Architettura programmazione Sme.UP

Nel seguente documento verrà esposta l'architettura generale della programmazione Sme.UP.
Tale documento è indispensabile sia per chi sviluppa Sme.UP che per chi scrive programmi personali o personalizzati.

Non seguire le indicazione inserite in questo documento può portare a problemi nel funzionamento o a comportamenti imprevisti.

Nella release V3R1 di Sme.UP è stata fatta una profonda revisione dell'architettura, sopratutto per quanto concerne l'uso dei gruppi di attivazione. E' molto importante quindi leggere con attenzione queste indicazioni.

Linee guida della struttura dei programmi

Con la V3R1 è stato consolidato il disegno architetturale della programmazione Sme.UP.

Queste sono le linee guida che sono state seguite:

1. Nessun programma deve essere eseguito nel gruppo di attivazione di default (comunemente chiamato DAG - Default Active Group)
2. I programmi con gruppo di attivazione *NEW lo sono per scelta e non bisogna abusarne
3. Gli RCLACTGRP vanno fatti in modo mirato e specifico, non va mai fatto un RCLACTGRP ACTGRP(*ELIGIBLE) (se non al cambio ambiente)
4. Va garantita la certezza che al cambio ambiente tutti i file precedentemente aperti vengano chiusi
5. Esistono gruppi di attivazione che non si chiudono mai (se non al cambio ambiente) e altri che invece possono essere chiusi
6. I programmi che si "chiudono in RT" devono essere tali. Ossia devono essere in grado di gestire la non esecuzione della £INIZI senza basarsi ad esempio sul fatto che essi siano richiamati da un programma che si "chiude in LR".
7. Gli override che vengono impostati hanno normalmente lo scope di "livello di chiamata". I casi in cui hanno validità nel gruppo di attivazione sono le eccezioni
8. Gli attributi di job riguardanti la localizzazione (ad esempio il separatore decimale) devono essere corretti "a livello di job", altrimenti tutti i comandi o le bif che si basano su questi valori (come il %EDITC) danno risultati non corretti
9. Se gli errori vengono intercettati, vanno gestiti nello specifico. Non viene utilizzata nessuna struttura "generalizzata" di intercettazione degli errori
10. E' necessario essere indipendenti dai default di sistema dei comandi

L'adozione di queste linee guida ha portato ad alcuni cambiamenti importanti nello stile di programmazione da adottare.

Gruppi di attivazione e impostazioni di localizzazione linguistica

Con la V3R1 è stato consolidato il disegno dell'uso dei gruppi di attivazione.
I programmi Sme.UP di default vengono compilati con gruppo di attivazione *CALLER. Quindi per non essere eseguiti MAI nel gruppo di attivazione di default, nella lista richiami deve essere sempre presente "in testa" un programma che viene eseguito in un gruppo di attivazione non default.
Inoltre, come detto, i job devono avere impostati correttamente i valori relativi alla localizzazione.
Quindi tutti i job in cui vengono eseguiti "programmi Sme.up" devo avere un programma di partenza particolare.
Questo programma ha essenzialmente due compiti: "impostare" il gruppo di attivazione non default e impostare gli attributi di localizzazione corretti.

Per quanto riguarda la parte interattiva non c'è nessun problema, in quanto il B£UT50 e gli altri programmi collegati si occupano di queste impostazioni.
Discorso analogo per Looc.UP.
Anche la sottomissione in batch tramite £GPE effettua tutte le inizializzazioni in modo corretto (l'esecuzione "interattiva" della £GPE esegue comunque il comando in un gruppo di attivazione *NEW).

Quali situazioni rimangono da controllare?

1. SBMJOB
2. Azioni eseguite non tramite menù Sme.UP (ad esempio perché lanciate da modulo base)
3. Lavori schedulati

Per risolvere tutte queste situazioni è stato creato un comando, il B£QQ02, che si occupa di eseguire tutte le impostazioni richieste e poi di eseguire il comando che gli viene passato in input.
Per i lavori schedulati c'è anche la possibilità (per altro consigliata) di utilizzare il programma B£QQ01 che oltre ad eseguire le impostazioni richieste, consente anche di eseguire il lavoro con una lista librerie particolare.
Nel seguito del documento verrà approfondito l'argomento e risulterà quindi chiara l'estrema importanza del fatto che tutti i programmi/comandi Sme.UP che vengono sottomessi da un qualunque job (sia Sme.UP che no) od eseguiti da un job Non-Sme.UP debbano essere eseguiti "tramite" il B£QQ02.

