di job sottomessi (num_job_subm) e del numero di job completati (num_job_ended).

Analizziamoli uno alla volta:

$num_job_subm = 0 ; num_job_ended = 0$

Il problema si è verificato nel job capostipite (cfr.5.4.2.1 per i riferimenti ai passi fatti di seguito) e, in funzione del punto di arresto, il problema si può individuare come segue:

1. il job si è arrestato ai passi antecedenti il passo 5; il problema si è verificato nel codice di personalizzazione RGS (per esempio non è stato trovato spazio per l'allocazione della libreria H0T4.&packid..COMMAND sul volume indicato per l'allocazione);
2. il job si è arrestato al passo 5. o al passo 6. sotto il controllo del ASG/LCM. Deve essere consultato l'output dello step e verificato il motivo del malfunzionamento con l'ausilio della diagnostica del ASG/LCM e della manualistica del prodotto (cfr. [3] per la spiegazione della diagnostica del prodotto). Deve inoltre essere effettuato un summary report del pacchetto con l'opzione 4.8 del prodotto (cfr. sez 6.10.6 del manuale [1]) per verificare quale è lo stato del pacchetto e in base a questo determinare se la migrate per il prodotto è completata (stato del pacchetto coincidente con la configurazione di arrivo) o è incompleta (stato del pacchetto coincidente con la configurazione di partenza).

Una volta individuato e rimosso il problema si deve procedere al restart del trasferimento come segue:

- per un trasferimento interrotto ai passi descritti al punto 1., si deve riportare il pacchetto in stato 1 con il job HFORCE sulla libreria H0T4.TRASF.CNTL affinché venga schedulato nel batch serale, ovvero effettuare un trasferimento urgente (cfr. sez.5.4.9) a seconda delle necessità dell'utente;
- per un trasferimento interrotto ai passi descritti al punto 2. e che si trovi in stato incomplete si deve:

- sottomettere il job HCANC3 per il pacchetto in errore in modo da cancellare dal log il record relativo al trasferimento;
- cancellare il membro il cui nome coincide con quello del pacchetto dalla libreria H0T4.TRASF.JOB.
- contattare l'utente affinché inneschi nuovamente l'iter di richiesta di Approve, Approve, Migrate del pacchetto., come descritto in [1] avendo cura di specificare il parametro PARTIAL MIGRATE = NO.
- per un trasferimento interrotto ai passi descritti al punto 2. per il quale la migrate sia completata si deve:
 - effettuare il fallback del pacchetto con l'opzione 4.6 del prodotto;
 - sottomettere il job HCANC3 per il pacchetto in errore in modo da cancellare dal log il record relativo al trasferimento;
 - cancellare il membro il cui nome coincide con quello del pacchetto dalla libreria H0T4.TRASF.JOB.
 - contattare l'utente affinché inneschi nuovamente l'iter di richiesta di Approve, Approve, Migrate del pacchetto, come descritto in [1] avendo cura di specificare il parametro PARTIAL MIGRATE = NO.

$num_job_subm = n ; num_job_ended = 0$

La fase di analyze e migrate del job capostipite si è conclusa correttamente con la sottomissione degli n job di build ma questi o non sono partiti (problemi sugli initiator in classe 9) o si sono conclusi tutti in maniera anomala per problemi di sistema.

Dall'analisi dell'output dei job di build e dal log degli eventi¹² si deve individuare la causa, correggerla e quindi effettuare il restart del trasferimento secondo la prassi che segue:

- effettuare il fallback del pacchetto con l'opzione 4.6 del prodotto come descritto in [1];
- sottomettere il job HCANC3 per il pacchetto in errore in modo da cancellare dal log il record relativo al trasferimento;

¹² Se nel log degli eventi non sono registrate entrate per alcuno dei job del pacchetto, si è trattato di un jcl error, in caso contrario di unabend.

- cancellare il membro il cui nome coincide con quello del pacchetto dalla libreria H0T4.TRASF.JOB.
- contattare l'utente affinché inneschi nuovamente l'iter di richiesta di Approve, Approve, Migrate del pacchetto, come descritto in [1].

$num_job_subm = n ; num_job_ended = m \text{ con } m < n$

Non tutti i job di build (n) sottomessi si sono correttamente completati e quindi non si è completata la fase di build.

Il problema è probabilmente di natura applicativa, soprattutto se si tratta di un trasferimento da SVIL (problema di compilazione) o da MAN (per esempio non si trova un eseguibile del pacchetto in quanto non si è fatta la compilazione in MAN).

Per individuare il/i job di build andato/i male, fra gli n+1 sottomessi per il pacchetto, si dovrà procedere come segue:

- individuare i configuration item la cui build è andata in errore, acquisendo dal log degli eventi, la lista dei record relativi alla migrate del pacchetto in esame, ordinata per data e ora ascendente;
- dalla lista verificare quali degli n job di build sottomessi sono andati in errore, visto che il log degli eventi è scritto rispettando l'ordine con il quale i job sono sottomessi e quindi eseguiti;
- supponiamo che siano andati in errore il 3° e il 5° degli n job di build (il configuration item coinvolto lo si individua già dal log degli eventi); ottenendo, attraverso SDSF o l'opzione 3.8 dell'ISPF su PLT1, l'elenco dei job omonimi rispetto al pacchetto che sono in coda di output, si dovrà andare a controllare il 4° e il 6° in ordine di esecuzione.

In questo caso è necessario:

- effettuare il fallback del pacchetto con l'opzione 4.6 del prodotto come descritto in [1];
- sottomettere il job HCANC3 per il pacchetto in errore in modo da cancellare dal log il record relativo al trasferimento;
- cancellare il membro il cui nome coincide con quello del pacchetto dalla libreria H0T4.TRASF.JOB.
- rimuovere la causa dell'errore eventualmente contattando l'utente se il problema è applicativo, come probabile in questo caso;
- contattare l'utente affinché inneschi nuovamente l'iter di richiesta di Approve, Approve, Migrate del pacchetto, come descritto in [1].

$num_job_subm = n ; num_job_ended = n$

I job di build si sono correttamente completati, ma si è verificato un errore nella generazione dei job che devono espletare la fase comune a valle della fase di build su PLT1 e sulla partizione di arrivo. Va individuato e rimosso l'errore che si è verificato nell'ultimo job di build sottomesso ed eseguito per il pacchetto, a valle dello step LGUCT010 che ha restituito codice di ritorno = 6.

Per il restart del pacchetto in questo caso si deve:

- effettuare il fallback del pacchetto con l'opzione 4.6 del prodotto come descritto in [1];
- sottomettere il job HCANC3 per il pacchetto in errore in modo da cancellare dal log il record relativo al trasferimento;
- cancellare il membro il cui nome coincide con quello del pacchetto dalla libreria H0T4.TRASF.JOB.
- contattare l'utente affinché inneschi nuovamente l'iter di richiesta di Approve, Approve, Migrate del pacchetto, come descritto in [1].

