

IBM

@server

iSeries

Controllo sincronizzazione





@server

iSeries

Controllo sincronizzazione

Indice

| | |
|---|----|
| Controllo sincronizzazione | 1 |
| Novità per V5R2 | 2 |
| Stampare questo argomento | 2 |
| Concetti sul controllo sincronizzazione | 3 |
| Come funziona il controllo sincronizzazione | 3 |
| Come funzionano le operazioni di commit e rollback | 4 |
| Operazione sincronizzazione | 5 |
| Operazione di rollback | 5 |
| Definizione sincronizzazione | 6 |
| Ambito di una definizione sincronizzazione | 7 |
| Nomi definizioni sincronizzazione | 10 |
| Esempio: lavori e definizioni sincronizzazione | 13 |
| Come vengono gestiti gli oggetti dal controllo sincronizzazione | 13 |
| Tipi di risorse sincronizzabili | 14 |
| Risorse sincronizzabili locali e remote | 16 |
| Possibilità di accesso di una risorsa sincronizzabile | 16 |
| Protocollo sincronizzazione di una risorsa sincronizzabile | 17 |
| File registrati su giornale e controllo sincronizzazione | 17 |
| Sequenza di voci giornale sotto il controllo sincronizzazione | 18 |
| Identificativo ciclo sincronizzazione | 20 |
| Blocco record | 21 |
| Controllo sincronizzazione e lotti dischi indipendenti | 22 |
| Considerazioni e limitazioni relative al controllo sincronizzazione | 23 |
| Controllo sincronizzazione per applicazioni batch | 25 |
| Controllo sincronizzazione a due fasi | 25 |
| Ruoli nell'elaborazione della sincronizzazione | 26 |
| Stati della transizione per il controllo sincronizzazione a due fasi | 29 |
| Definizioni sincronizzazione per controllo sincronizzazione a due fasi | 31 |
| Definizione di sincronizzazione per la sincronizzazione a due fasi: consentire indicazione di sola lettura | 31 |
| Definizione sincronizzazione per sincronizzazione a due fasi: non attendere risultato | 34 |
| Definizione sincronizzazione per sincronizzazione a due fasi: indicare OK per uscire | 39 |
| Definizione sincronizzazione per sincronizzazione a due fasi: Non selezionare l'ultimo agent | 39 |
| Influenza dell'indicazione attendibile sul flusso di elaborazione della sincronizzazione | 40 |
| Supporto transazioni XA per il controllo sincronizzazione | 42 |
| Modalità server SQL e transazioni nell'ambito del sottoprocesso per il controllo sincronizzazione | 46 |
| Avviare il controllo sincronizzazione | 47 |
| Oggetto di notifica sincronizzazione | 49 |
| Livello blocco sincronizzazione | 53 |
| Terminare controllo sincronizzazione | 53 |
| Termine del controllo sincronizzazione avviato dal sistema | 54 |
| Controllo sincronizzazione durante la fine del gruppo attivazione | 55 |
| Operazioni di commit e rollback implicite | 60 |
| Controllo sincronizzazione durante la fine normale della fase di instradamento | 61 |
| Controllo sincronizzazione durante la fine anomala di un sistema o di un lavoro | 61 |
| Aggiornamenti sull'oggetto di notifica | 63 |
| Ripristino controllo sincronizzazione durante l'IPL (initial program load) in seguito ad una fine anomala | 65 |
| Gestire le transazioni e il controllo sincronizzazione | 65 |
| Visualizzare informazioni sul controllo sincronizzazione | 66 |
| Visualizzare oggetti bloccati per una transazione | 66 |
| Visualizzare lavori associati ad una transazione | 67 |
| Visualizzare stato risorsa di una transazione | 67 |

| | |
|---|-----|
| Visualizzare proprietà transazione | 67 |
| Ottimizzare le prestazioni per il controllo sincronizzazione | 69 |
| Ridurre i blocchi | 70 |
| Gestire la dimensione della transazione | 72 |
| Scenari ed esempi: controllo sincronizzazione | 72 |
| Scenario: controllo sincronizzazione. | 73 |
| Problema pratico per il controllo sincronizzazione. | 82 |
| Flusso logico di un problema pratico | 82 |
| Fasi associate al flusso logico del problema pratico | 84 |
| Esempio: utilizzare un file di registrazione transazione per avviare un'applicazione | 85 |
| Esempio: utilizzare un oggetto di notifica per avviare un'applicazione | 90 |
| Esempio: oggetto di notifica univoco per ogni programma | 91 |
| Esempio: oggetto di notifica singolo per tutti i programmi | 96 |
| Esempio: utilizzare un programma di elaborazione standard per avviare un'applicazione | 96 |
| Esempio: codice per un programma di elaborazione standard | 97 |
| Esempio: codice per un programma di elaborazione della sincronizzazione standard. | 98 |
| Esempio: utilizzare un programma di elaborazione standard per decidere se riavviare l'applicazione. | 101 |
| Risolvere i problemi relativi alle transazioni e al controllo sincronizzazione | 102 |
| Errori controllo sincronizzazione. | 102 |
| Condizioni di errore | 103 |
| Condizioni di non errore | 104 |
| Messaggi di errore da monitorare durante il controllo sincronizzazione | 104 |
| Monitorare gli errori in seguito ad un comando CALL | 107 |
| Errore nell'elaborazione normale del commit o del rollback. | 108 |
| Rilevare stalli | 109 |
| Ripristinare le transazioni dopo un errore nelle comunicazioni. | 110 |
| Quando forzare i commit e i rollback e quando annullare la risincronizzazione. | 111 |
| Informazioni correlate al controllo sincronizzazione | 112 |

Controllo sincronizzazione

Il controllo sincronizzazione è una funzione che garantisce l'integrità dei dati. Consente la definizione e l'elaborazione di un gruppo di modifiche apportate alle risorse, come ad esempio, file database o tabelle, come una transazione. Il controllo sincronizzazione garantisce che l'intero gruppo di modifiche individuali venga apportato a tutti i sistemi partecipanti oppure garantisce che non venga apportata nessuna modifica. DB2 Universal Database for iSeries™ utilizza la funzione di controllo sincronizzazione per sincronizzare ed effettuare il rollback delle transazioni database che vengono eseguite con un livello di isolamento diverso da *NONE (Non sincronizzare).

E' possibile utilizzare il controllo sincronizzazione per progettare un'applicazione in modo tale che il sistema possa riavviarla se un lavoro, un gruppo attivazione all'interno del lavoro o il sistema terminano in modo anomalo. Tramite il controllo sincronizzazione, è possibile avere la garanzia che quando l'applicazione viene riavviata, non esistano aggiornamenti parziali nel database, a causa delle transazioni incomplete dovute ad un errore precedente.

Vedere le seguenti informazioni relative a come ottenere il controllo sincronizzazione ed eseguirlo sul server iSeries.

Novità per V5R2

Questo argomento elenca nuove informazioni per il controllo sincronizzazione.

Stampare questo argomento

Stampare queste informazioni per intero.

Concetti del controllo sincronizzazione

Leggere queste informazioni per apprendere come funziona il controllo sincronizzazione.

Avviare controllo sincronizzazione

Leggere queste informazioni per i dettagli sull'avvio del controllo sincronizzazione

Terminare controllo sincronizzazione

Leggere queste informazioni per apprendere quali sono i prerequisiti necessari per terminare il controllo sincronizzazione e come terminarlo.

Termine del controllo sincronizzazione avviato dal sistema

Leggere le attività da eseguire quando il sistema avvia la fase di termine del controllo sincronizzazione.

Gestire le transazioni e il controllo sincronizzazione

Leggere le attività da eseguire per gestire un sistema con controllo sincronizzazione.

Scenari ed esempi: controllo sincronizzazione

Leggere questi scenari ed esempi per vedere come una società imposta il controllo sincronizzazione. Leggere gli esempi di codici per i programmi che utilizzano il controllo sincronizzazione.

Risolvere i problemi relativi alle transazioni e al controllo sincronizzazione

Leggere queste informazioni quando è necessario risolvere i problemi relativi al controllo sincronizzazione.

Informazioni correlate al controllo sincronizzazione

Vedere gli argomenti, i manuali, gli IBM Redbook, e i siti Web esterni relativi al controllo sincronizzazione.

Nota: leggere la clausola di esonero della responsabilità inerente all'esempio di codice per quanto riguarda le informazioni legali rilevanti.

Novità per V5R2

Per V5R2, esistono alcuni miglioramenti e aggiunte al controllo sincronizzazione. Le seguenti voci contengono un riepilogo di tali miglioramenti e aggiunte.

- **Controllo sincronizzazione e lotti dischi indipendenti**
La capacità della nuova libreria dei lotti dischi indipendenti fornisce il supporto per il controllo sincronizzazione.
- **Supporto transazioni XA per il controllo sincronizzazione**
Il supporto transazioni XA per il controllo sincronizzazione fornisce una migliore conformità con la Specifica XA.
- **Supporto iSeries Navigator per il controllo sincronizzazione**
iSeries Navigator fornisce il supporto per il controllo sincronizzazione. I seguenti argomenti forniscono i dettagli:
 - Visualizzare le informazioni relative al controllo sincronizzazione
 - Supporto transazioni XA per il controllo sincronizzazione
 - Ripristinare le transazioni dopo un errore nelle comunicazioni
 - Quando forzare i commit e i rollback e quando annullare la risincronizzazione

Per trovare ulteriori informazioni sulle novità e sulle modifiche di questo rilascio, consultare la Memo per gli utenti



Stampare questo argomento

Per visualizzare o scaricare la versione PDF, selezionare Controllo sincronizzazione (circa 600 KB o 112 pagine).