Gestione degli errori

La filosofia seguita è "se non so precisamente come gestire un errore, meglio non gestirlo del tutto".
Questo significa che la tecnica di monitorare con indicatori (o con costrutto MONITOR-ENDMON) le CALL o le operazioni sui file vanno fatte solo se poi si è in grado di gestire correttamente l'errore.
Monitorare un'operazione su un file o la CALL di un programma implica due possibili scenari:

1. per ogni errore monitorato so quale azione correttiva intraprendere
2. l'operazione monitorata è "facoltativa", la sua esecuzione non è vincolante

La seconda possibilità è decisamente remota... se faccio qualcosa è perché mi interessa farlo.
Quindi l'unica possibilità è che si sappia con precisione cosa fare al verificarsi di un determinato errore.

L'indicatore come strumento di monitor è sempre sconsigliato, perché monitora qualunque tipo di errore. Il costrutto MONITOR-ENDMON è più flessibile perché consente di monitorare solo alcuni errori.

Monitorare una CALL con un indicatore ad esempio è sconsigliatissimo, perché QUALUNQUE errore possa presentarsi nel programma richiamato, viene "nascosto" da questo indicatore.
Spesso l'indicatore sulle CALL viene messo come "controllo di programma inesistente". Il problema è che se il programma esiste e va in errore, viene trattato alla stessa stregua di programma inesistente.
In questo caso la soluzione migliore è usare il costrutto MONITOR ed andare a gestire l'errore "programma non trovato", lasciando non gestito ogni altro errore.
Per gestire situazioni come questa, è consigliato l'uso della parola chiave £MON nei commenti, per il cui utilizzo di rimanda alla documentazione specifica.

a standard la gestione "generalizzata" degli errori è stata eliminata., Nello specifico:

1. è stata eliminata la *PSSR dalla £INIZI
2. è stato eliminato l'indicatore dalla CALL delle /COPY
3. è stato eliminato l'indicatore dalle CALL delle exit sostituendo con "£MON" per il controllo delle exit inesistenti

La *PSSR può essere reintrodotta nei programmi in cui la si vuole esplicitamente. Per farlo è necessario definire a inizio programma la direttiva
/DEFINE SI_PSSR

Inoltre l'indicatore sulle /COPY può essere "reintrodotto" semplicemente andando ad agire sul flag ££B£2J della tabella B£2.
In realtà nella /COPY c'è una struttura tipo:

```
IF      ££B£2J = '1'
CALL    'XXXXXX'                    37
ELSE
CALL    'XXXXXX'
ENDIF
```

Attenzioni di programmazione

Per poter restare all'interno delle linee guida precedentemente citate è necessario, in fase di programmazione, seguire alcuni accorgimenti.

Gruppi di attivazione non default

Il fatto che tutti i programmi Sme.UP vengono eseguiti in un gruppo di attivazione diverso da quello default, porta alcuni cambiamenti (rispetto al passato) nell'esecuzione del programma stesso.
Tali differenze vanno tenute in considerazione per scrivere un programma corretto.
Le differenze riguardano:

1. Scope di Override
2. Chiusura programmi in LR/RT
3. RCLRSC

Scope di override

I comandi OVRDBF, OVRPRTF e DLTOVR hanno un comportamento di default diverso, per quanto riguarda il loro ambito di attivazione, a seconda che vengano eseguiti nel gruppo di attvazione
Il parametro che guida questo comportamento è OVRSCOPE per i comandi OVRDBF e OVRPRTF e il parametro LVL per il comando DLTOVR.
Il valore di default è sempre *DFTACTGRP, che fa assumere il seguente comportamento:
Comandi OVRDBF e OVRPRTF

1. se eseguito nel gruppo di default, coincide con il valore *CALLLVL, vale a dire che il suo ambito di applicazione è il livello di chiamata del programma in cui è eseguito. Il suo effetto termina alla fine di questo programma (sia in LR sia in RT)
2. se eseguito in un altro gruppo, il suo ambito di applicazione è l'intero gruppo: il suo effetto termina alla chiusura del gruppo.

Comando DLTOVR

1. se eseguito nel gruppo di default, coincide con il valore '*', vale a dire che vengono eliminate le override aperte dal livello di chiamata in cui eseguito in poi.
2. se eseguito in un altro gruppo, vengono eliminate TUTTE le override eseguite in precedenza nello stesso gruppo di attivazione (comprese quelle eseguite in programmi chiamanti).

Nelle linee guida abbiamo detto che gli override hanno normalmente lo scope di "livello di chiamata". Quindi normalmente sarà necessario impostare OVRSCOPE(*CALLLVL) per i comandi OVRDBF e OVRPRTF e LVL(*) per DLTOVR . A meno che non si voglia specificamente uno scope diverso (gruppo di attivazione o job).
Le /COPY £OVR e £DOV utilizzano già tali valori come default.
Anche per questo motivo, nei programmi RPG non deve mai essere esguito il comando di OVRDBF direttamente ma bisogna sempre utilizzare la /COPY £OVR (e analogamente DLTOVR - £DOV).