5.4.11.3 Trasferimento in stato 4

La fase di build si è completata correttamente con la memorizzazione del job n+2-simo sulla libreria H0T4.TRASF.PLT1 e del job n+3-simo sulla libreria H0T4.TRASF.PLT2/(ESER) e la sottomissione del primo che però non è partito.

Va identificata e rimossa la causa per la quale il job n+2-simo non è partito. A questo punto per effettuare il restart del trasferimento si deve procedere con modalità analoga a quella descritta per i trasferimenti urgenti (cfr. sez.5.4.9), come di seguito descritto:

- predisporre in NIP.SYSIN un membro (per esempio HPIPP0) con i comandi per lo schedulatore:
AMB=MAN/SVIL/COLL,PACKID=(packid-1,packid2,...)
ovvero aggiornare il membro HTRASPL1 che è il default della procedura:
- chiedere alla sala macchine del PLT1 di sottomettere ovvero sottomettere direttamente attraverso la console SDSF il comando:
S HTRASPL1,MEMBER=HPIPP0
o
S HTRASPL1
a seconda che si sia predisposto un membro di sysin ad hoc o che si sia scritto sul membro di default.

La procedura in questo caso a partire dal log dei trasferimenti schedula i pacchetti che soddisfano la condizione fornita in input e che si trovano in stato 4.

La procedura e l'intero processo da questa innescato hanno i medesimi requisiti di sistema già indicati alla sez. 5.4.9 per i trasferimenti urgenti.

5.4.11.4 Trasferimento in stato 5

Il job n+2-simo di cui alla sez. 5.4.2.4 è partito ma non si è concluso correttamente.

Va quindi, analizzando l'output del job, ricercata e rimossa la causa di errore del job stesso.

Quindi per procedere al restart del trasferimento si deve:

- andare in edit sul job che si trova sulla libreria H0T4.TRASF.PLT1 e codificare sulla scheda job dello stesso il parametro restart impostato al valore dello step andato in errore da cui riprendere l'elaborazione;
- predisporre in NIP.SYSIN un membro (per esempio HPIPP0) con i comandi per lo schedulatore:
AMB=MAN/SVIL/COLL,PACKID=(packid-1,packid2,...)
ovvero aggiornare il membro HTRASPL1 che è il default della procedura:
- chiedere alla sala macchine del PLT1 di sottomettere ovvero sottomettere direttamente attraverso la console SDSF il comando:
S HTRASPL1,MEMBER=HPIPP0,STATO=5
o
S HTRASPL1,STATO=5

a seconda che si sia predisposto un membro di sysin ad hoc o che si sia scritto sul membro di default.

La procedura in questo caso a partire dal log dei trasferimenti schedula i pacchetti che soddisfano la condizione fornita in input e che si trovano in stato 5.

La procedura e l'intero processo da questa innescato hanno i medesimi requisiti di sistema già indicati alla sez. 5.4.9 per i trasferimenti urgenti.

5.4.11.5 Trasferimento in stato 6

Il job n+2-simo di cui alla sez. 5.4.2.4 si è concluso correttamente, ma la schedulazione del job n+3-simo sulla partizione di arrivo non è andata a buon fine.

Va quindi, analizzando l'output del job, ricercata e rimossa la causa di errore del job stesso.

Quindi per procedere al restart del trasferimento si deve:

- predisporre in NIP.SYSIN un membro (per esempio HPIPPPO) con i comandi per lo schedulatore:
AMB=MAN/SVIL/COLL,PACKID=(packid-1,packid2,...)
ovvero
aggiornare il membro HTRASPL2 (HTRASESE) che sono i default in funzione della destinazione del trasferimento.
- chiedere alla sala macchine del PLT2 o di esercizio, in funzione della destinazione del trasferimento di sottomettere ovvero sottomettere direttamente attraverso la console SDSF il comando:
S HTRASPL2,MEMBER=HPIPPPO,STATO=6
o
S HTRASPL2,STATO=6

su PLT2

```
S HTRASESE,MEMBER=HPIPPPO,STATO=6
o
S HTRASESE,STATO=6
su esercizio
```

a seconda che si sia predisposto un membro di sysin ad hoc o che si sia scritto sul membro di default e a seconda il pacchetto sia destinato su PLT2 o su esercizio.

La procedura in questo caso a partire dal log dei trasferimenti schedula i pacchetti che soddisfano la condizione fornita in input e che si trovano in stato 6.

La procedura e l'intero processo da questa innescato richiedono solo l'attività del sottosistema DB2 della partizione di arrivo

5.4.11.6 Trasferimento in stato 7

Il job n+3-simo di cui alla sez. 5.4.2.5 è partito ma non si è concluso correttamente sulla partizione di destinazione.

Va quindi, analizzato l'output del job, ricercata e rimossa la causa di errore del job stesso.

Quindi per procedere al restart del trasferimento si deve:

- andare in edit sul job che si trova sulla libreria H0T4.TRASF.PLT2 (ESER) e codificare sulla scheda

job dello stesso il parametro restart impostato al valore dello step andato in errore da cui riprendere l'elaborazione;

- predisporre in NIP.SYSIN sulla partizione di destinazione ¹³ un membro (per esempio HPIPPPO) con i comandi per lo schedulatore:

AMB=MAN/SVIL/COLL,PACKID=packid-1

ovvero

aggiornare il membro HTRASPL2 (HTRASESE) che sono i default in funzione della destinazione del trasferimento.

- chiedere alla sala macchine del PLT2 o di esercizio, in funzione della destinazione del trasferimento di sottomettere ovvero sottomettere direttamente attraverso la console SDSF il comando:

S HTRASPL2,MEMBER=HPIPPPO,STATO=7

o

S HTRASPL2,STATO=7

su PLT2

S HTRASESE,MEMBER=HPIPPPO,STATO=7

o

S HTRASESE,STATO=7

su esercizio

a seconda che si sia predisposto un membro di sysin ad hoc o che si sia scritto sul membro di default e a seconda il pacchetto sia destinato su PLT2 o su esercizio.

La procedura in questo caso a partire dal log dei trasferimenti schedula i pacchetti che soddisfano la condizione fornita in input e che si trovano in stato 7.

La procedura e l'intero processo da questa innescato richiedono solo l'attività del sottosistema DB2 della partizione di arrivo.

5.4.12 Cancellazione della richiesta di Migrate di un pacchetto

Laddove un utente effettui la richiesta di cancellazione dei trasferimenti attraverso il modulo riportato in Allegato 2 deve essere sottomesso il jcl HCANC1 appoggiato sulla libreria H0T4.TRASF.CNTL opportunamente personalizzato con il nome del pacchetto.

5.5. LA GESTIONE DEGLI UTENTI NEL ASG/LCM

¹³ Attualmente esistono due NIP.SYSIN: una su COMUNE, condivisa a PLT1 e esercizio e una su un volume visibile solo su PLT2

Le utenze di ASG/LCM devono essere gestite tramite l'opzione CCC.1.1 e coincidere con le chiavi TSO. La password iniziale da assegnare a ogni utenza è RGSCDC ma per garantire maggior sicurezza l'utente può modificarla tramite l'opzione CCC.0.1.