Salvare i file PDF

Per salvare un PDF sulla propria stazione di lavoro per la visualizzazione o per la stampa:

1. Fare clic con il tastino destro del mouse sul PDF nel proprio browser (fare clic con il tastino destro sul collegamento precedente).
2. Fare clic su **Salva destinazione con nome...**
3. Spostarsi nell'indirizzario in cui si desidera salvare il PDF.
4. Fare clic su **Salva**.

Scaricare Adobe Acrobat Reader

Se è necessario Adobe Acrobat Reader per visualizzare o stampare questi PDF, è possibile scaricare una copia dal sito Web Adobe (www.adobe.com/products/acrobat/readstep.html)



Concetti sul controllo sincronizzazione

Queste pagine forniscono le informazioni per comprendere il modo in cui funziona il controllo sincronizzazione e come interagisce con il sistema e con gli altri sistemi nella rete:

- Come funziona il controllo sincronizzazione
- Come funzionano le operazioni di commit e rollback
- Definizione sincronizzazione
- Risorse sotto controllo sincronizzazione
- Controllo sincronizzazione e lotti dischi indipendenti
- Considerazioni e limitazioni relative al controllo sincronizzazione
- Controllo sincronizzazione per applicazioni batch
- Controllo sincronizzazione a due fasi
- Supporto transazioni XA per controllo sincronizzazione
- Modalità server SQL e transazioni nell'ambito del sottoprocesso

Come funziona il controllo sincronizzazione

Il controllo sincronizzazione è una funzione che consente di definire ed elaborare un gruppo di modifiche apportate alle risorse, come ad esempio i file o le tabelle database, come una transazione. Il controllo sincronizzazione garantisce che l'intero gruppo di modifiche individuali venga apportato a tutti i sistemi partecipanti oppure garantisce che non venga apportata nessuna modifica. Ad esempio, quando i fondi vengono trasferiti da un conto di risparmi a un conto corrente, si verifica più di una modifica come gruppo. E' possibile che tale trasferimento appaia come una singola modifica. Tuttavia, si verifica più di una modifica al database poiché sia i conti di risparmi sia i conti correnti vengono aggiornati. Per mantenere accurati i conti, è necessario che tutte o nessuna modifica venga apportata ai conti di risparmi o ai conti correnti.

Il controllo sincronizzazione consente di:

- Accertarsi che tutte le modifiche in una transazione siano completate per tutte le risorse interessate.
- Accertarsi che tutte le modifiche in una transazione vengano eliminate se l'elaborazione viene interrotta.
- Eliminare le modifiche effettuate durante una transazione quando l'applicazione stabilisce l'errore di una transazione.

E' possibile creare un'applicazione in modo tale che il controllo sincronizzazione possa riavviare l'applicazione se un lavoro, un gruppo attivazione all'interno di un lavoro o il sistema termina in modo anomalo. Tramite il controllo sincronizzazione, è possibile avere la garanzia che, al riavvio dell'applicazione, non sia presente alcun aggiornamento parziale nel database a causa di transazioni incomplete in seguito ad un errore precedente.

Transazione

Una transazione è un gruppo di modifiche individuali apportate agli oggetti sul sistema che dovrebbero essere visualizzate come un'unica modifica all'utente.

Nota:

iSeries navigator utilizza il termine transazione, mentre l'interfaccia basata sul carattere utilizza il termine LUW (Logical Unit of Work). I due termini sono intercambiabili. In questo argomento, a meno che non esista un riferimento specifico all'interfaccia basata sul carattere, viene utilizzato il termine transazione.

Una transazione può essere una delle seguenti:

- Richieste in cui non si verificano modifiche del file database.

- Transazioni semplici che modificano un file database.
- Transazioni complesse che modificano uno o più file database.
- Transazioni complesse che modificano uno o più file database, ma tali modifiche rappresentano solo una parte di un gruppo logico di transazioni.
- Transazioni semplici o complesse che coinvolgono file database su più di una ubicazione. I file di database possono essere:
 - Su un sistema remoto singolo.
 - Sul sistema locale e su uno o più sistemi remoti.
 - Assegnati a più di un giornale sul sistema locale. Ogni giornale può essere considerato come una **ubicazione locale**.
- Transazioni semplici o complesse sul sistema locale che coinvolgono oggetti diversi dai file database.

Come funzionano le operazioni di commit e rollback

Due operazioni influiscono sulle modifiche effettuate sotto il controllo sincronizzazione:

- Operazione di commit
Un'operazione di commit rende permanenti tutte le modifiche effettuate sotto il controllo sincronizzazione dalla precedente operazione di commit o rollback. Il sistema inoltre rilascia tutti i blocchi relativi alla transazione.
- Operazione di rollback
Un'operazione di rollback elimina tutte le modifiche effettuate dalla precedente operazione di commit o rollback. Il sistema inoltre rilascia tutti i blocchi relativi alla transazione.

I seguenti linguaggi di programmazione e API supportano le operazioni di commit e rollback:

| Linguaggio o API | Commit | Rollback |
|-----------------------------------|--|--------------------------|
| CL | Comando COMMIT | Comando ROLLBACK |
| ILE RPG/400 | Codice operazione COMIT | Codice operazione ROLBK |
| ILE COBOL/400 ^(R) | Verbo COMMIT | Verbo ROLLBACK |
| ILE C/400 ^(R) | Funzione _Rcommit | Funzione _Rrollbck |
| PL/I | Sottoroutine PLICOMMIT | Sottoroutine PLIROLLBACK |
| SQL | Istruzione COMMIT | Istruzione ROLLBACK |
| CLI (Call Level Interface) SQL | Funzione SQLTransact() (Utilizzare per il commit e il rollback di una transazione) | |
| API XA | API db2xa_commit() | API db2xa_rollback() |

I seguenti collegamenti forniscono ulteriori informazioni su questi linguaggi di programmazione e API:

- COBOL/400 User's Guide



- RPG/400 User's Guide



- ILE C/C++ Programmer's Guide



- CL Programming



- System API Programming



- DB2 UDB for iSeries SQL Call Level Interface
- DB2 UDB for iSeries Programming Concepts

Operazione sincronizzazione

Un'operazione di sincronizzazione rende permanenti tutte le modifiche effettuate sotto il controllo di sincronizzazione dalla precedente operazione di commit o rollback. Il sistema inoltre rilascia tutti i blocchi relativi alla transazione.

Quando riceve una richiesta di sincronizzazione, il sistema effettua quanto segue:

- Il sistema salva l'identificazione di sincronizzazione, se viene fornita, per utilizzarla durante il ripristino.
- Il sistema scrive i record sul file prima di eseguire l'operazione di sincronizzazione se si verifica quanto segue:
 - I record vengono aggiunti ad un file database locale o remoto sotto il controllo di sincronizzazione.
 - Viene specificato SEQONLY(*YES) quando il file è stato aperto in modo tale che venga utilizzato dal sistema il feedback I/E bloccato ed esista un blocco parziale dei record.

In caso contrario, l'area feedback I/E e i buffer I/E non vengono modificati.

- Il sistema richiama il programma di uscita commit e rollback per ogni risorsa di sincronizzazione API presente nella definizione di sincronizzazione. Se un'ubicazione dispone di più di un programma di uscita registrato, il sistema richiama i programmi di uscita per quell'ubicazione nell'ordine secondo il quale sono stati registrati.
- Se le modifiche dei record sono state effettuate sulle risorse assegnate ad un giornale, il sistema scrive una voce giornale C CM su ogni giornale locale associato alla definizione di sincronizzazione. La sequenza di voci di giornale sotto il controllo di sincronizzazione visualizza le voci che solitamente vengono scritte mentre una definizione di sincronizzazione è attiva.
- Il sistema effettua delle modifiche a livello oggetto permanenti che sono in sospeso.
- Il sistema sblocca i record e gli oggetti acquisiti e conservati per scopi di controllo di sincronizzazione. Queste risorse sono rese disponibili per gli altri utenti.
- Il sistema modifica le informazioni nella definizione di sincronizzazione per mostrare che la transazione corrente è stata terminata.

E' necessario che il sistema esegua correttamente tutte le fasi precedenti affinché l'operazione di sincronizzazione abbia esito positivo.

Operazione di rollback

Un'operazione di rollback elimina tutte le modifiche effettuate dalla precedente operazione di commit o rollback. Il sistema inoltre rilascia tutti i blocchi relativi alla transazione. Il sistema effettua quanto segue quando riceve una richiesta di rollback:

- Il sistema elimina i record dal buffer I/E se si verificano entrambe le seguenti situazioni:
 - Se i record sono stati aggiunti ad un file database locale o remoto sotto il controllo di sincronizzazione.
 - Se è stato specificato SEQONLY(*YES) all'apertura del file, in modo tale che venga utilizzata dal sistema l'I/E bloccata ed esista un blocco parziale dei record non ancora scritto sul database.

In caso contrario, l'area feedback I/E e i buffer I/E rimangono invariati.