RCLRSC

L'RCLRSC invocato da un programma eseguito in un gruppo di attivazione non default è inefficace.
Non effettua la chiusura di nessun file e di nessun programma.
Per questo motivo il comando RCLRSC non ha ragione di essere utilizzato .

LR/RT
Dato che il comando RCLRSC nella nuova "struttura" è inefficace, è stato tolto dalla £INZSR.
Di conseguenza quando un programma si chiude in LR, non effettua la chiusura dei programmi da esso richiamati e che avevano una chiusura in RT .
Questa che sembrerebbe una limitazione in realtà è un "pulizia" dei programmi. Infatti in questo modo un programma che si chiude in RT resta "aperto" fino a quando non si chiude il gruppo di attivazione" in cui sta girando, indipendentemente da quale programma l'ha richiamato.
E' quindi di fondamentale importanza considerare questo fatto.
Chiariamo la situazione con un esempio:
Il programma A (LR) richiama il programma B (RT). Entrambi sono nello stesso gruppo di attivazione.
Al primo richiamo del programma A (che poi richiamerà B), entrambi i programmi passeranno dalla loro £INIZI, quindi entrambi effettueranno l'operazione di IN della LDA (inclusa nella £INZSR).
Alla chiusura di A, esso effettuerà la OUT (inclusa anch'essa nella £INZSR) e chiuderà i file da esso aperti.
Al secondo di chiamo del programma A, esso ripasserà dalla £INIZI e ripeterà l'operazione di OUT. Il secondo richiamo del programma B invece non ripasserà dalla £INIZI e non comporterà l'operazione di IN.
Nella precedente struttura dei programmi Sme.UP (Pre V3R1), la chiusura del programma A avrebbe causato l'invocazione del comando RCLRSC che avrebbe causato la chiusura ANCHE del programma B. Anche il secondo richiamo di B avrebbe quindi comportato il passaggio presso la £INIZI e l'esecuzione del comando IN.
Bisogna precisare che questo comportamento non era "certo". Infatti c'erano in gioco altre variabili come la possibilità che altri programmi (C) chiamassero a loro volta B. Questo è uno dei motivi per cui questo tipo di struttura non era considerata "solida".

Uso di RCLACTGRP
Come indicato nelle linee guida, esistono gruppi di attivazione che non si chiudono mai (se non al cambio ambiente) e altri che invece possono essere chiusi.
Per mantenere valida questa impostazione ovviamente è imperativo che nessun programma esegua il comando RCLACTGRP ACTGRP(*ELIGIBLE) .
L'utilizzo di tale comando è comunque sconsigliatissimo in ogni manuale o forum che si rispetti.
Attualmente l'unico gruppo di attivazione che non si chiude mai è il SM_ENDLESS.

Menù e gruppi di attivazione
Tutte le azioni richieste da menù vengono eseguite in un nuovo gruppo di attivazione. questo per consentire il recupero della memoria utilizzata una volta tornati a menù.

Riga di comando
Come detto le azioni invocate da riga di comando vengono eseguite in un nuovo gruppo di attivazione.
Per ottenere da riga di comando lo stesso "effetto" è necessario che anche le "azioni" richiamate da essa, vengano eseguite in un nuovo gruppo di attivazione. Di questa operazione si occupano tutti i comandi Sme.UP invocabili (UP, T, C ecc.).
Ovviamente non è possibile effettuare questa operazione sul comando CALL.
E' quindi necessario che le "call" da riga comando vengano effettuate tramite il comando C e non CALL .

Cambio ambiente
La nuova struttura di Sme.UP, come detto, è stata pensata (tra l'altro) per garantire un corretto cambio ambiente.
Per poter garantire questo comportamento è necessario rispettare alcune regole.
Innanzitutto non è possibile effettuare un cambio ambiente "run time". Non è possibile cioè in un programma effettuare una cambio lista librerie .
Questo perché la probabilità che alcuni file restino "aperti" nel precedente ambiente (quindi puntino ad un libreria non in lista) è troppo alta. L'unico modo per essere sicuri che questo cambio ambiente sia "sicuro" (ossia tutti i file vengano chiusi) è quello di tornare a menù e scegliere un ambiente differente.