Nel campo indicato sotto (ultimo parametro del pannello) va inserita la nuova password senza necessità di specificare quella vecchia.

CHANGE CCC USER PASSWORD:

NEW PASSWORD ==>

5.5.1 La definizione/cancellazione degli utenti al ASG/LCM

Le operazioni di definizione e cancellazione di un'utenza possono essere effettuate solamente da un DBA.

E' possibile definire nuove utenze al prodotto ASG/LCM solo se già definite come chiavi TSO.

Per inserire una nuova utenza a ASG/LCM occorre utilizzare l'opzione CCC.1.1.

Nella schermata relativa occorre inserire il nome dell'utenza, la password che inizialmente va impostata a RGSCDC e la classe che indica la tipologia di utenza. La classe, pur non costituendo un vincolo del prodotto, dovrà coincidere con il nome del gruppo RACF corrispondente, quindi sarà CCCUSxxx nel caso di programmatore e CCCCCPxxx nel caso di capo progetto dell'area xxx.

Il fatto di avere specificato la classe con il nome del gruppo CCC, al momento della creazione, non è sufficiente ad assegnare al nuovo utente le funzioni di sua competenza.

A tale scopo occorre utilizzare l'opzione CCC.4.S.2.U., specificando il system xxx. cui l'utente appartiene, e effettuare l'assign dell'utenza al gruppo CCCUSxxx oppure a CCCCCPxxx.

Nel caso il gruppo non sia definito occorre crearlo con l'opzione CCC.4.S.2.A.

In parallelo occorre aggiungere l'utenza nei gruppi RACF CCCUSxxx o CCCCCPxxx.

Questo oltre a garantire l'accesso alle risorse dell'area xxx nel DB RGSCDC, permette di accedere alla procedura di logon ALOGCCC attraverso la quale è possibile utilizzare il prodotto.

In caso di cancellazione di un'utenza il pannello da utilizzare è quello legato all'opzione CCC.1.1 dove occorre indicare il nome da cancellare.

Con l'opzione CCC.4.S.2.U l'utenza da cancellare va sottoposta a DEASSIGN all'interno del gruppo.

In parallelo occorre cancellare l'utenza dal gruppo RACF di appartenenza.

5.6 LA CANCELLAZIONE DI UN OGGETTO DAL DATABASE RGSCDC

Laddove sia richiesta la cancellazione definitiva di un oggetto dal database del ASG/LCM attraverso il modulo riportato in Allegato 2¹⁴, si dovrà procedere con le seguenti operazioni:

1. con l'opzione M - Member utility attivata dal menù principale del prodotto ASG/LCM si deve procedere alla cancellazione (opzione Delete) dell'oggetto su tutte e 4 quattro le configurazioni (SVIL, COLL, MAN, ESER) del sistema (area) specifico del prodotto;
2. cancellazione dell'oggetto dalla base dati DB2 del prodotto LCMSW attraverso lo statement SQL

```
DELETE FROM D01.TDLCOGG  
WHERE TIPO_OGGETTO = 'tipo-oggetto-da cancellare' AND  
NOME_OGGETTO = 'nome-oggetto-da-cancellare'
```

sottomesso interattivamente sul DB2 di sviluppo attraverso SPUI o QMF ovvero batch con DSNTIAD da un'utenza riconosciuta dal DB2 come D02 o D01 o da un SYSADM.

3. la cancellazione dell'oggetto dalle export host library di tutti gli ambienti nel quale è presente.
4. la cancellazione degli eventuali target item associati dalle target library di tutti gli ambienti in cui è presente nonché da quelle sulla partizione di esercizio

La cancellazione di un oggetto dell'area xxx dalle configurazioni SVIL e COLL può essere fatta dal capo progetto dell'area xxx, mentre dalle configurazioni COLL e ESER solo dal DBA.

L'operazione di cancellazione di un oggetto, nel caso sia presente nelle quattro configurazioni, deve essere fatta per tutte e quattro le configurazioni pena l'impossibilità di ricrearne uno con lo stesso nome; questo poichè il prodotto, per come è stato personalizzato, verifica, prima della creazione di un oggetto, la non presenza nelle altre configurazioni.

5.7 MIGLIORAMENTO DELLE PRESTAZIONI DELL'ANALYZE

Le work area di Configurazione SVIL, COLL, MAN, ESER, create per accelerare il processo di ANALYZE vanno periodicamente ricreate per rendere maggiormente attendibile il miglioramento prestazionale.

La modalità di refresh delle work aree di Configurazione non è attualmente schedulata in nessun job batch, pertanto viene effettuata all'occorrenza. Per informazioni su quando occorre effettuare il refresh delle work aree si rimanda al paragrafo 6.4.

¹⁴ Si intende con il termine cancellazione definitiva di un oggetto la sua rimozione da tutte le configurazioni dell'area di appartenenza

6. LA MANUTENZIONE DEGLI AMBIENTI A SUPPORTO DEL C.V.S. CON IL ASG/LCM

Per manutenzione degli ambienti a supporto del C.V.S. non si intendono eventuali upgrade del prodotto ASG/LCM, bensì modifiche ed implementazioni della struttura del database del prodotto, quali ad esempio, la definizione di una nuova area, di una nuova tipologia di oggetti, di un nuovo sottotipo nell'ambito dei programmi

6.1. DEFINIZIONE DI UNA NUOVA AREA

I passi da effettuare per la creazione di una nuova area, che per comodità chiameremo abc, richiedono di operare con una procedura di logon che attivi il ASG/LCM nativo e sono i seguenti:

1. definizione del system abc con l'opzione interattiva CCC 1.4 del prodotto specificando C copy nel campo OPTION e XXX¹⁵ nel campo FROM SYSTEM
2. personalizzazione con l'opzione CCC 3.1.1 delle relazioni fra i type apportando le seguenti modifiche:
 - a. nei secondary components delle relazioni di impact, reference e cross reference è necessario correggere i riferimenti ai type del system abc perché il prodotto introduce un errore modificando i valori del template da xxx a abc nei campi type e member
 - b. tutti i secondary components che si riferiscono al generico system yyy vanno ripetuti per quanti sono i cross system di abc (a parte SSY) modificando opportunamente yyy con il corretto identificativo dei sistemi cross
3. modifica con l'opzione CCC 3.1.2 dei parameter set come di seguito indicato;
 - a. cambiare XXX con abc
 - b. cambiare il parametro PACKOWN x02 con quello coerente per l'area
 - c. cambiare WRS.PGM.LOADx con la libreria congruente per l'area
4. dimensionamento ed allocazione delle export e target host library di SVIL e MAN specifiche di area;
5. definizione dei gruppi RACF di area CCCCCPabc e CCCUSabc e dei profili RACF per l'accesso al database RGSCDC e alle export host library dell'area come descritto alla sezione 4.4.1 e 4.4.2;
6. modifica con l'opzione CCC 4.s.1 DEFINE HOST DATASET delle export host library a livello di configuration cambiando xxx in abc laddove compare;

¹⁵ XXX è il nome del system predisposto sul database RGSCDC predisposto come template (sistema campione) da cui partire per la definizione di nuovi sistemi.