- Il sistema effettua una chiamata al programma di uscita commit o rollback per ogni risorsa sincronizzazione API presente nella definizione sincronizzazione. Se un'ubicazione dispone di più di un programma di uscita registrato, il sistema richiama i programmi di uscita per quell'ubicazione in ordine inverso all'ordine in cui sono stati registrati.
- Se un record è stato cancellato da un file, il sistema aggiunge di nuovo il record al file.
- Il sistema elimina qualsiasi modifica sui record effettuata durante questa transazione e inserisce di nuovo i record originali (immagini precedenti) nel file.
- Se eventuali record sono stati aggiunti al file durante questa transazione, rimangono nel file come record cancellati.
- Se le modifiche dei record sono state effettuate sulle risorse assegnate ad un giornale durante la transazione, il sistema aggiunge una voce giornale di C RB al giornale, indicando che si è verificata un'operazione di rollback. Inoltre, il giornale contiene le immagini delle modifiche dei record sottoposte a rollback. Prima di richiedere l'operazione di rollback, le immagini precedenti e successive dei record modificati sono state ubicate nel giornale. Il sistema scrive la voce C RB sul giornale predefinito se sono state assegnate a quel giornale delle risorse sincronizzabili.
- Il sistema posiziona i file di apertura sotto il controllo sincronizzazione in una delle seguenti posizioni:
 - L'ultimo record a cui si è avuto accesso durante la precedente transazione
 - Sulla posizione aperta se non è stata eseguita nessuna operazione di sincronizzazione per il file utilizzando questa definizione sincronizzazione

Questa considerazione è importante se si sta eseguendo un'elaborazione sequenziale.

- Il sistema non esegue il rollback delle modifiche non sincronizzabili per i file database. Ad esempio, i file aperti non vengono chiusi e quelli eliminati non vengono memorizzati. Il sistema non riapre o riposiziona i file che sono stati chiusi durante questa transazione.
- Il sistema sblocca i blocchi dei record ottenuti per scopi di controllo sincronizzazione e rende disponibili questi record ad altri utenti.
- L'identificazione sincronizzazione attualmente salvata dal sistema rimane la stessa di quella fornita con l'ultima operazione di sincronizzazione per la stessa definizione sincronizzazione.
- Il sistema inverte o esegue il rollback delle modifiche sincronizzabili a livello oggetto effettuate durante questa transazione.
- I blocchi oggetti ottenuti per scopi di controllo sincronizzazione vengono sbloccati e tali oggetti vengono resi disponibili ad altri utenti.
- Il sistema stabilisce il precedente limite di sincronizzazione come limite di sincronizzazione corrente.
- Il sistema modifica le informazioni nella definizione sincronizzazione per mostrare che la transazione corrente è stata terminata.

E' necessario che il sistema esegua correttamente tutte le fasi precedenti affinché l'operazione di sincronizzazione abbia esito positivo.

Definizione sincronizzazione

Una definizione sincronizzazione viene creata quando viene utilizzato il comando Avvio controllo sincronia (STRCMTCTL) per avviare il controllo sincronizzazione sul sistema. Inoltre, DB2 UDB for iSeries crea automaticamente una definizione sincronizzazione quando il livello di isolamento è diverso da Non sincronizzare. La definizione sincronizzazione contiene informazioni che appartengono alle risorse modificate sotto il controllo sincronizzazione all'interno di quel lavoro. Durante la modifica delle risorse sincronizzazione, il sistema mantiene le informazioni relative al controllo sincronizzazione nella definizione sincronizzazione, fino a quando quest'ultima non viene completata. Ogni transazione attiva sul sistema viene rappresentata da una definizione sincronizzazione. Una transazione successiva può riutilizzare una definizione sincronizzazione in seguito ad ogni commit o rollback di una transazione attiva.

Una definizione sincronizzazione generalmente include:

- I parametri sul comando STRCMTCTL.

- Lo stato corrente della definizione sincronizzazione.
- Le informazioni relative ai file database e alle altre risorse sincronizzabili che contengono le modifiche apportate durante la transazione corrente.

Per le definizioni con blocchi nell'ambito del lavoro, la definizione sincronizzazione è nota solo al lavoro che avvia la definizione sincronizzazione. Nessun altro lavoro conosce la definizione sincronizzazione.

I programmi possono avviare ed utilizzare più definizioni sincronizzazione. Ogni definizione sincronizzazione per un lavoro identifica una transazione separata che dispone di risorse sincronizzabili ad essa associate. Queste transazioni possono essere sottoposte a commit o a rollback indipendentemente dalle transazioni associate ad altre definizioni sincronizzazione che sono state avviate per il lavoro.

Le sezioni seguenti forniscono ulteriori dettagli sulle definizioni sincronizzazione:

- Ambito di una definizione sincronizzazione
- Nomi definizioni sincronizzazione
- Esempio: lavori e definizioni sincronizzazione

Per le norme relative alle definizioni sincronizzazione e ai lotti dischi indipendenti, vedere Controllo sincronizzazione e lotti dischi indipendenti.

Ambito di una definizione sincronizzazione

L'**ambito** di una definizione sincronizzazione stabilisce quali programmi utilizzeranno quella definizione sincronizzazione e come vengono inseriti nell'ambito i blocchi ottenuti durante le transazioni. L'interfaccia che avvia la definizione sincronizzazione determina l'ambito della definizione sincronizzazione. Esistono quattro ambiti possibili per una definizione sincronizzazione, che vengono raggruppati in due categorie generali:

Definizioni sincronizzazione con blocchi nell'ambito del lavoro

- Definizione sincronizzazione a livello del gruppo attivazione
- Definizione sincronizzazione a livello del lavoro
- Definizione sincronizzazione denominata esplicitamente

Definizioni sincronizzazione con blocchi nell'ambito della transazione

- Definizione sincronizzazione nell'ambito della transazione

Le definizioni sincronizzazione con blocchi nell'ambito del lavoro possono essere utilizzate solo da programmi eseguiti sul lavoro che ha avviato le definizioni sincronizzazione. In confronto, più di un lavoro può utilizzare le definizioni sincronizzazione con blocchi nell'ambito della transazione.

Solitamente, le applicazioni utilizzano sia le definizioni sincronizzazione alivello del gruppo attivazione sia quelle a livello del lavoro. Tali definizioni sincronizzazione vengono create esplicitamente tramite il comando Avvio controllo sincronia (STRCMTCTL) o implicitamente dal sistema quando un'applicazione SQL è in esecuzione con un livello di isolamento diverso da *NONE.

Definizione sincronizzazione a livello del gruppo attivazione

L'ambito più comune è sul gruppo attivazione. La definizione sincronizzazione alivello del gruppo attivazione è l'ambito predefinito quando il comando STRCMTCTL avvia esplicitamente la definizione sincronizzazione o quando un'applicazione SQL in esecuzione con un livello di isolamento diverso da Non sincronizzare avvia implicitamente la definizione sincronizzazione. Solo i programmi in esecuzione all'interno di quel gruppo attivazione utilizzano tale definizione sincronizzazione. Molte definizioni sincronizzazione a livello del gruppo attivazione possono essere attive allo stesso tempo per un lavoro. Tuttavia, ogni definizione sincronizzazione a livello del gruppo attivazione può essere associata solo ad un

gruppo attivazione singolo. I programmi in esecuzione all'interno di tale gruppo attivazione possono associare le relative modifiche sincronizzabili solo con tale definizione sincronizzazione a livello del gruppo attivazione.

In iSeries Navigator, il comando Gestione definizione sincronia (WRKCMTDFN), il comando DSPJOB (Visualizzazione lavoro) o il comando WRKJOB (Gestione lavoro) visualizzano una definizione sincronizzazione a livello del gruppo attivazione, in questi campi viene visualizzato quanto segue:

- Nel campo definizione sincronizzazione viene visualizzato il nome del gruppo attivazione. Viene visualizzato il valore speciale *DFACTGRP per indicare il gruppo attivazione predefinito.
- Nel campo gruppo attivazione viene visualizzato il numero del gruppo attivazione.
- Nel campo lavoro viene visualizzato il lavoro che ha avviato la definizione sincronizzazione.
- Nel campo sottoprocesso viene visualizzato *NONE.

Definizione sincronizzazione a livello del lavoro

Una definizione sincronizzazione può essere inserita nell'ambito del lavoro solo emettendo STRCMTCTL CMTSCOPE(*JOB). Qualsiasi programma in esecuzione in un gruppo attivazione che non dispone di una definizione sincronizzazione a livello del gruppo attivazione avviata, utilizza la definizione sincronizzazione a livello del lavoro, se è già stata avviata da un altro programma per il lavoro. E' possibile avviare solo una singola definizione sincronizzazione a livello del lavoro per un lavoro.

In iSeries Navigator, il comando Gestione definizione sincronia (WRKCMTDFN), il comando DSPJOB (Visualizzazione lavoro) o il comando WRKJOB (Gestione lavoro) visualizzano una definizione sincronizzazione a livello del lavoro, in questi campi viene visualizzato quanto segue:

- Nel campo definizione sincronizzazione viene visualizzato il valore speciale *JOB.
- Nel campo gruppo attivazione viene visualizzato uno spazio vuoto.
- Nel campo lavoro viene visualizzato il lavoro che ha avviato la definizione sincronizzazione.
- Nel campo sottoprocesso viene visualizzato *NONE.

Per un determinato gruppo attivazione, i programmi in esecuzione all'interno di tale gruppo attivazione possono utilizzare solo una singola definizione sincronizzazione. Quindi, i programmi in esecuzione all'interno di quel gruppo attivazione possono utilizzare o la definizione sincronizzazione a livello del lavoro o quella a livello del gruppo attivazione ma non entrambe allo stesso tempo. In un lavoro a più sottoprocessi che non utilizza la modalità server SQL, il lavoro di transazione per un programma verrà inserito nell'ambito dell'appropriata definizione sincronizzazione in base al gruppo attivazione del programma, a prescindere da quale sottoprocesso viene eseguito. Se più sottoprocessi utilizzano lo stesso gruppo attivazione, devono cooperare per eseguire il lavoro di transazione e accertarsi che i commit e i rollback si verifichino nell'ora corretta.