Esecuzione "comandi" in un diverso sistema informativo

Vista l'impossibilità di effettuare un cambio ambiente in un programma, è stato creato un comando apposito (B£QQ01) che consente di sottomettere un comando con una lista librerie diversa da quella del job sottomettente .
Tale sottomissione può essere bloccante o meno. Se è bloccante, il job resta in attesa del completamento, altrimenti prosegue normalmente.
Questi sono i parametri che è possibile specificare nel comando:

- B£B£B: B£B da utilizzare (parametro obbligatorio)
- B£CMD: comando da eseguire (parametro obbligatorio)
- B£JOBQ: JOBQ in cui eseguire il comando. Se tale campo viene lasciato blank, l'esecuzione del comando diventa bloccante, ossia il programma che esegue il comando B£QQ01 attende che il comando termini prima di proseguire.
- B£MSGQ: MSGQ del lavoro
- B£JOB: Nome JOB
- B£USER: Utente di esecuzione del comando

Tale comando ha attualmente alcune limitazioni:

- non consente di ricevere una risposta dal comando eseguito
- non consente di eseguire operazioni "interattive" all'interno del comando eseguito (essendo eseguito appunto in un lavoro batch)

Gestione ambienti non Sme.UP e lavori sottomessi

Un discorso analogo va fatto per i comandi (e i programmi) sottomessi o eseguiti da ambienti non Sme.UP (come ad esempio i moduli base non Sme.UP - ACG, Gruppo Pro ecc. -).
Tali comandi, come detto precedentemente, devono essere "incapsulati" nel comando B£QQ02 .
Il comando B£QQ02 esegue tutte le impostazioni necessarie (gruppi di attivazione, B£1, B£2, impostazioni di localizzazione ecc.) e successivamente esegue il comando indicato.
Quindi non sarà consentito effettuare l'operazione
SBMJOB CMD(CALL PGM(XXX))
la quale andrà sostituita con:
SBMJOB CMD(B£QQ02 B£CMD(CALL PGM(XXX)))
Identico discorso per azioni eseguite ad esempio da modulo base ACG. Comandi e CALL dovranno puntare sempre al comando B£QQ02.

Indipendenza dai valori di sistema

L'esecuzione di Sme.UP dovrebbe essere il più possibile indipendente dai valori di default impostati per i comandi (almeno alcuni).
In questo modo si lascia piena libertà agli amministratori del sistema di impostare i valori che preferiscono. Inoltre preveniamo possibili malfunzionamenti dovuti al cambio dei default stessi da parte di altri tool.

SQL

In quest'ottica di indipendenza sono state create le /COPY £INIZSQ1 e £INIZSQ2.
Tali /COPY impostano i parametri di compilazione dei programmi SQLRPGLE in cui sono inserite disinteressandosi di quanto impostato come default di sistema.

Alcuni "comportamenti" errati

Gruppo di attivazione *NEW

Erroneamente si pensa che l'esecuzione di un comando (o un programma) in un gruppo di attivazione *NEW significhi aprire un "ramo" isolato e indipendente da quanto è stato eseguito precedentemente.
Questo non è vero, infatti eventuali programmi eseguiti all'interno di questo "ramo" che non sono stati compilati con Gruppo di Attivazione *CALLER potrebbero essere già stati aperti e quindi essere già inizializzati.

Adeguamenti programmi pre V3R1
I programmi scritti prima della V3R1 (e che quindi non beneficiavano di questa nuova architettura) devono essere adeguati per evitare comportamenti dannosi od errati.
I programmi standard sono ovviamente stati adeguati. Per quanto riguarda quelli personali si rimanda alla PTF

B£01027A - Revisione architettura programmi
::REL D(21/11/2010)

Revisione architettura programmi

Nelle release precedenti di Sme.UP i Gruppi di Attivazione erano utilizzati in modo non strutturato.

Questo comportava vari problemi, il più grave dei quali era il malfunzionamento del cambio ambiente. In alcuni casi, dopo aver effettuato un cambio di sistema informativo, alcuni file restavano "aperti" sul precedente ambiente.
Un altro problema "minore" era il funzionamento disomogeneo dello stesso programma in base al "tipo" di gruppo di attivazione in cui veniva eseguito. Lo stesso programma poteva avere due funzionamenti diversi in base al fatto che venisse eseguito nel gruppo di attivazione di default o uno specifico.

Per ovviare a tutti questi problemi è stata fatta una profonda revisione della struttura dei richiami dei programmi Sme.UP.

La differenza fondamentale rispetto al passato è che ora tutti i programmi Sme.Up vengono eseguiti in un gruppo di attivazione NON DEFAULT.
Nessun programma di Sme.UP viene eseguito nel gruppo di attivazione di default (come è giusto che sia in una struttura ILE).
Questo risultato è stato ottenuto associando un gruppo diverso dal default a tutti i "primi" programmi nello stack dei richiami.
Per questo motivo, lo stack dei richiami di un qualunque job Sme.UP avrà in testa un programma specifico (B£QQ50, B£QQ02, B£QQ01 ecc.).
Tali programmi, oltre a garantire l'esecuzione in un gruppo di attivazione non default si occupreanno della corretta impostazione dei separatori decimali e delle tabelle B£1 e B£2.

Per mantenere tutta questa struttura coerente, il programmatore dovrà tenere conto di alcuni accorgimenti.
Questi accorgimenti sono elencati di seguito e saranno approfonditi in PTF specifiche.