7. Caricare l'eventuale software dell'area nella configurazione BASELINE
8. Effettuare l'analyze per il software caricato
9. definizione con l'opzione CCC 4.s.2 GROUPS dei gruppi di autorizzazioni ASG/LCM ad operare sui pacchetti:

CCCUSabc con il seguente profilo di autorizzazioni
 FUNCTION PACKAGE ID

1	Cnn*
1	Ann*
1	Mnn*
1	Enn*
1	Snn*
1	Fnn*
1	Inn*
1	Tnn*
5	Cnn*
5	Ann*
5	Mnn*
5	Enn*
5	Snn*
5	Fnn*
5	Inn*
5	Tnn*
6	Cnn*
6	Ann*
6	Mnn*
6	Enn*
6	Snn*
6	Fnn*
6	Inn*
6	Tnn*
8	Cnn*
8	Ann*
8	Mnn*
8	Enn*
8	Snn*
8	Fnn*
8	Inn*
8	Tnn*

CCCCPabc con il seguente profilo di autorizzazioni
 FUNCTION PACKAGE ID

1	Cnn*
1	Ann*
1	Mnn*
1	Enn*
1	Snn*
1	Fnn*
1	Inn*
1	Tnn*
2	Cnn*
2	Ann*
2	Mnn*
2	Enn*
2	Snn*
2	Fnn*
2	Inn*
2	Tnn*
3	Cnn*
3	Ann*
3	Mnn*
3	Enn*
3	Snn*
3	Fnn*
3	Inn*
3	Tnn*
5	Cnn*
5	Ann*
5	Mnn*
5	Enn*
5	Snn*
5	Fnn*
5	Inn*
5	Tnn*
6	Cnn*
6	Ann*
6	Mnn*
6	Enn*
6	Snn*
6	Fnn*
6	Inn*
6	Tnn*
8	Cnn*
8	Ann*
8	Mnn*
8	Enn*
8	Snn*

8 Fnn*
 8 Inn*
 8 Tnn*

dove nn è l'identificativo numerico dell'area.

10. creare le configurazioni SVIL, MAN, COLL, ESER a partire da BASELINE
11. modificare l'initial packge status con l'opzione CCC 2.1 portandolo a SVIL
12. definire con l'opzione 4.2 Manage della Package Functions i template SnnTEMPL e MnnTEMPL con le seguenti caratteristiche

<i>Nome</i>		<i>Stato iniziale:SVIL</i>		
<i>template:SnnTEMPL</i>				
<i>Conf</i>	<i>Conf.</i>	<i>Approve</i>	<i>Promote</i>	<i>Clear</i>
<i>.partenza</i>	<i>Arrivo</i>		<i>to host</i> ¹⁶	<i>approvals</i>
SVIL	COLL	CCCCPabc	Y	N
COLL	ESER	CCCCPabc	Y	N

<i>Nome</i>		<i>Stato iniziale:MAN</i>		
<i>template:MnnTEMPL</i>				
<i>Conf.</i>	<i>Conf.</i>	<i>Approve</i>	<i>Promote</i>	<i>Clear</i>
<i>partenza</i>	<i>Arrivo</i>		<i>to host</i> ¹⁷	<i>approvals</i>
MAN	ESER		Y	N

13. definire la nuova area e gli standard di nomenclatura degli oggetti che le appartengono come indicato alla sez. 6.1.1

6.1.1. La nuova area e gli standard di nomenclatura degli oggetti nella base dati DB2

Una nuova area deve essere definita anche nella base dati DB2 del prodotto, in termini di entità area e di standard di nomenclatura che consentono l'attribuzione degli oggetti al prodotto.

E' necessario a tal fine inserire un'occorrenza nella tabella D01.TDLCARE interattivamente attraverso SPUI o QMF o batch tramite DSNTIAD sul sottosistema DB2 di sviluppo, prendendo come esempio il membro H0T4.LCMSW.DDL(INLCARE) di precaricamento della suddetta tabella.

In maniera analoga è necessario definire gli standard di nomenclatura per SOURCE, DGIGRP, DGIPNL, COPY, INCLUDE, DCLGEN,

¹⁶ Come già anticipato alla sez. 6.6.1 a proposito delle export host library di COLL, si è indicato PROMOTE Y in questo migration path per poter sfruttare l'exit di promote a livello di package per inserire elaborazioni specifiche di installazione.

¹⁷ Come già anticipato alla sez. 6.6.1 a proposito delle export host library di COLL, si è indicato PROMOTE Y in questo migration path per poter sfruttare l'exit di promote a livello di package per inserire elaborazioni specifiche di installazione.

PSBSRC ed inserirli nella tabella D01.TDLCSTD, prendendo come esempio il membro H0T4.LCMSW.DDL(INLCSTD) di precaricamento della suddetta tabella.

6.2. DEFINIZIONE DI UNA NUOVA TIPOLOGIA DI OGGETTI

La definizione di una nuova tipologia (type) di oggetti è un'attività che non è incanalabile in specifiche procedure operative, in quanto può variare molto in dipendenza dalla specificità della tipologia (di tipo COPY, piuttosto che di tipo source).

Diamo quindi un elenco di punti che devono essere studiati ed eventualmente predisposti nel caso in cui sorga tale necessità.

1. devono essere studiate e definite le relazioni di tipo impact, reference e cross-reference nelle quali la presente tipologia è coinvolta come elemento primario (chi può richiamare) o come elemento secondario (da quale delle tipologie attualmente esistenti può essere chiamata);
2. devono essere studiate ed eventualmente definite le relazioni di tipo build e se presenti quale tipologia target originano;
3. nel caso in cui sia definita una nuova relazione di tipo build, occorre disegnare e predisporre la build procedure in SVIL MAN e COLL per quanto riguarda la fase di 'compilazione propriamente detta;
4. vanno valutati gli impatti sui trasferimenti in termini di operazioni che tale tipologia richiede siano espletate durante i trasferimenti e disegnare allo scopo sia la build procedure in COLL ed ESER specifica per la tipologia, sia le modifiche alla parte comune dei trasferimenti (skeleton LCS0ETRA e LCS0CTRA) che implementano la fase comune a valle della fase di build;
5. deve essere disegnata e realizzata la parametrizzazione delle build procedure a livello di tipologia e configurazione (skeleton LCSxyPAR), a livello di configurazione (skeleton LCS0yPAR) a livello di area, configurazione e tipologia (parameter set a livello di system);
6. deve essere valutata la possibilità di utilizzare uno dei due scanner attualmente esistenti ovvero la possibilità di definire uno scanner nuovo, da associarle;
7. devono essere individuate e predisposte correttamente export, build e target library per la tipologia a livello di area e/o ambiente;
8. si deve definire la nuova tipologia della base dati DB2 a livello di tabella tipologie TDLCTIP, di tabella sottotipi TDLCSTP prendendo spunto dai membri INDLCTIP e INDLCTP presenti in H0T4.LCMSW.DDL per il caricamento iniziale delle suddette tabelle,
9. devono essere individuati gli standard di nomenclatura di tipologia-area ed inseriti nella tabella TDLCSTD analogamente a quanto descritto alla sezione 6.1.1

6.3. GESTIONE DEI SOTTOTIPI NELL'AMBITO DEI SOURCE

Per consentire di gestire anche in ambito ASG/LCM differenti tipologie di programmi, in modo da poterne differenziare le modalità di compilazione, sono stati definiti i sottotipi associati al programma.