Anche quando la definizione sincronizzazione a livello del lavoro è attiva per il lavoro, un programma può avviare la definizione sincronizzazione a livello del gruppo attivazione se nessun programma in esecuzione all'interno di quel gruppo attivazione ha eseguito delle richieste o operazioni di controllo sincronizzazione per la definizione sincronizzazione a livello del lavoro. In caso contrario, è necessario terminare la definizione sincronizzazione a livello del lavoro prima di poter avviare la definizione sincronizzazione a livello del gruppo attivazione. Le richieste o le operazioni di controllo sincronizzazione per la definizione sincronizzazione a livello del lavoro che può impedire l'avvio della definizione sincronizzazione a livello del gruppo attivazione, includono:

- Apertura (completa o condivisa) di un file database sotto il controllo sincronizzazione.
- Utilizzo dell'API QTNADDCR (Aggiunta risorsa sincronizzazione) per aggiungere una risorsa sincronizzazione API.
- Sincronizzazione di una transazione.
- Rollback di una transazione.
- Aggiunta di una risorsa remota sotto il controllo sincronizzazione.

- Utilizzo dell'API QTNCHGCO (Modifica opzioni sincronizzazione) per modificare le opzioni di sincronizzazione.
- Portare la definizione sincronizzazione ad uno stato di rollback richiesto tramite l'API QTNRBRQD (Rollback richiesto).
- Invio di una voce giornale utente che comprende l'identificativo ciclo sincronizzazione corrente utilizzando l'API QJOSJRNE (Invio voce giornale) con il parametro Includere identificativo ciclo sincronizzazione.

Allo stesso modo, se i programmi all'interno di un gruppo attivazione utilizzano attualmente la definizione sincronizzazione a livello del gruppo attivazione, la definizione sincronizzazione deve essere terminata prima che i programmi in esecuzione all'interno dello stesso gruppo attivazione possano utilizzare la definizione sincronizzazione a livello del lavoro.

Durante l'apertura di un file database, l'ambito di apertura per il file aperto può essere il gruppo attivazione o il lavoro, con una limitazione: se un programma sta aprendo un file sotto il controllo sincronizzazione e il file si trova nell'ambito del lavoro, il programma che effettua la richiesta di apertura deve utilizzare la definizione sincronizzazione a livello del lavoro.

Definizione sincronizzazione denominata esplicitamente

Le definizioni sincronizzazione denominate esplicitamente vengono avviate dal sistema quando quest'ultimo necessita di eseguire le relative transazioni di controllo sincronizzazione senza influenzare le transazioni utilizzate da un'altra applicazione. Un esempio di funzione che avvia questi tipi di definizioni sincronizzazione è la registrazione problemi. Un'applicazione non può avviare le definizioni sincronizzazione denominate esplicitamente.

In iSeries Navigator, il comando Gestione definizione sincronia (WRKCMTDFN), il comando DSPJOB (Visualizzazione lavoro) o il comando WRKJOB (Gestione lavoro) visualizzano una definizione sincronizzazione denominata esplicitamente, in questi campi viene visualizzato quanto segue:

- Nel campo definizione sincronizzazione viene visualizzato il relativo nome specificato dal sistema.
- Nel campo gruppo attivazione viene visualizzato uno spazio vuoto.
- Nel campo lavoro viene visualizzato il lavoro che ha avviato la definizione sincronizzazione.
- Nel campo sottoprocesso viene visualizzato *NONE.

Definizioni sincronizzazione nell'ambito della transazione

Le definizioni sincronizzazione nell'ambito della transazione vengono avviate tramite API XA per i Blocchi nell'ambito della transazione.

Queste API utilizzano i protocolli controllo sincronizzazione basati su sottoprocessi o su collegamenti SQL e non basati sul gruppo attivazione. In altre parole, le API vengono utilizzate per associare la definizione sincronizzazione ad un sottoprocesso particolare o ad un collegamento SQL mentre viene eseguito il lavoro di transazione, e vengono utilizzate per eseguire il commit o il rollback delle transazioni. Il sistema collega queste definizioni sincronizzazione ai sottoprocessi che eseguono il lavoro di transazione, in base ai protocolli API. Possono essere utilizzate dai sottoprocessi nei differenti lavori.

In iSeries Navigator, il comando Gestione definizione sincronia (WRKCMTDFN), il comando DSPJOB (Visualizzazione lavoro) o il comando WRKJOB (Gestione lavoro) visualizzano una definizione sincronizzazione nell'ambito della transazione, in questi campi viene visualizzato quanto segue:

- Nel campo definizione sincronizzazione viene visualizzato il valore speciale *TNSOBJ.
- Nel campo gruppo attivazione viene visualizzato uno spazio vuoto.
- Nel campo lavoro viene visualizzato il lavoro che ha avviato la definizione sincronizzazione.

- Nel campo sottoprocesso viene visualizzato il sottoprocesso a cui è collegata la definizione sincronizzazione (o *NONE se la definizione sincronizzazione non è collegata attualmente a nessun sottoprocesso).

Nomi definizioni sincronizzazione

Il sistema assegna i nomi a tutte le definizioni sincronizzazione avviate per un lavoro. Nella seguente tabella vengono visualizzate le varie definizioni sincronizzazione e i nomi associati per un lavoro particolare.

| Gruppo attivazione | Ambito sincronizzazione | Nome definizione sincronizzazione |
|---|-------------------------|--|
| Qualsiasi | Lavoro | *JOB |
| Gruppo attivazione predefinito | Gruppo attivazione | *DFACTGRP |
| Gruppo attivazione definito dall'utente | Gruppo attivazione | Nome gruppo attivazione (ad esempio, PAYROLL) |
| Gruppo attivazione definito dal sistema | Gruppo attivazione | Numero gruppo attivazione (ad esempio. 000000145) |
| Nessuno | Definito esplicitamente | QDIR001 (esempio di una definizione sincronizzazione definita dal sistema solo per l'utilizzo del sistema). I nomi definizione sincronizzazione definiti dal sistema iniziano con Q. |
| Nessuno | Transazione | *TNSOBJ |

Solo i programmi compilati ILE (Integrated Language Environment) possono avviare il controllo sincronizzazione per i gruppi attivazione diversi da quello predefinito. Quindi, un lavoro può utilizzare più definizioni sincronizzazione solo se il lavoro è in esecuzione su uno o più programmi compilati ILE. Per ulteriori informazioni relative all'ILE (Integrated Language Environment) ^(R), vedere ILE Concepts



I programmi OPM (Original Program Model) vengono eseguiti sul gruppo attivazione predefinito, e per impostazione predefinita utilizzano la definizione sincronizzazione *DFACTGRP. In un ambiente OPM e ILE misto, se tutte le modifiche sincronizzabili effettuate dai programmi devono essere sottoposte a commit o a roll back insieme, i lavori devono utilizzare la definizione sincronizzazione a livello del lavoro.

E' possibile associare un file database aperto nell'ambito di un gruppo attivazione, con una definizione sincronizzazione a livello del gruppo attivazione o del livello lavoro. Un file database aperto nell'ambito del lavoro può essere associato solo alla definizione sincronizzazione a livello del lavoro. Quindi, qualsiasi programma, OPM o ILE, che apre un file database sotto il controllo sincronizzazione nell'ambito del lavoro, richiede l'utilizzo della definizione sincronizzazione a livello del lavoro.

I programmi dell'applicazione non utilizzano il nome definizione sincronizzazione per identificare una definizione sincronizzazione particolare durante una richiesta di controllo sincronizzazione. I nomi definizioni sincronizzazione vengono utilizzati principalmente nei messaggi per identificare una definizione sincronizzazione particolare per un lavoro.

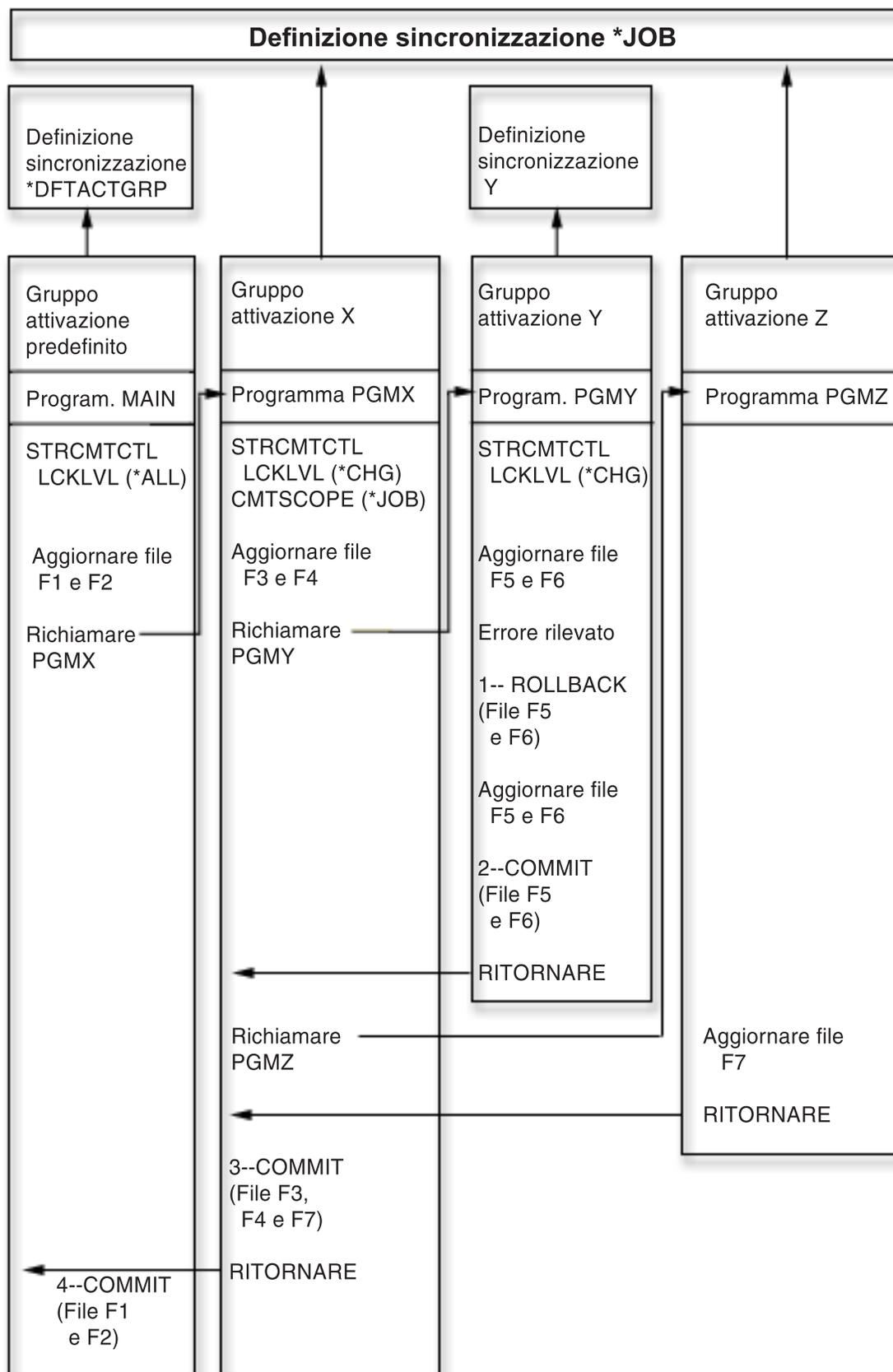
Per le definizioni sincronizzazione a livello del gruppo attivazione, il sistema determina quale definizione utilizzare in base al gruppo attivazione su cui è in esecuzione il programma richiedente. Questo è possibile poiché i programmi che vengono eseguiti in un gruppo attivazione in qualsiasi momento possono utilizzare solo un'unica definizione sincronizzazione.