SBMJOB tramite B£QQ01 o B£QQ02
Per le motivazioni esposte precedentemente non sarà più possibile effettuare il SBMJOB di un programma qualunque. Bisognerà sempre fare il SBMJOB del B£QQ02 oppure usare direttamente il comando B£QQ01

B£01027D - SBMJOB tramite B£QQ01 o B£QQ02
::REL D(21/11/2010)

SBMJOB tramite B£QQ01 o B£QQ02

In seguito alle modifiche aportate alla struttura dei richiami, ogni job Sme.UP deve richiamare come primo programma un programma particolare. Sarà quindi necessario adeguare gli eventuali SBMJOB effettuati.

Effettuare un SBMJOB di un programma qualunque potrebbe tradursi nell'avvio di un JOB con alcuni

parametri non corretti (come il grupppo di attivazione, la lingua o il separatore decimale da usare
o le tabelle B£1 e B£2).
Non è quindi consentito effettuare il SBMJOB di un programma qualunque.
Il comando sottomesso tramite SBMJOB deve SEMPRE essere il B£QQ02, il quale riceverà poi in input
il "vero" comando o programma da eseguire.

Inoltre come spiegato anche nella PTF

B£10328A - Variabili di ambiente

Tutte le SBMJOB devono avere indicato CPYENVVAR(*YES)

Azioni da eseguire

Il comando
SBMJOB CMD(CALL PGM(XXXX) PARM('AAAA' 'BBBB'))
JOBQ(XXJOBQ)
va quindi sostituito con
SBMJOB CMD(B£QQ02 B£CMD(CALL PGM(XXXX) PARM('AAAA' 'BBBB')))
JOBQ(XXJOBQ) CPYENVVAR(*YES)
Per ulteriore documentazione riferirsi all'help del comando B£QQ02.

E' possibile utilizzare in alternativa il comando B£QQ01.
Per la descrizione di tale comando si rimanda alla documentazione specifica, si ricorda solo che
è il comando utilizzato per sottomettere un job con una B£B diversa dall'attuale.

No cambio ambiente run-time
Non è possibile effettuare un cambio ambiente tramite programmi o comandi. Per cambiare ambiente
è necessario tornare alla lista ambienti iniziale.

B£01027E - Cambio ambiente solo da lista
::REL D(21/11/2010)

Cambio ambiente solo da lista

L'unico cambio ambiente "sicuro" è quello effettuato dalla "lista ambienti".
Con sicuro si intende la certezza di aver chiuso tutti i file e quindi la certezza di riaprirli
nella nuova lista librerie.
La "lista ambienti" è appunto la lista degli ambienti (sistemi informativi) che in 5250
si ottiene alla connessione ed è raggiungibile con F23 (se abilitato) da menù.
In loocup è quella che si ottiene scegliendo la voce "cambio ambiente".

Per questo motivo è stata eliminata ogni possibilità di effettuare un cambio ambiente run-time.
Sono quindi stati eliminati i comandi UP AMB e AM e il programma B£GGP1S.

Per i programmatori era comodo potersi collegare ad un sistema informativo "non in lista".
Per poter effettuare questa operazione è possibile (solo dalla lista ambienti 5250) utilizzare
il tasto funzioanle F21 ed immettere un qualunue sistema informativo valido.
Tale funzionalità è inibita se l'utente ha possibilità limitate.

Operazioni da eseguire
I programmi che utilizzavano i comandi AM o UP AMB o direttamente il programma B£GGP1S vanno
rivisti in quest'ottica.
Non è possibile effettuare un cambio di lista librerie.
La necessità di eseguire operazioni (non interattive) in un sistema informativo diverso può
essere soddisfatta utilizzando il comando B£QQ01.

No RCLACTGRP(*ELIGIBLE)
Questo comando non può essere utilizzato in quanto andrebbe a chiudere anche gruppi di attivazione pensati per non essere mai chiusi.

No RCLRSC
Il comando RCLRSC diventa inutile, tale comando infatti ha effetto solo se eseguito nel gruppo di attivazione di default.

Attenzione a *INLR e *INRT
La chiusura in *INLR di un programma non causa più la chiusura effettiva dei programmi da lui richiamati.

*B£01027C - *INLR, *INRT, e comandi di reclaim*
::REL D(21/11/2010)

*INLR, *INRT, e comandi di reclaim

I programmi eseguiti in un gruppo di attivazione specifico o nuovo (come ora accade per tutti i programmi Sme.UP) hanno un comportamento diverso rispetto a quelli che vengono eseguiti nel gruppo di attivazione di default (*DFTACRGRP).

RCLRSC
Questo comando è ora inefficace. Non fa nulla. Se un programma utilizzava questo comando, è necessario capire perché lo faceva e modificare di conseguenza il programma.
In questo caso le soluzioni vanno valutate di volta in volta. In alcuni casi può essere necessario chiudere esplicitamente alcuni file, oppure chiudere il programma in LR, oppure eseguire il programma in un gruppo di attivazione *NEW.