Il sottotipo attribuito ad un programma è un attributo di max tre caratteri, memorizzati però in tre colonne distinte della base dati (vedi tabelle TDLCSSTP e TDLCOGG in Allegato 1):

<i>codice_at</i>	di un carattere, coincide attualmente con la tipologia GIS;
<i>livello</i>	può assumere i valori M S o space a seconda che si tratti di un main di una subroutine o di una tipologia per la quale non ha senso specificare tale ulteriore distinzione (per esempio è il caso dei batch assembler)
<i>sotto_trattam</i>	di un carattere attualmente valorizzato solo a I o a space, serve a gestire, laddove necessario ulteriori differenziazioni di trattamento nell'ambito della stessa tipologia (attualmente è valorizzato ad I per le tipologie main SGC per consentire la gestione corretta degli inquiry a scorrere)

Il sottotipo è definito nella tabella TDLCSSTP unitamente alle seguenti informazioni necessarie alla sua corretta gestione:

<i>regola_main</i>	se codificata, può assumere i valori: INPUT, quando per il sottotipo è previsto che il main CICS da associare in tabella PCT sia fornito dall'utente (è il caso delle routine C/S di tipo E); NAME quando per il sottotipo è previsto che il main da specificare in PCT sia l'oggetto stesso (per esempio i processi VM); CXPBTSCH o qualunque altra stringa differente da space, NAME INPUT, indica il nome esplicito del main da associare al transid in PCT.
<i>regola_psb</i>	se codificata, può assumere i valori: INPUT, quando per il sottotipo è previsto che il PSB sia fornito in input all'atto della creazione del programma (batch DL/I); NAME quando per il sottotipo è previsto che il PSB associato sia omonimo rispetto all'oggetto stesso (tp DL/I).

Per ciascun sottotipo, composto come sopra descritto, devono essere individuati e definiti in base dati i trattamenti previsti all'atto del processo di build.

I trattamenti attualmente previsti in base dati sono quelli elencati nella tabella di Fig. 6.3.a

NOME	DESCR
TRATT	TRATT
WALIAS	TRATTAMENTO ALIAS INQUIRY A SCORRERE
WASM	IEV90 PER PGM ASSEMBLER
WCBLTDLI	STUB CBLTDLI PER MAIN BATCH DLI
WCOBBTCH	COMP.COB II CON PARM NORENT OFFSET
WCOBBTC1	COMP.COB II CON PARM RENT OFFSET
WCOBBTC2	COMP.COB II CON PARM NORENT LIST

WCOBDFLT	COMP.COB II CON PARM DI DEFAULT
WDFSLIIN	STUB DFSLIIN PER BATCH DLI DB2
WLKDAM24	TRATTAMENTO AMODE(24) NEL LINK EDIT
WMETRWSP	STEP METACOBOL CON MACROSET WASP E TRACE
WMETWASP	STEP METACOBOL CON MACROSET WASP
WMETWPCE	STEP METACOBOL CON MACROSET WASP E YPCE10
WRCOPY	PRELABORAZIONE DEGLI STATEMENT DI RCOPY
WTRACXAS	TRADUZIONE CICS PER PROGRAMMI ASSEMBLER
WTRACXCB	TRADUZIONE CICS PER PROGRAMMI COBOL II
WTRADLI	TRADUZIONE DL/I E STANDARD PER PGM.DL/I
WTRASGC	TRADUZIONE SGC
WTRASTD	TRADUZIONE YM1001 PER MAIN SGC NON DL/I
WTRDB2CB	PRECOMPILAZIONE DB2 PER PROGRAMMI COBOL II

Fig.6.3.a

Come si può evidenziare dalla figura, i differenti trattamenti relativi alla traduzione metacobol, previsti per consentire la differenziazione dei macroset, iniziano e **devono** iniziare tutti con la stringa WMET, così come deve essere per i differenti trattamenti legati ai differenti parametri di compilazione COBOL che iniziano e **devono** iniziare tutti con la stringa WCOB. Nella tabella TDLCSUB, a ciascun sottotipo sono associati i trattamenti previsti per lo stesso (tante occorrenze di tabella quanti sono i trattamenti previsti).

Nel corso della compilazione dei programmi (o meglio nel corso della build dei load) sono richiamate all'inizio del job di compilazione le procedure di gestione dei trattamenti ed in particolare

è richiamata:

WSCARTRT	richiamata una volta, consente di scaricare i trattamenti previsti per il programma oggetto della build;
WTESTTRT	richiamata tante volte quanti sono i trattamenti previsti nella base dati del prodotto (il nome dello step che richiama la procedura per il trattamento Wxxxxxxx si chiama Txxxxxxx), verifica se per l'oggetto in esame il trattamento fornito in input è previsto oppure no. Se è previsto lo step st020 della procedura è impostato a 0, se non è previsto è valorizzato a 4.

Nel seguito del job di compilazione, prima di eseguire step che sono specifici di un particolare trattamento Wxxxxxxx, viene effettuato un test sul codice di ritorno dello step Txxxxxxx.ST020 ed il trattamento è eseguito solo se questo è 0.

Poiché quindi la sequenza di esecuzione dei trattamenti è determinata runtime, ogni trattamento prende in input sempre il medesimo sequenziale temporaneo e, al suo completamento, copia il suo output (per esempio l'oggetto sottoposto a precompilazione DB2) su tale temporaneo di input per predisporlo per il trattamento successivo.

6.4 GESTIONE DELL'EXCLUSION LIST

L'exclusion list, gestita tramite l'opzione CCC.3.1.3, fornisce la lista degli oggetti che non devono concorrere al processo di Analyze svolto da una particolare area.

Esiste, infatti, un'exclusion list per ogni area, configurata nella fase di attivazione del prodotto, ma destinata ad essere movimentata a secondo delle esigenze.

Possibili cause di modifica dell'exclusion list di un'area sono le seguenti:

- utilizzo di nuovo software appartenente ad altre aree per le quali non è impostato il 'cross system';
- utilizzo di software sviluppato al di fuori di ASG/LCM.

Occorre mettere in evidenza il fatto che una modifica all'exclusion list pregiudica il corretto funzionamento del meccanismo dell'analyze ottimizzato riportato al paragrafo 5.7; pertanto ogni qualvolta essa viene modificata, occorre effettuare il refresh di tutte le quattro work aree di configurazione (SVIL, MAN, COLL, ESER) relative all'area.

7. VARIAZIONI DI GESTIONE ORDINARIA DEGLI AMBIENTI GESTITI DAI PRODOTTI PROPRIETARI

In questa sezione sono descritte per differenza, rispetto alla situazione precedente l'introduzione del prodotto ASG/LCM, le modalità di gestione dei prodotti proprietari modificati per la compatibilità con il nuovo ambiente.