Per le transazioni con blocchi nell'ambito delle transazioni, le API XA e gli attributi correlati alla transazione aggiunti a CLI, determinano quale definizione sincronizzazione viene utilizzata dal sottoprocesso richiamante.

Esempio: lavori e definizioni sincronizzazione

La seguente figura mostra un esempio di un lavoro che utilizza più definizioni sincronizzazione. Viene indicato quali aggiornamenti di file sono sottoposti a commit o a rollback su ogni livello di gruppo attivazione. L'esempio presuppone che tutti gli aggiornamenti effettuati sui file database da tutti i

programmi, vengono eseguiti sotto il controllo sincronizzazione.



*

Nella seguente tabella viene illustrato il modo in cui viene eseguito il commit o il rollback dei file se lo scenario nella figura precedente viene modificato:

Ulteriori esempi di più definizioni sincronizzazione in un lavoro

| Modifica dello scenario | Effetto delle modifiche apportate a questi file: | | | |
|---|--|--------------------------------------|-------------------|---|
| | F1 e F2 | F3 e F4 | F5 e F6 | F7 |
| PGMX esegue un'operazione di rollback invece di un'operazione di commit (3= =COMMIT diventa ROLLBACK). | Ancora in sospeso | Eseguito roll back | Già sincronizzato | Eseguito roll back |
| PGMZ esegue un'operazione di sincronizzazione prima di ritornare al PGMX. | Ancora in sospeso | Sincronizzato da PGMZ | Già sincronizzato | Sincronizzato |
| PGMZ tenta di avviare il controllo sincronizzazione specificando CMTSCOPE(*ACTGRP) dopo l'aggiornamento del file F7. Il tentativo non ha esito positivo poiché le modifiche sono in sospeso utilizzando la definizione sincronizzazione a livello del lavoro. | Ancora in sospeso | Ancora in sospeso | Già sincronizzato | Ancora in sospeso |
| PGMX non avvia il controllo sincronizzazione e non apre i file F3 e F4 con COMMIT(*YES). PGMZ tenta di aprire il file F7 con COMMIT(*YES). | Ancora in sospeso | Non sotto controllo sincronizzazione | Già sincronizzato | Impossibile aprire il file F7 poiché non esiste nessuna definizione sincronizzazione *JOB (non è stata creata da PGMX). |

Come vengono gestiti gli oggetti dal controllo sincronizzazione

Quando un oggetto viene ubicato sotto il controllo sincronizzazione, diventa una risorsa sincronizzabile. Viene registrato con la definizione sincronizzazione. Partecipa ad ogni operazione di commit e rollback che si verifica per quella definizione sincronizzazione.

I seguenti argomenti descrivono gli attributi di una risorsa sincronizzabile:

- Tipo di risorsa
- Ubicazione
- Protocollo sincronizzazione
- Possibilità di accesso

I seguenti collegamenti contengono ulteriori informazioni relative alle risorse sotto il controllo sincronizzazione:

- Tipi di risorse sincronizzabili
- Risorse locali e remote sincronizzabili
- Possibilità di accesso di una risorsa sincronizzabile
- Protocollo sincronizzazione di una risorsa sincronizzabile
- File registrati su giornale e controllo sincronizzazione
- Sequenza di voci di giornale sotto il controllo sincronizzazione
- Identificativo ciclo sincronizzazione
- Blocco record

Tipi di risorse sincronizzabili

Nella seguente tabella vengono descritti:

- I tipi di risorse sincronizzabili.
- Come sono ubicate sotto il controllo sincronizzazione.
- Come vengono eliminate dal controllo sincronizzazione.
- Limitazioni applicate al tipo di risorsa.

| Tipo di risorsa | Come ubicarla sotto il controllo sincronizzazione | Come eliminarla dal controllo sincronizzazione | Quali tipi di modifiche sono sincronizzabili | Limitazioni |
|--|---|---|--|--|
| FILE- file database locale | Aprire sotto il controllo sincronizzazione ¹ | Chiudere il file, se non esistono modifiche in sospeso. Se le modifiche sono in sospeso quando il file è chiuso, eseguire la successiva operazione di commit o rollback. | Modifiche livello record | Non è possibile bloccare più di 500 000 000 di record per una singola transazione ² . |
| DDL- modifiche a livello oggetto su tabelle SQL locali e raccolte SQL. | Eseguire SQL sotto il controllo sincronizzazione | Eseguire un'operazione di commit o rollback dopo una modifica a livello oggetto. | Modifiche livello oggetto, ad esempio: <ul style="list-style-type: none"> • Creare pacchetto SQL • Creare tabella SQL • Rilasciare la tabella SQL | Solo le modifiche a livello oggetto effettuate tramite SQL si trovano sotto il controllo sincronizzazione. |
| DDM- file DDM (distributed data management) remoto | Aprire sotto il controllo sincronizzazione. In Supporto controllo sincronizzazione per DDM è possibile trovare ulteriori informazioni sul controllo sincronizzazione e sulla DDM (distributed data management). | Chiudere il file, se non esistono modifiche in sospeso. Se le modifiche sono in sospeso quando il file è chiuso, eseguire la successiva operazione di commit o rollback. | Modifiche livello record | |
| LU 6.2- conversazione protetta | Avviare la conversazione ³ | Terminare la conversazione | | |

| Tipo di risorsa | Come ubicarla sotto il controllo sincronizzazione | Come eliminarla dal controllo sincronizzazione | Quali tipi di modifiche sono sincronizzabili | Limitazioni |
|---|---|--|---|--|
| DRDA ^(R) - RDB (relational database) distribuito | Utilizzare istruzione SQL CONNECT | Chiudere collegamento | | |
| API- risorsa sincronizzazione API locale | Aggiungere API QTNADDCR (Risorsa sincronizzazione) | Eliminare API QTNRMVCR (Risorsa sincronizzazione) | Viene stabilito dal programma utente. Le voci giornale possono essere scritte dal programma utente tramite l'API QJOSJRNE (Invio voce giornale) per fornire l'assistenza a queste modifiche tramite la traccia. | L'applicazione deve fornire un programma di uscita da richiamare durante le operazioni di commit, rollback o risincronizzazione. |
| Collegamento TCP-TCP/IP | Utilizzare l'istruzione SQL CONNECT su un RDB definito per usare i collegamenti TCP/IP o aprire un file DDM definito con un'ubicazione TCP/IP | Chiudere il collegamento SQL o chiudere il file DDM se non esistono modifiche in sospeso. Se il file DDM viene chiuso con delle modifiche in sospeso, il collegamento verrà chiuso in seguito alla successiva operazione di commit o rollback. | | |

Note:

¹Per i dettagli relativi a come ubicare un file database sotto il controllo sincronizzazione, fare riferimento all'appropriato manuale di riferimento del linguaggio. Le Informazioni correlate al controllo sincronizzazione si collegano ai manuali del linguaggio che è possibile utilizzare.

²E' possibile utilizzare un file QAQQINI per ridurre il limite di 500 000 000. Per le istruzioni, vedere Gestire la dimensione della transazione.

³Quando viene avviato un collegamento DDM, il file DDM specifica PTCCNV(*YES) e viene definito con un'ubicazione SNA remota, una risorsa LU6.2 viene aggiunta alla risorsa DDM.

Quando viene avviato un collegamento DRDA, una risorsa LU6.2 viene aggiunta alla risorsa DRDA se si verificano entrambe le seguenti condizioni:

- Il programma sta utilizzando i protocolli collegamento DUW (distributed unit of work).
- Viene eseguito il collegamento ad un RDB definito con un'ubicazione SNA remota. Per ulteriori informazioni sull'avvio delle conversazioni protette, vedere Programmazione APPC



Risorse sincronizzabili locali e remote

Una risorsa sincronizzabile può essere locale o remota.