*INLR e *INRT
I programmi che si chiudono in LR, non effettuano nessuna "chiusura automatica" degli eventuali programmi da essi richiamati.
E' quindi fondamentale non dare per scontato questo meccanismo che invece prima era presente (grazie all'RCLRSC che era presente nella £INZSR). Va fatto notare che anche nelle versioni precedenti questo comportamento non era comunque garantito per vari motivi.

RCLACTGRP(*ELIGIBLE)
Questo comando non va mai usato. Questo comando viene eseguito dai programmi standard solo al cambio ambiente. Ogni altro utilizzo può dare luogo a comportamenti inaspettati, perché verrebbero chiusi anche gruppi di attivazione che devono esserlo solo al cambio ambiente.

Operazioni da eseguire
Operazioni di Reclaim
Eliminare tutti gli RCLACTGRP(*ELIGIBLE) e tutti gli RCLRSC con i comandi più opportuni. Da notare che spesso questi comandi erano utilizzati come "forzatura" di chiusura prima di un cambio ambiente. Dato che ora non è più possibile effettuare cambi ambiente Run-time, questi comandi potrebbero essere del tutto inutili.

LR/RT
Analizzare la struttura dei programmi che si chiudono in LR per controllare che non si dia per scontata la chiusura di altri programmi a causa della loro chiusura.
Per controllare i programmi problematici è possibile utilizzare la seguente utility:
Essa ncessita due parametri: la libreria di personalizzazione con i sorgenti da analizzare e una libreria di appoggio (che deve esistere ed essere in linea) in cui verrà creato un file con i risultati ottenuti.
Oggetto generico

Nella libreria di appoggio indicata, verrà creato un file in cui verrà scritto un record per ogni situazione potenzialmente problematica.
Impostare la libreria più opportuna
Oggetto generico

Analisi dei risultati
Il file B£UX090F è così strutturato:
Campo BTCD01: libreria
Campo BTCD02: file sorgente
Campo BTCD02: membro
Campo BTCD20: motivazione della segnalazione, che può assumere i seguenti valori:

- OUT LDA: programma che termina in RT e contiene una OUT della £UDLDA (verificare se è necessario eseguire una IN in modo da riempire la LDA con dati aggiornati prima di eseguire la OUT)
- £PDS: programma che termina in RT e nel quale sono dichiarati sottocampi per la £PDS (verificare se è necessario eseguire una IN in modo da riempire la LDA con dati aggiornati prima di utilizzare i sottocampi della £PDS)
- RIT: programma che termina in RT e contiene una £RITES nella £INIZI (verificare se è necessario spostare la lettura della tabella in modo che venga eseguita ad ogni richiamo e non solo all'inizializzazione del programma, ad esempio perché si basa su paramentri ricevuti nella entry del programma)
- GENERICO: programma che termina in RT e contiene istruzioni da controllare nella £INIZI (verificare se è necessario spostare le istruzioni in modo che vengano eseguite ad ogni richiamo e non solo all'inizializzazione del programma, ad esempio perché si basano su parametri ricevuti nella entry del programma)

Scope degli override
Lo scope di default degli override (OVRDBF, OVRPRTF e DLTOVR) è il gruppo di attivazione.
Per riportarsi ad una struttura di override a livello di chiamata, è necessario impostare i parametri corretti.

B£01027B - Normalizzazione Override
::REL D(21/11/2010)

Normalizzazione OVRDBF / OVRPRTF / DLTOVR

OBIETTIVO

I comandi OVRDBF, OVRPRTF e DLTOVR hanno un comportamento di default diverso, per quanto riguarda il loro ambito di attivazione, a seconda che vengano eseguiti nel gruppo di attvazione base (*DFTACTGRP) o in un altro gruppo.
Il parametro che guida questo comportamento è OVRSCOPE per i comandi OVRDBF e OVRPRTF e il parametro LVL per il comando DLTOVR.
Il valore di default è sempre *DFTACTGRP, che fa assumere il seguente comportamento:

Comandi OVRDBF e OVRPRTF
- se eseguito nel gruppo di default, coincide con il valore *CALLLVL, vale a dire che il suo ambito di applicazione è il livello di chiamata del programma in cui è eseguito.
Il suo effetto termina alla fine di questo programma (sia in LR sia in RT)
- se eseguito in un altro gruppo, il suo ambito di applicazione è l'intero gruppo: il suo effetto termina alla chiusura del gruppo.

Comando DLTOVR

- se eseguito nel gruppo di default, coincide con il valore '*', vale a dire che vengono eliminate le override aperte dal livello di chiamata in cui eseguito in poi.
- se eseguito in un altro gruppo, vengono eleminate TUTTE le override eseguite in precenenza nello stesso gruppo di attivazione (comprese quelle eseguite in programmi chiamanti).