I prodotti proprietari che comportano variazioni di gestione sono:

- Gestione Oggetti DL/I,
- Funzioni Accessorie Applicative.

Quest'ultimo è un nuovo instradamento che sostituisce quello di GIS. Attraverso esso si attivano le funzioni accessorie precedentemente contenute in GIS.

Tale instradamento è reso disponibile nella procedura di logon ALOGCCC mediante l'esecuzione dei seguenti passi:

- concatenazione, in SYSPROC, della libreria H0T2.FAA.CLIST prima di N0T2.ISPFAPPL.CLIST,
- concatenazione, in ISPLLIB, di H0T2.FAA.PANELS prima di N0T2.ISPFAPPL.PANELS,
- concatenazione, in ISPLLIB, di H0T2.FAA.LOAD prima di N0T2.ISPFAPPL.LOAD,
- concatenazione, in ISPTLIB, di H0T2.FAA.TABS prima di N0T2.ISPFAPPL.TABS,
- concatenazione, in ISPSLIB, di H0T2.FAA.SKELS prima di N0T2.ISPFAPPL.SKELS,
- introduzione, nel menu 'Prodotti' corrispondente alla procedura di logon ALOGCCC, dell'attivazione di 'Funzioni Accessorie Applicative' nel seguente modo:
inserimento dell'opzione

```
F .APPL .Funzioni accessorie applicative
```

cui corrisponde, nell'area)PROC, l'entrata:

```
F, ' PGM(HGXXX00) PARM(&ZALTTR) NEWAPPL(HGIS) NEWPOOL  
NOCHECK '
```

```
dove: &ZALTTR=TRUNC(&ZCMD, ' . ' ) ,  
&ZALTTR= . TRAIL .
```

7.1. GESTIONE OGGETTI DL/I

Questo prodotto ha un'interfaccia ISPF che instrada a diverse funzioni sia su PLT che su Esercizio.

Su PLT sono state variate le seguenti opzioni:

- opz. 1: Trasferimento PSB,

- opz. E: Trasferimento PSB (anche EDA),
- opz. 6: Trasferimento SCHEMI COBOL DL/I,
- opz. 7: Allineamento SCHEMI COBOL DL/I (amb. C/R),
- opz. 8: Generazione SCHEMI COBOL DL/I.

Su Esercizio le opzioni che hanno subito modifiche sono:

- opz. 1: Trasferimento PSB,
- opz. E: Trasferimento PSB (anche EDA),
- opz. 3: Trasferimento SCHEMI COBOL DL/I.

Sul sistema di PLT è possibile utilizzare il prodotto gestione Oggetti DL/I tramite la procedura di logon ACCCTCON mentre sul sistema di esercizio la procedura allestita è ACCCY.

L'instradamento al prodotto sul sistema PLT viene fatto tramite l'opzione 9.A.D.2 mentre sul sistema di Esercizio l'opzione è 9.D.

Le opzioni relative ai PSB adesso permettono di gestire solo i PSB di caricamento ed i PSB EDA. Tutti gli altri tipi di PSB sono memorizzati nel database di ASG/LCM e trasferiti attraverso tale prodotto.

I PSB EDA e di caricamento, invece, continuano ad essere memorizzati nei data set che li contenevano anche nella vecchia configurazione di ambiente.

Si noti che, per tali oggetti, gli ambienti gestiti agli effetti del trasferimento non sono cambiati. Infatti, i trasferimenti verso l'ambiente di collaudo possono essere effettuati con le stesse modalità (ed utility) impiegate prima dell'introduzione di ASG/LCM.

Per quanto riguarda le opzioni relative agli Schemi COBOL DL/I, non esiste nessuna variazione da segnalare per quanto concerne l'interfaccia utente. È, invece, importante sottolineare che i data set contenenti gli Schemi COBOL sono, nella presente configurazione, variati per composizione. Infatti, oltre agli Schemi propriamente detti essi contengono le COPY dei dati di intersezione (precedentemente memorizzate sui Master File delle COPY).

Tale variazione nella composizione dei data set si ripercuote, naturalmente, sulle operazioni effettuate dai job batch sottomessi mediante le opzioni ISPF.

Anche nel caso degli Schemi, occorre sottolineare che non sono cambiati gli ambienti gestiti agli effetti dei trasferimenti.

7.2. FUNZIONI ACCESSORIE APPLICATIVE

Tale instradamento alle funzioni di GIS non sostituite da ASG/LCM deve essere disponibile solamente a chiavi di 'vecchio tipo' ovvero aventi l'iniziale coincidente con il carattere identificativo dell'area.

Ciò è indispensabile per garantire i controlli sull'area e l'impostazione dell'ambiente di lavoro opportuno.

La compatibilità con la diversa granularità (in base all'area oltre che all'ambiente operativo) delle librerie contenenti alcuni oggetti (es. DCLGEN,

DBRM etc.) è gestita dal nuovo prodotto sfruttando l'informazione 'iniziale della chiave di logon' dell'utente applicativo.

L'associazione: 'iniziale della chiave di logon' → 'librerie di competenza' viene effettuata in base alla tabella seguente.

Identificativo System	Denominazione area	Identificativo area
BIL	Bilancio	V
CE	Comunità Europea	I
PAT	Conto del Patrimonio	O
CP	Conti Pubblici	J
ENT	Entrate	E, U
SPE	Spese	S,Z
BDC	Banca Dati Conoscitiva (BDC)	R
PER	Personale	M,Q
LES	Liquidazione Enti Disciolti (IGED)	JL
MON	Monitoraggio	W,X
CDC	Corte dei Conti	A, B
PI	Pubblico Impiego	JH
GR	GRURAG	G
AP	IGESPES	T
SSY	Tecnica	C, D, H, K, N, Y

Il nuovo prodotto non usa più, infatti, la tabella HGISLIBN che definiva le librerie utente diversificate in base all'area.

Un'altra tabella appartenente alla base dati di GIS ma non a quella di 'Funzioni Accessorie Applicative' è HGISUADN.

Continuano, invece ad essere utilizzate nella base dati del prodotto:

- HGISLIPN, contenente le variabili di ambiente di diverso uso,
- HGISFUNZ, che descrive, per ogni funzione possibile dell'instradamento il suo stato (se presente o assente e, se presente, se attiva o inibita).

La gestione ordinaria di tali tabelle è a carico del componente gestionale del prodotto (GISGES).

Tale componente si attiva dalla procedura di logon ALOGCCC con il comando:

TSO GISGES.

La versione di GISGES disponibile nel nuovo ambiente è, sostanzialmente, una semplificazione del componente omonimo presente nel vecchio ambiente. Sono state conservate, infatti, solo le funzioni di gestione delle tabelle HGISLIPN e HGISFUNZ. Questo comporta che, nel menu prospettato attraverso il comando 'GISGES', siano selezionabili solo le opzioni che instradano ai menu 3.0 e 4.0.