Risorsa locale sincronizzabile

Una risorsa locale sincronizzabile risiede sullo stesso sistema dell'applicazione. Ogni giornale associato alle risorse sotto il controllo sincronizzazione può essere considerato come una ubicazione locale. Tutte le

risorse registrate senza un giornale (facoltativo sia per le risorse DDL sia per le risorse API) possono essere considerate come una ubicazione locale distinta.

Se una risorsa sincronizzabile risiede su un lotto dischi indipendente, la risorsa non viene considerata locale nel caso in cui la definizione sincronizzazione si trovi su un lotto dischi differente. Per ulteriori informazioni sulle risorse sincronizzabili e i lotti dischi indipendenti, vedere Controllo sincronizzazione e lotti dischi indipendenti.

Risorse remote sincronizzabili

Una risorsa remota sincronizzabile risiede su un sistema differente dall'applicazione. Per ogni conversazione univoca su un sistema remoto esiste un'ubicazione remota. Una definizione sincronizzazione può disporre di una o più ubicazioni remote su uno o più sistemi remoti.

Quando una risorsa locale viene posta sotto il controllo sincronizzazione per il lotto dischi di sistema o qualsiasi lotto dischi indipendente, è necessario utilizzare DRDA per accedere alle risorse sotto il controllo sincronizzazione in qualsiasi altro lotto dischi indipendente.

Quanto segue indica i tipi di risorse sincronizzabili e le relative ubicazioni:

| Tipo di risorsa | Ubicazione |
|-----------------|-----------------|
| FILE | Locale |
| DDL | Locale |
| API | Locale |
| DDM | Remota |
| LU62 | Remota |
| DRDA | Locale o remota |
| TCP | Remota |

Possibilità di accesso di una risorsa sincronizzabile

Quando una risorsa è ubicata sotto il controllo sincronizzazione, il gestore risorse indica il modo in cui sarà possibile accedere alla risorsa:

- Aggiornamento
- Sola lettura
- Indeterminato

La possibilità di accesso determina il modo in cui le risorse partecipano ad una transazione. La seguente tabella mostra le possibilità di accesso per un tipo particolare di risorsa e il modo in cui il sistema determina tali possibilità quando la risorsa viene registrata:

| Tipo di risorsa | Possibilità di accesso | Come vengono determinate le possibilità di accesso |
|-----------------|-----------------------------|--|
| FILE | Aggiornamento, sola lettura | In base al modo in cui sono stati aperti i file |
| DDL | Aggiornamento | Sempre con l'aggiornamento |
| API | Aggiornamento | Sempre con l'aggiornamento |
| DDM | Aggiornamento, sola lettura | In base al modo in cui sono stati aperti i file |
| LU62 | Indeterminato | Sempre indeterminato |

| Tipo di risorsa | Possibilità di accesso | Come vengono determinate le possibilità di accesso |
|-----------------|--|---|
| DRDA | Aggiornamento, sola lettura, indeterminato | Per DRDA Livello 1, la possibilità di accesso è l'aggiornamento se non vengono registrate altre risorse remote. In caso contrario, la possibilità di accesso è di sola lettura. Per DRDA Livello 2, la possibilità di accesso è sempre indeterminata. |
| TCP | Indeterminato | Sempre indeterminato |

Le possibilità di accesso delle risorse già registrate determinano se è possibile registrare una nuova risorsa. Le seguenti regole si riferiscono a:

- Una risorsa ad una fase la cui possibilità di accesso è l'aggiornamento, non può essere registrata in uno dei seguenti casi:
 - Le risorse la cui possibilità di accesso è l'aggiornamento, sono già registrate su altre ubicazioni.
 - Le risorse la cui possibilità di accesso è indeterminata, sono già registrate su altre ubicazioni.
 - Le risorse la cui possibilità di accesso è indeterminata, sono già registrate sulla stessa ubicazione e le risorse sono state modificate durante la transazione corrente.
- Una risorsa a due fasi la cui possibilità di accesso è l'aggiornamento, non può essere registrata quando una risorsa ad una fase, la cui possibilità di accesso è l'aggiornamento, è già registrata.

Protocollo sincronizzazione di una risorsa sincronizzabile

Il **Protocollo sincronizzazione** è la capacità di una risorsa di partecipare ad un'elaborazione di sincronizzazione ad una o a due fasi. Le risorse locali, ad eccezione delle risorse sincronizzabili API, sono sempre risorse a due fasi.

Se una risorsa sincronizzabile risiede su un lotto dischi indipendente, la risorsa non viene considerata locale o a due fasi se la definizione sincronizzazione risiede su un lotto dischi differente. Per ulteriori informazioni sulle risorse sincronizzabili e i lotti dischi indipendenti, vedere Controllo sincronizzazione e lotti dischi indipendenti.

Una risorsa a due fasi viene anche denominata **risorsa protetta**. Le risorse remote e le risorse sincronizzabili API devono essere registrate come risorse a una fase o a due fasi quando vengono ubicate sotto il controllo sincronizzazione. Nella seguente tabella vengono visualizzati i tipi di risorse sincronizzabili che possono coesistere in una definizione sincronizzazione con una risorsa ad una fase:

| Tipo di risorsa | Può coesistere con |
|----------------------------|--|
| Risorsa API ad una fase | Altre risorse locali. Non con risorse remote. |
| Risorsa remota ad una fase | Altre risorse ad una fase sulla stessa ubicazione. Non con risorse locali. |

File registrati su giornale e controllo sincronizzazione

E' necessario giornalizzare (registrare) un file database (tipo di risorsa FILE o DDM) prima di poterlo aprire per l'emissione sotto controllo sincronizzazione o prima che possa essere preso come riferimento da un'applicazione SQL che utilizza un livello di isolamento diverso da Non sincronizzare. Non è necessario registrare su giornale un file per aprirlo per l'immissione solo sotto il controllo sincronizzazione. Si verifica un errore se:

- Viene effettuato un tentativo di aprire un file database per l'emissione sotto il controllo sincronizzazione, ma il file non viene registrato correntemente su giornale.
- Non viene avviata una definizione sincronizzazione che può essere utilizzata dal file aperto sotto il controllo sincronizzazione.

Se vengono registrate su giornale solo le immagini successive di un file database quando tale file viene aperto sotto il controllo sincronizzazione, la registrazione su giornale delle immagini precedenti e successive viene avviata automaticamente dal sistema. Le immagini precedenti vengono scritte solo per le modifiche al file che si verificano sotto il controllo sincronizzazione. Se si verificano, nello stesso momento, altre modifiche al file che non si trovano sotto il controllo sincronizzazione, vengono scritte solo le immagini-successive per queste modifiche.

Il sistema scrive automaticamente sul giornale le modifiche sincronizzabili a livello record e a livello oggetto. Per le modifiche a livello record, il sistema utilizza le voci di giornale, se necessario, per scopi di ripristino; il sistema non utilizza le voci dalle modifiche sincronizzabili a livello oggetto per scopi di ripristino. Inoltre, il sistema non scrive automaticamente le voci di giornale per le risorse sincronizzabili API. Tuttavia, il programma di uscita per la risorsa API può utilizzare l'API QJOSJRNE (Invio voce di giornale) per scrivere le voci di giornale per fornire una traccia di controllo o per facilitare il ripristino. Il contenuto di queste voci viene controllato dal programma di uscita dell'utente.

Il sistema utilizza un meccanismo diverso da un giornale per eseguire il ripristino delle risorse sincronizzazione a livello oggetto. Il ripristino delle risorse sincronizzazione API viene eseguito richiamando il programma di uscita commit e rollback associato ad ogni particolare risorsa sincronizzazione API. Il programma di uscita è responsabile per l'esecuzione del ripristino attuale necessario per tale situazione.

Per ulteriori informazioni relative alla registrazione su giornale, vedere l'argomento Gestione giornali.

Sequenza di voci giornale sotto il controllo sincronizzazione

La seguente tabella descrive la sequenza di voci che solitamente vengono scritte mentre è attiva una definizione sincronizzazione. E' possibile utilizzare il Programma di ricerca codice di giornale per ottenere ulteriori informazioni sui contenuti delle voci di giornale.