In precedenza i programmi (salvo eccezioni, ed in modo non ben controllato), erano eseguiti nel gruppo di attivazione base, e quindi il comportamento di OVRDBF, OVRPRTFe DLTOVR era corretto.

Con il presente rilascio i programm sono eseguiti SEMPRE in un gruppo di attivazione diverso da quello base.

E' quindi necessario ricondursi al comportamento precedente (ambito dell'OVERRIDE e della DLTOVR a partire dal livello di richiamo) che si ottiene impostando in questo modo i parametri dei comandi:
OVRDBF OVRSCOPE(*CALLLVL)
OVRPRTF OVRSCOPE(*CALLLVL)
DLTOVR LVL(*)

Per far questo è stato aggiunto alla /COPY interna £OVR

£OVR - Override di un file di data base

il paramtro £OVROS (OVRSCOPE) che assume i seguenti valori:
- 'CLV' OVRSCOPE(*CALLLVL) <--Default
- 'AGD' OVRSCOPE(*ACTGRPDFN)
- 'JOB' OVRSCOPE(*JOB)

La /Copy £DOV invece prevedeva già il paramtreo £DOVLC (dove si impostava il valore del parametro LVL).

£DOV - Delete di una override

Se non impostato, esso restava bianco, quindi si assumeva li default (*ACTGRP).
Questo comportamento è stato modificato: il valore di default è '*' (livello dic hiamata).

Adeguamento sorgenti personali CLLE

Per semplificare l'operazione di adeguamento dei CLLE è stata realizzata l'utility:

B£UTX10 - UTIL Aggiunta Scopo sulle OVRDBF de

deve ricevere in ingresso la libreria da esaminare e la libreria su cui riversare i sorgenti trattati, questa libreria non deve essere presente nel sistema, in quanto viene creata in questa fase.
Una volta processati i sorgenti, si consiglia di eseguire una compilazione di massa, questa operazione serve ad identificare le eventuali anomalie di conversione.
I limiti di quest'utiliti sono:

::PAR L(P
UN)
- Tratta i soli sorgenti di tipo CLLE
- Tratta i soli sorgenti contenenti i soli comandi di override senza definizione di ambito
- Produce una lista cartacea dei sorgenti omessi dalla conversione perchè conengono OVRDBF,

OVRPRTF o DLTOVR con ambito. Questa lista è da analizzare manualmente

Adeguamento sorgenti personali RPGLE

I programmi personali contenenti £OVR e £DOV vanno soltanto ricompliati Quelli che contengono esplicitamente i comandi OVRDBF, OVRPRTF e DLTOVR, se non hanno un OVRSCOPE o un LVL particolare, devono lanciare questi comandi con le seguenti opzioni:
OVRDBF OVRSCOPE(*CALLLVL)
OVRPRTF OVRSCOPE(*CALLLVL)
DLTOVR LVL(*)
per ricondursi al comportamento precedente.

Nota su comandi di precompilazione

Tutti gli Override inseriti nei programmi come "comandi di precompilazione" (PRP*) non necessitano

dell'aggiunta del parametro OVRSCOPE in quanto il compilatore effettua automaticamente l'override opportuno.

Nuovi gruppi di attivazione PARTICOLARI
In questa versione sono stati introdotti 2 gruppi di attivazione che meritano un'attenzione particolare.
SM_ENDLESS
Questo gruppo di attivazione si chiude solo al cambio ambiente.
SMEUP
Questo gruppo di attivazione non si chiude MAI, nemmeno al cambio ambiente. Va usato con molta cautela solo da (pochi e ben identificati) programmi standard. Tali prorammi devono necessariamente appartenere alla categoria dei B£QQ in quanto devono sempre chiudere con certezza tutti i file che aprono.

Nota sui programmi richiamati da menù e da riga di comando
I programmi richiamati da menù vengono sempre eseguiti in un gruppo di attivazione nuovo, quindi alla loro uscita consentono la riacquisizione delle risorse (chiudono i file aperti).
Per ottenre lo stesso effetto da riga di comando, è necessario utilizzare il comando C e non il comando CALL.
Quindi quando di vuole eseguire un programma da riga di comando è estremamente consigliato l'utilizzo del comando C (a cui è stata aggiunta la possibilità di ricevere i parametri).
I comandi T e UP hanno lo stesso comportamento (eseguono l'azione in un nuovo gruppo di attivazione.

Contestualmente a questo intervento è stata modificata la £INZSR e anche il meccanismo di lettura dele tabelle B£1 e B£2.
Anche in merito a questi cambiamenti sarà necessario prestare alcune attenzioni.