BIBLIOGRAFIA

- [1] **R04-412-XXX-0001**
 27 Dicembre 2000
 Versione 1

Contratto per la "Realizzazione degli ambienti di sviluppo, manutenzione e collaudo per la gestione del ciclo di vita del software in condizioni di multiprogetto sul sistema centrale MVS del Sistema Informativo integrato del Dipartimento della Ragioneria Generale dello Stato e della Corte dei Conti" - Manuale d'uso

- [2]

PLATINUM - ASG/Life Cycle Manager - Installation and Administration Guide - version 3.0

- [3]

PLATINUM - ASG/Life Cycle Manager - Reference Manual - Volume 1 e Volume 2

- [4] **R04-301-XXX-0002**
 1 Febbraio 2001
 Versione 2

Contratto per la "Realizzazione degli ambienti di sviluppo, manutenzione e collaudo per la gestione del ciclo di vita del software in condizioni di multiprogetto sul sistema centrale MVS del Sistema Informativo integrato del Dipartimento della Ragioneria Generale dello Stato e della Corte dei Conti" - Manutenzione dei prodotti proprietari e nuove funzioni da implementare - Specifica di disegno

GLOSSARIO

Ambiente • Insieme di librerie e sottosistemi all'interno dei quali è modificata, eventualmente compilata ed eseguita un'applicazione in una ben specifica fase del C.V.S. (sviluppo, collaudo, etc.). In questo manuale ha il medesimo senso di configurazione o configuration

Analyze • Processo che consente di popolare le relazioni definite a livello di type con le occorrenze relative ai singoli item dell'applicazione

Applicazione • Vedi descrizione relativa a system

Area operativa • Area di applicazione nell'ambito del Sistema Informativo integrato della Ragioneria Generale dello Stato e della Corte dei Conti; nel documento questo termine o, più semplicemente, area è utilizzato come sinonimo di system o di sistema.

Baseline • E' una configurazione in un sistema considerata come il punto di partenza o di chiusura del ciclo di vita di sviluppo e manutenzione.

Build (relazione di) • E' una relazione tra un configuration item definito target (per esempio un load) e i configuration item primari a partire dai quali è costruito (per esempio un programma sorgente)

Build Procedure (o procedura di Build) • E' una procedura utente che fornisce l'insieme delle istruzioni necessarie a costruire un target a partire dai suoi componenti primari

Complete Import • funzione del prodotto che salva nel database CCC una versione dell'oggetto che è considerata definitiva. Viene creata una nuova versione del membro nel CCC e il suo stato è riportato a Normal. E' quindi possibile per un altro utente esportare nuovamente l'oggetto per update.

Configuration • Struttura nel database CCC che contiene tutti i componenti correlati che costituiscono una versione di un'applicazione.

Configuration item (CI) • E' il componente del CCC nel quale sono effettivamente memorizzate le informazioni e oggetto di modifica da parte degli sviluppatori. Sono assimilabili ai membri di libreria MVS e sono univocamente identificati dalla coppia type.member nell'ambito della configurazione. I termini Configuration item, item, membro possono essere usati come sinonimi.

Configurazione • Vedi descrizione relativa a Configuration

Controlled Build (o Build) • Funzione che, prendendo in input i risultati di un processo di analyze genera uno specifico target item

Cross Reference (relazione di) • relazione fra due configuration item nella quale un CI detto primario chiama, referencia o comunque usa un altro CI detto secondario. Il componente secondario non è richiesto per costruire il target. Quando si modifica un componente secondario, il primario non diventa obsoleto (esempio chiamata dinamica ad un programma)

Cross system • un sistema si definisce cross-system quando tipologie di oggetti che gli appartengono sono referenziate come componenti secondari in relazione con tipologie di oggetti appartenenti ad altri sistemi

Database • il livello più alto della gerarchia del CCC che contiene al suo interno tutto il complesso delle informazioni gestite dal CCC

Dependency work area • strumento utilizzato dal prodotto per raggruppare e memorizzare le informazioni relative alle dipendenze fra CI, che consente di fornire un ambiente operativo comune ad utenti che operano nella medesima configurazione o nell'ambito dello stesso pacchetto

Export • funzione del prodotto che estrae gli oggetti dal database CCC per portarli sugli host library dell'MVS per poterli modificare o leggere.

EXPORTED status • stato in cui viene posto un configuration item che è stato esportato per update a partire da una data configurazione. In questo stato nessun utente può nuovamente esportarlo per update nella medesima configurazione. Lo stato di exported per un oggetto è mantenuto finché il proprietario non ne effettua l'import con l'opzione complete.

Impact (relazione di) • relazione fra due configuration item nella quale un CI detto primario chiama un altro CI detto secondario. Quando si modifica un componente secondario, il primario diventa obsoleto (esempio Include di una DCLGEN).

Import • funzione del prodotto che copia/muove oggetti dalle host library dell'MVS nel database CCC.

Intermediate Import • funzione del prodotto che salva nel database CCC una versione dell'oggetto che non è considerata definitiva (sulla quale non è stata ancora portata a compimento l'attività di modifica). Viene creata una nuova versione del membro nel CCC, ma il suo stato rimane exported.

Member • componenti del data base CCC nei quali sono realmente memorizzati i dati. I membri sono l'unità modificata nell'ambito dell'attività di sviluppo e manutenzione e ad essi ci si riferisce con il termine di Configuration Item (vedi definizione). system

Migrate • funzione che copia tutte le modifiche associate ad un pacchetto da una configurazione ad un'altra, implementando in tal modo il passaggio da una fase del CVS alla fase successiva.

Migration path • il percorso previsto per un pacchetto nel ciclo di vita del software

NORMAL status • stato in cui si trova un configuration item quando non è sottoposto ad alcuna attività di modifica.

Pacchetto di modifiche o pacchetto • vedi package

Package • insieme di componenti software (CI) associati ad un progetto di sviluppo o ad un intervento di manutenzione. E' definito dall'intersezione fra la lista di item e la lista di change name assegnati al pacchetto.

Reference (relazione di) • relazione fra due configuration item nella quale un CI detto primario chiama, referencia o comunque usa un altro CI detto secondario. Entrambi i componenti della relazione sono necessari per costruire il target. Quando si modifica un componente secondario, il primario non diventa obsoleto (esempio chiamata statica ad un programma).

Sistema • Vedi descrizione relativa a system

System • Struttura nel database CCC che solitamente racchiude una applicazione nella sua completezza. Il sistema si articola in configurazioni, ciascuna delle quali rappresenta una versione dell'applicazione

Tipo o tipologia • Vedi descrizione relativa a type

Type • Struttura del database CCC utilizzata per raggruppare elementi simili, analoga al livello TYPE nella struttura delle librerie MVS.