Le voci del controllo sincronizzazione sono scritte su un giornale (locale o remoto) solo se si verifica almeno una delle seguenti situazioni:

- Il giornale viene specificato come giornale predefinito sul comando Avvio controllo sincronia (STRCMTCTL).
- Almeno un file assegnato al giornale viene aperto sotto il controllo sincronizzazione.
- Almeno una risorsa sincronizzazione API associata al giornale viene registrata sotto il controllo sincronizzazione.

| Tipo di voce | Descrizione | Dove è stato scritto | Quando è stato scritto |
|--------------|-----------------------------------|--|---|
| C BC | Inizio controllo sincronizzazione | Sul giornale predefinito, se ne viene specificato uno sul comando STRCMTCTL. | Quando viene emesso il comando STRCMTCTL |
| | | Sul giornale di ogni ubicazione locale. | Quando il primo file assegnato ad un giornale viene aperto o quando viene registrata una risorsa API per un giornale. |
| C SC | Avvio ciclo sincronizzazione | Sul giornale di ogni ubicazione locale. | Quando viene eseguita la prima modifica record per la transazione di un file assegnato a questo giornale ¹ . |
| | | Sul giornale di una risorsa API. | Quando l'API QJOSJRNE è la prima utilizzata tramite il tasto <i>Inclusione identificativo ciclo sincronizzazione</i> . |

| Tipo di voce | Descrizione | Dove è stato scritto | Quando è stato scritto |
|--------------------------|---------------------------------|--|--|
| Codici di giornale D e F | Voci livello oggetto DDL | Sul giornale associato all'oggetto da aggiornare. Solo le voci di giornale che contengono un identificativo ciclo sincronizzazione rappresentano una modifica livello oggetto DDL che fa parte della transazione. | Quando vengono eseguiti gli aggiornamenti. |
| Codice di giornale R | Voci livello record | Sul giornale associato al file da aggiornare. | Quando vengono eseguiti gli aggiornamenti. |
| Codice di giornale U | Voci create dall'utente | Sul giornale associato ad una risorsa API. | Se il programma dell'applicazione utilizza il comando SNDJRNE o l'API QJOSJRNE. |
| C CM | Commit | Sul giornale di ogni ubicazione. | Quando la sincronizzazione è stata completata con esito positivo. |
| | | Sul giornale predefinito. | Se alcune risorse sincronizzabili vengono associate al giornale. |
| C RB | Rollback | Sul giornale di ogni ubicazione locale. | Dopo il completamento dell'operazione di rollback. |
| | | Sul giornale predefinito. | Se alcune risorse sincronizzabili vengono associate al giornale. |
| C LW | Fine transazione | Sul giornale predefinito, se ne viene specificato uno sul comando STRCMTCTL. Il sistema scrive un record dell'intestazione LW e uno o più record dei dettagli. Queste voci vengono scritte solo se viene specificato OMTJRNE(*NONE) sul comando STRCMTCTL o se si verifica un errore di sistema. | Dopo il completamento dell'operazione di commit o rollback. |
| C EC | Fine controllo sincronizzazione | Sul giornale di ogni ubicazione locale. | Quando viene completato il comando ENDCMTCTL (Fine controllo sincronizzazione). |
| | | Su un giornale locale che non è il giornale predefinito. | Quando viene stabilito il limite di sincronizzazione, in seguito al punto in cui tutte le risorse sincronizzabili associate al giornale sono state eliminate dal controllo sincronizzazione. |

| Tipo di voce | Descrizione | Dove è stato scritto | Quando è stato scritto |
|--|-------------|----------------------|------------------------|
| <p>Nota: ¹E' possibile specificare che la parte di lunghezza-fissa della voce di giornale include le informazioni relative alla transazione, specificando il valore *LUW (Unità logica di lavoro) per il parametro FIXLENDTA (Dati lunghezza-fissa) del comando CRTJRN (Creazione giornale) o CHGJRN (Modifica giornale). Specificando il parametro *LUW (FIXLENDTA), la parte di lunghezza-fissa di ogni voce giornale C SC conterrà il LUWID (ID unità logica di lavoro) della transazione corrente. Allo stesso modo delle transazioni XA, se viene specificato il parametro *LUW (FIXLENDTA), la parte di lunghezza-fissa di ogni voce giornale C SC conterrà l'XID della transazione corrente. Tramite il LUWID o XID è possibile trovare tutti i cicli sincronizzazione per una transazione particolare se più giornali o sistemi sono coinvolti nella transazione.</p> | | | |

Identificativo ciclo sincronizzazione

Un **ciclo sincronizzazione** è il tempo che intercorre tra un limite sincronizzazione e il successivo. Il sistema assegna un **identificativo ciclo sincronizzazione** per associare tutte le voci giornale per un particolare ciclo sincronizzazione. Ogni giornale che partecipa ad una transazione possiede il relativo ciclo sincronizzazione e il relativo identificativo ciclo sincronizzazione.

L'identificativo ciclo sincronizzazione è il numero di sequenza giornali della voce giornale C SC scritta per il ciclo sincronizzazione. L'identificativo ciclo sincronizzazione è ubicato in ogni voce giornale scritta durante il ciclo sincronizzazione. Se più di un giornale viene utilizzato durante il ciclo sincronizzazione, l'identificativo ciclo sincronizzazione per ogni giornale è differente.

E' possibile specificare che la parte di lunghezza-fissa della voce giornale include le informazioni relative alla transazione, specificando il valore *LUW (unità logica di lavoro) per il parametro FIXLENDTA (dati lunghezza-fissa) del comando CRTJRN (Creazione giornale) o CHGJRN (Modifica giornale). Specificando il parametro *LUW (FIXLENDTA), la parte di lunghezza-fissa di ogni voce giornale C SC conterrà il LUWID (ID unità logica di lavoro) della transazione corrente. Allo stesso modo delle transazioni XA, se viene specificato il parametro *LUW (FIXLENDTA), la parte di lunghezza-fissa di ogni voce giornale C SC conterrà l'XID della transazione corrente. Tramite il LUWID o XID è possibile trovare tutti i cicli sincronizzazione per una transazione particolare se più giornali o sistemi sono coinvolti nella transazione.

E' possibile utilizzare l'API QJOSJRNE (Invio voce giornale) per scrivere le voci giornale per le risorse API. E' disponibile l'opzione per includere l'identificativo ciclo sincronizzazione su queste voci giornale.

E' possibile utilizzare l'identificativo ciclo sincronizzazione per applicare o eliminare le modifiche registrate su giornale apportate ad un limite sincronizzazione, tramite il comando Applicazione modifiche su giornale (APYJRNCHG) o il comando Eliminazione modifiche su giornale (RMVJRNCHG). Queste limitazioni vengono applicate a:

- La maggior parte delle modifiche a livello oggetto apportate sotto il controllo sincronizzazione vengono scritte sul giornale ma non vengono applicate o eliminate tramite i comandi APYJRNCHG e RMVJRNCHG.
- L'API QJOSJRNE scrive le voci giornale create dall'utente con un codice giornale di U. Tali voci non possono essere applicate o eliminate tramite i comandi APYJRNCHG e RMVJRNCHG. E' necessario applicarle ed eliminarle con un programma scritto dall'utente.

Blocco record

Quando un lavoro possiede un blocco record e un altro lavoro tenta di richiamare tale record per l'aggiornamento, il lavoro richiedente attende e viene eliminato dall'elaborazione attiva fino a quando si verifica una delle seguenti situazioni:

- Il blocco record viene rilasciato.
- Il tempo di attesa specificato termina.

Più di un lavoro può richiedere che un record venga bloccato da un altro lavoro. Quando il blocco record viene rilasciato, il primo lavoro per richiedere il record riceve tale record. Durante l'attesa di un record bloccato, specificare il tempo di attesa nel parametro WAITRCD sui seguenti comandi di creazione, modifica o sostituzione:

- CRTPF (Creazione file fisico)
- CRTLF (Creazione file logico)
- CRTSRCPF (Creazione file fisico origine)
- CHGPF (Modifica file fisico)
- CHGLF (Modifica file logico)
- CHGSRC PF (Modifica file fisico origine)
- OVRDBF (Sostituzione file database)

Quando si specifica il tempo di attesa, considerare quanto segue:

- Se non viene specificato un valore, il programma attende il tempo di attesa predefinito per il processo.
- Per le definizioni sincronizzazione con blocchi solo nell'ambito della transazione, il tempo di attesa predefinito del lavoro può essere sostituito con un tempo di attesa blocco transazione che può essere specificato su:
 - L'API xa_open.
 - Un'interfaccia JDBC o JTA. In Transazioni distribuite vengono elencate queste API.
- Se il record non può essere assegnato entro il tempo specificato, un messaggio di notifica viene inviato al programma di linguaggio ad alto livello.
- Se viene superato il tempo di attesa per un record, il messaggio inviato alla registrazione lavori fornisce il nome al lavoro che possiede il record bloccato che ha causato l'attesa del lavoro richiedente. Se si verificano errori di blocco record, è possibile utilizzare la registrazione lavori per determinare quali programmi alterare in modo tale che non conservino i blocchi per periodi di tempo molto lunghi.

I programmi conservano i blocchi record per lunghi periodi per una delle seguenti ragioni:

- Il record rimane bloccato mentre l'utente stazione di lavoro sta analizzando una modifica.
- Il blocco record fa parte di una lunga transazione sincronizzazione. Effettuare transazioni più piccole in modo tale che sia possibile eseguire più frequentemente un'operazione di sincronizzazione.
- Si è verificato un blocco non desiderato. Ad esempio, supporre che un file venga definito come un file di aggiornamento con chiavi univoche e che il programma aggiorni e aggiunga ulteriori record al file. Se l'utente stazione di lavoro desidera aggiornare un record sul file, il programma può tentare di accedere al record per stabilire se la chiave è già esistente. In questo caso, il programma informa l'utente stazione di lavoro che la richiesta effettuata non è valida. Quando il record viene richiamato da un file, viene bloccato fino a quando non viene implicitamente rilasciato da un'altra operazione di lettura sullo stesso file o fino a quando non viene rilasciato esplicitamente.

Nota:

Per ulteriori informazioni su come utilizzare ogni interfaccia di linguaggio ad alto livello per rilasciare i blocchi record, vedere l'adeguato manuale di riferimento del linguaggio ad alto livello. In Informazioni correlate al controllo sincronizzazione sono presenti i collegamenti ai manuali di linguaggio ad alto livello che è possibile utilizzare con il controllo sincronizzazione.

La durata di un blocco è maggiore se viene specificato LCKLVL(*ALL), poiché il record richiamato dal file rimane bloccato fino alla successiva operazione di commit o rollback. Non viene rilasciato implicitamente da un'altra operazione di lettura e non può essere rilasciato esplicitamente.

Un'altra funzione tramite la quale è possibile assegnare un blocco ad un file è la funzione salva-mentre-attivo. L'argomento Salvare il server mentre è attivo contiene ulteriori informazioni sulla funzione salva-mentre-attivo.