B£01027F - Modifica e "obbligatorietà" £INZSR
::REL D(21/11/2010)

Modifica e "obbligatorietà" £INZSR

Modifica £INZSR
Innanzitutto la £INZSR è stata adeguata alle modifiche strutturali introdotte con il rilascio V3R1.
E' stato quindi eliminato l'RCLRSC che veniva eseguito in caso di *INLR acceso.

Per "pulizia" è stato anche eliminato il tag ££FINE.
Chi avesse effettuato all'interno dei propri programmi un GOTO ££FINE deve inserire questo tag in modo espliciti oppure utilizzare un altro tag.
Analogamente è stata rimossa dalla £NAV la funzione FINE che faceva SETON LR e GOTO ££FINE.

"Obbligatorietà" £INZSR
Alcune modifiche introdotte, hanno fatto si che la presenza di una qualunque /COPY smeup abbia come prerequisito la /COPY £TABB£1DS.
La presenza di quest'ultima /COPY (£TABB£1DS) richiede "di fatto" la presenza anche della £INZSR, in quanto altrimenti tutti i campi risulterebbero vuoti.
Si consiglia "vivamente" quindi di dotare tutti i programmi smeup della struttura standard di /COPY, quindi delle /COPY £INIZH, £TABB£1DS, £PDS e £INZSR.

B£01027G - Nuova gestione B£1 e B£2 e nuova Ex
::REL D(21/11/2010)

Nuova gestione B£1 e B£2 e nuova Exit

In merito alle tabelle B£1 e B£2 sono stati eseguiti due interventi.
E' stato modificato il metodo di lettura ed è stato introdotto un ulteriore livello di exit.

Nuovo metodo di lettura

Per migliorare le performance è stata modificata la modalità di accesso alle tabelle
B£1 e B£2.
All'avvio del job, (dopo aver scelto l'ambiente in caso di utenti multiambiente), il contenuto
delle tabelle (Campo TTLIBE) viene scritto in una variabile di ambiente (di livello *JOB
ovviamente).
Successivamente ogni lettura di queste tabelle avviene da queste variabili e non dal file TABELx0F.

Per il programmatore non cambia nulla in quanto le letture continuano ad essere effettuate
automaticamente dal programma B£INZ0.
A scopo documentativo si informa che le variabili di ambiente contenenti i TTLIBE delle due tabelle
in questione, si chiamano SMEB£1 e SMEB£2.

Dato che tali variabili vengono scritte solo dopo aver scelto l'ambiente, eventuali modifiche
fatte a tali tabelle verranno recepite dal job solo dopo aver fatto un cambio ambiente.
Fino a tale momento tutti i programmi (anche quelli aperti dopo il cambio della tabella)
continueranno a leggere i valori vecchi.

Si ribadisce anche in questo caso di prestare attenzione alle autorizzazioni del comando
WRKENVVAR. Solo i programmatori devono essere in grado di modificare "manualmente" queste nuove
variabili.

Nuova exit

E' stato introdotto un nuovo livello di exit per queste tabelle.
Il suffisso di tale exit è specificabile in tabella B£2 (Campo ££B£2K).
Se impostato, induce il richiamo del programma B£QQ50V_x (x è il suffisso indicato), ad ogni
cambio ambiente.
Mediante tale programma è possibile modificare il contenuto dei campi delle tabelle B£1 e B£2.

Il campo preesistente ££B£2V mantiene il significato di Aggiustamento di tutta la £INZDS.
IL programma di exit associato a tale campo (B£INZ0_x) viene richiamato tutte le volte che viene
inizializzato un programma (tutte le volte che ripassa dalla *INZSR).

Per motivi prestazionali, in caso si debbano semplicemente modificare i campi delle tabelle B£1
o B£2, si consiglia di utilizzare la exit ££B£2K in quanto tale programma di aggiustamento
viene appunto eseguito solo al cambio ambiente e non all'inizializzazione di ogni programma.

Per rendere chiaro il significato dei due campi, è stata modificata anche la descrizione del campo
preesistente ££B£2V.
I due campi ora recano le descrizioni:
££B£2K Suff.Agg.Amb. B£1/2S (questo è il nuovo campo relativo al B£QQ50V_x)
££B£2V Suff.Agg. £INZDS (questo è il campo preesistente relativo al B£INZ0_x)

Per ulteriori dettagli si rimanda all'help della tabella B£2

NOTA sul B£UT50
Il programma B£UT50 è stato sostituito dal B£QQ50. L'oggetto PGM B£UT50 rimane (come replica del B£QQ50) per non dover necessariamente modificare tutti i profili utente.
Si consiglia comunque, ove possibile, di modificare i profili utente indicando il programma B£QQ50 invece del B£UT50.

Indice

Sme.UP

Via Iseo, 43 - 25030 - Erbusco BS
Via Varese 6/A - 20037 - Paderno Dugnano MI
www.smeup.com - info@smeup.com