Work area • Vedi dependency work area

ALLEGATI

Allegato 1	La base dati dell'LCMSW v. 1.0
Allegato 2	Modulistica

ALLEGATO 1

LA BASE DATI DELL' LCMSW V.1.0

Elenco degli attributi dell'oggetto TDLCSTD (T1)						
Nome Colonna	Tipo dato	Null	Valori ammessi	Prim. key	For. Key	Tab. design.
tipo_oggetto	char(8)	NN		P(1)	P1	TDLCSTIP
nome_area	char(8)	NN		P(2)	P2	TDLCARE
progr_std	smallint	NN		P(3)		
standard	varchar(8)	NN				

Elenco degli attributi dell'oggetto TDLCARE (T2)						
Nome Colonna	Tipo dato	Null	Valori ammessi	Prim. key	For. Key	Tab. design.
nome_area	char(8)	NN		P(1)		
codice_area	char(2)	NN	numerico			
descr_area	char(30)	ND				

Elenco degli attributi dell'oggetto TDLCSTIP (T3)						
Nome Colonna	Tipo dato	Null	Valori ammessi	Prim. key	For. Key	Tab. design.
tipo_oggetto	char (8)	NN		P(1)		
descr_tipo	char (30)	NN				

Elenco degli attributi dell'oggetto TDLCSTP (T4)						
Nome Colonna	Tipo dato	Null	Valori ammessi	Prim. key	For. Key	Tab. design.
tipo_oggetto	char(8)	NN		P(1)	P1	TDLCSTIP
codice_at	char(1)	NN	blank, alfanum	P(2)		
livello	char(1)	NN	blank, M, S	P(3)		
sotto_trattam ¹⁸	char(1)	ND	blank, I,A	P(4)		
regola_main	char(8)	ND	blank, NAME, INPUT, CXPBTSC			
regola_PSB	char(8)	ND	blank,NAME, INPUT			
descr_sottotipo	char(60)	NN				

Elenco degli attributi dell'oggetto TDLCSUB (T5)						
Nome Colonna	Tipo dato	Null	Valori ammessi	Prim. key	For. Key	Tab. design.
tipo_oggetto	char(8)	NN		P(1)	P2	TDLCSTP
codice_at	char(1)	ND		P(2)	P2	TDLCSTP

¹⁸ Il campo sotto-trattam è utilizzato per gestire casi particolari nell'ambito delle attuali tipologie GIS, che richiedano trattamenti specifici. Il valore I serve per indicare gli "Inquiry a scorrere" nell'ambito delle tipologie Main SGC, mentre il valore "A" serve per gestire i programmi batch dell'area Spese con l'interfaccia "APIEDI", la cui compilazione viene attualmente implementata al di fuori di GIS.

livello	char(1)	ND		P(3)	P2	TDLCSSTP
sotto-trattam	char(1)	ND		P(4)	P2	TDLCSSTP
nome_area ¹⁹	char(8)	ND	blank o nome_ area	P(5)	P1	TDLCSARE
nome_tratt	char(8)	NN		P(6)	P3	TDLCSRTT

¹⁹ Le occorrenze della relazione Subisce possono o essere, per un dato sottotipo, generalizzate per tutte le aree (nome_area a blank) o essere, per un dato sottotipo, tutte specifiche di area.

Elenco degli attributi dell'oggetto TDLCOGG (T6)						
Nome Colonna	Tipo dato	Null	Valori ammessi	Prim. key	For. Key	Tab. design.
tipo_oggetto	char(8)	NN		P(1)	P1,R2	TDLCTIP, TDLCSSTP
nome_oggetto	char(8)	NN		P(2)		
orig_oggetto	char(1)	NN	S ,P	P(3)		
codice_at	char(1)	ND			R2	TDLCSSTP
livello	char(1)	ND			R2	TDLCSSTP
sotto_trattam	char(1)	ND			R2	TDLCSSTP
nome_area	char(8)	NN			R3	TDLCSARE
nome_main	char(8)	ND				
nome_PSB	char(8)	ND			A4	TDLCOGG

Elenco degli attributi dell'oggetto TDLCTRT (T7)						
Nome Colonna	Tipo dato	Null	Valori ammessi	Prim. key	For. Key	Tab. design.
nome_tratt	char(8)	NN		P(1)		
descr_tratt	char(70)	NN				

Elenco degli attributi dell'oggetto TDLCLGE (T8)						
Nome Colonna	Tipo dato	Null	Valori ammessi	Prim. key	For. Key	Tab. design.
tipo_evento	char(8)	NN	EXPORT MIGRATE			
tipo_oggetto	char(8)	NN				
nome_oggetto	char(8)	NN				
nome_area	char(8)	NN				
data_evento	data	ND				
ora_evento	time	ND				
from_config	char(8)	NN				
to_config	char(8)	ND	solo per MIGRATE			
package	char(8)	ND	solo per MIGRATE			
esito_build	char(30)	ND	solo per MIGRATE			
utente	char(8)	ND	solo per EXPORT			
exp_host_library	char(44)	ND	solo per EXPORT			

Elenco degli attributi dell'oggetto TDLCLGT (T9)						
Nome Colonna	Tipo dato	Null	Valori ammessi	Prim. key	For. Key	Tab. design.
package	char(8)	NN				
nome_area	char(8)	NN				
from_config	char(8)	NN				
data_ins	char(8)	NN	yyyymmdd			
ora_ins	char(6)	NN	hhmmss			
stato_trasf	char(1)	NN	numerico			
num_job_subm	char(5)	NN	numerico			
num_job_ended	char(5)	NN	numerico			

Elenco degli attributi dell'oggetto LCALOG (T10)						
Nome Colonna	Tipo dato	Null	Valori ammessi	Prim. key	For. Key	Tab. design.
package	X(8)					
nome_area	X(8)					
from_config	X(8)					
data_ins	X(8)		yyyymmdd			
ora_ins	X(6)		hhmmss			
stato_trasf ²⁰	X(1)		1,2,3,4,5,6,7,8			
num_job_subm	9(5)					
num_job_ended	9(5)					

²⁰ Lo stato dei trasferimenti può assumere i seguenti valori: 1, quando il record viene inserito nel log all'atto della richiesta di MIGRATE sotto ASG/LCM; 2, quando LCPLGSCH sottomette il job capostipite; 3, quando il job capostipite della Migrate parte; 4, quando finisce la fase su PLT1 a valle del completamento della fase di BUILD; 5, quando finisce la fase sulla partizione PLT2, per i trasferimenti verso COLL, o sulla partizione di esercizio, per i trasferimenti verso ESER.

ALLEGATO 2

MODULISTICA

RICHIESTA DI CANCELLAZIONE OGGETTI DAGLI AMBIENTI ASG/LCM

Nome richiedente:

Società:

Matricola:

Area di appartenenza degli oggetti:

TIPOLOGIA OGGETTO	NOME OGGETTO	AMBIENTE DA CUI CANCELLARE L'OGGETTO

RICHIESTA DI MODIFICA ALLA SCHEDULAZIONE DEI TRASFERIMENTI

Nome richiedente:

Società:

Matricola:

Tipo di richiesta:

1. Trasferimenti urgenti

☐

**2. Cancellazione
trasferimenti**

☐

**3. Modifiche alla
schedulazione dei
trasferimenti**

☐☐☐

Solo trasferimento pacchetti elencati

**Solo trasferimento pacchetti appartenenti
ad aree elencate**

**Solo trasferimento pacchetti NON
appartenenti ad aree elencate**

Ambiente di provenienza:

NOME PACCHETTO

NOME AREA