Controllo sincronizzazione e lotti dischi indipendenti

I lotti dischi indipendenti e i gruppi di lotti dischi indipendenti, possono avere una database SQL OS/400 specifico. E' possibile utilizzare il controllo sincronizzazione con questi database. Tuttavia, poiché ogni lotto dischi indipendente o gruppo di lotti dischi indipendenti possiede un database SQL specifico, è necessario seguire queste considerazioni.

Considerazioni sulle definizioni sincronizzazione

All'avvio del controllo sincronizzazione, la definizione sincronizzazione viene creata nella libreria QRECOVERY. Ogni lotto dischi indipendente o ogni gruppo di lotti dischi indipendente dispone delle relative versioni di una libreria QRECOVERY. Su un lotto dischi indipendente, il nome della libreria QRECOVERY è QRCYxxxxx, dove xxxxx è il numero del lotto dischi indipendente. Ad esempio, il nome della libreria QRECOVERY per il lotto dischi indipendente 39 è QRCY00039. Inoltre, se il lotto dischi indipendente è una parte di un gruppo di lotti dischi, solo il lotto principale possiede una libreria QRCYxxxxx.

All'avvio del controllo sincronizzazione, la definizione sincronizzazione viene creata nella libreria QRECOVERY del lotto dischi indipendente associato al lavoro, rendendo attivo il controllo sincronizzazione sul lotto dischi indipendente.

Di seguito vengono riportate altre considerazioni relative alle definizioni sincronizzazione:

- L'utilizzo del comando SETASPGRP (Impostazione gruppo ASP) mentre il controllo sincronizzazione è attivo su un lotto dischi indipendente, ha i seguenti effetti:
 - Se si passa da un lotto dischi indipendente e le risorse sono registrate con controllo sincronizzazione sul lotto dischi, il comando SETASPGRP causa un errore con messaggio CPDB8EC, codice errore 2, "Il sottoprocesso possiede una transazione non sincronizzata." Questo messaggio è seguito dal messaggio CPF8E9.
 - Se si passa da un lotto dischi indipendente e nessuna risorsa è registrata con controllo sincronizzazione, le definizioni sincronizzazione vengono spostate nel lotto dischi indipendente a cui l'utente è passato.
 - Se si passa dal lotto dischi del sistema (Gruppo ASP *NONE), il controllo sincronizzazione non viene influenzato. Le definizioni sincronizzazione rimangono sul lotto dischi del sistema.
 - Se viene utilizzato un oggetto di notifica, questo deve risiedere sullo stesso lotto dischi indipendente o sul gruppo di lotti dischi indipendente della definizione sincronizzazione. Se la definizione sincronizzazione viene spostata su un altro lotto dischi indipendente o un altro gruppo lotto dischi indipendente, è necessario che l'oggetto di notifica risieda sull'altro lotto o sull'altro gruppo. L'oggetto di notifica sull'altro lotto dischi indipendente o gruppo di lotti dischi indipendente viene aggiornato se la definizione sincronizzazione termina in modo anomalo. Se l'oggetto di notifica non viene trovato sull'altro lotto dischi indipendente o gruppo di lotti dischi indipendente, l'aggiornamento non avrà esito positivo e causerà un messaggio CPF8358.
- Il ripristino delle definizioni sincronizzazione che risiedono su un lotto dischi indipendente viene eseguito durante l'elaborazione dell'attivazione del lotto dischi indipendente ed è simile al ripristino IPL.
- Le definizioni sincronizzazione in un lotto dischi indipendente non vengono ripristinate durante l'IPL del sistema.
- La disattivazione di un lotto dischi indipendente ha i seguenti effetti sulle definizioni sincronizzazione:
 - I lavori associati al lotto dischi indipendente terminano.
 - Nessuna nuova definizione sincronizzazione può essere creata sul lotto dischi indipendente.
 - Le definizioni sincronizzazione che risiedono sul lotto dischi indipendente diventano inutilizzabili.

- Le definizioni sincronizzazione che risiedono sul lotto dischi indipendente, ma non collegate ad un lavoro, rilasciano i blocchi nell'ambito della transazione.
- Non è possibile utilizzare un collegamento SNA LU6.2 (conversazioni protette o DUW (Distributed Unit of Work)) per collegarsi ad un database remoto da un database lotto dischi indipendente. E' possibile utilizzare conversazioni SNA non protette per collegarsi da una database lotto dischi indipendente ad un database remoto.

Quando il controllo sincronizzazione è attivo per un lavoro o un sottoprocesso, l'accesso ai dati al di fuori del lotto dischi indipendente o del gruppo lotto dischi a cui appartiene la definizione sincronizzazione è possibile solo in modalità remota, come se i dati risiedessero su un altro sistema. Quando viene emessa un'istruzione CONNECT SQL per collegarsi al RDB (relational database) sul lotto dischi indipendente, il sistema rende remoto il collegamento.

Il lotto dischi del sistema e i lotti dischi di base non richiedono un collegamento remoto per l'accesso di sola lettura ai dati che risiedono su un lotto dischi indipendente. Allo stesso modo, un lotto dischi indipendente non richiede un collegamento remoto per l'accesso di sola lettura ai dati che risiedono sul lotto dischi del sistema o su un lotto dischi di base.

Considerazioni relative alle transazioni XA

Nell'ambiente XA, ogni database è considerato un gestore risorse specifico. Quando un gestore transazioni desidera accedere a due database sotto la stessa transazione, deve utilizzare i protocolli XA per eseguire la sincronizzazione a due fasi con i due gestori risorse.

Poiché ogni lotto dischi indipendente è un database SQL specifico, nell'ambiente XA ogni lotto dischi indipendente viene considerato anche un gestore risorse specifico. Per un server delle applicazioni che deve eseguire una transazione indirizzata a due differenti lotti dischi indipendenti, il gestore transazioni deve utilizzare un protocollo di sincronizzazione a due fasi.

Per ulteriori informazioni sui lotti dischi indipendenti, vedere l'argomento Lotti dischi indipendenti.

Considerazioni e limitazioni relative al controllo sincronizzazione

Di seguito vengono riportate varie considerazioni e limitazioni relative al controllo sincronizzazione:

Considerazioni file database

- Se si specifica che un file condiviso deve essere aperto sotto il controllo sincronizzazione, tutti gli utilizzi successivi di tale file devono essere aperti sotto il controllo sincronizzazione.
- Se SEQONLY(*YES) viene specificato per il file aperto per la sola lettura tramite LCKLVL(*ALL) (implicitamente o tramite un programma di linguaggio ad alto livello o esplicitamente dal comando Sostituzione con file database (OVRDBF)), allora SEQONLY(*YES) viene ignorato e viene utilizzato SEQONLY(*NO).
- Le modifiche a livello record effettuate sotto il controllo sincronizzazione vengono registrate in un giornale. Queste modifiche possono essere applicate o eliminate dal database con il comando Applicazione modifiche su giornale (APYJRNCHG) oppure con il comando Eliminazione modifiche su giornale (RMVJRNCHG).
- Sia le immagini precedenti sia le immagini successive dei file vengono registrate su giornale sotto il controllo sincronizzazione. Se si specifica solo di registrare su giornale le immagini successive dei file, il sistema registra automaticamente su giornale le immagini precedenti delle modifiche del file verificatesi sotto il controllo sincronizzazione. Tuttavia, poiché le immagini precedenti non vengono catturate per tutte le modifiche apportate ai file, non è possibile utilizzare il comando RMVJRNCHG per questi file.

Considerazioni relative alle modifiche livello oggetto e record

- La maggior parte delle modifiche del livello oggetto apportate sotto il controllo sincronizzazione vengono scritte sul giornale ma non vengono applicate o eliminate tramite i comandi APYJRNCHG e

RMVJRNG. Tuttavia, è possibile utilizzare l'API QJOSJRNE (Invio voce di giornale) per inviare le voci di giornale per altri eventi. Durante il ripristino, è possibile richiamare queste voci ed elaborarle con un programma scritto dall'utente.

- Le modifiche del livello oggetto e del livello record effettuate sotto il controllo sincronizzazione tramite SQL, utilizzeranno la definizione sincronizzazione attualmente attiva per il gruppo attivazione in cui è in esecuzione il programma richiedente. Se non è attiva né la definizione sincronizzazione a livello del lavoro né quella alivello del gruppo attivazione, SQL avvierà una definizione sincronizzazione alivello del gruppo attivazione. Per ulteriori informazioni relative alle modifiche effettuate sotto il controllo sincronizzazione tramite SQL, vedere DB2 UDB for iSeries SQL Programming Concepts

Considerazioni relative alla sincronizzazione ad una fase e a due fasi

- Quando viene stabilita una conversazione o un collegamento remoto a una fase, le conversazioni e i collegamenti remoti ad altre ubicazioni non sono consentiti. Se viene stabilito un limite di sincronizzazione e vengono eliminate tutte le risorse, l'ubicazione può essere modificata.
- Se si sta utilizzando una sincronizzazione a due fasi, non è necessario utilizzare il comando Immissione comando remoto (SBMRMTCMD) per avviare il controllo sincronizzazione o per eseguire qualsiasi altra operazione di controllo sincronizzazione sulle ubicazioni remote. Il sistema esegue queste funzioni automaticamente.
- Per un'ubicazione remota ad una fase, i comandi CL COMMIT e ROLLBACK non funzioneranno se l'SQL si trova nello stack di chiamata e l'RDB (relational database) remoto si trova su un sistema. Se SQL non si trova sullo stack di chiamata, i comandi COMMIT e ROLLBACK funzioneranno.
- Per un'ubicazione remota ad una fase, è necessario avviare il controllo sincronizzazione sul sistema sorgente prima di effettuare le modifiche sincronizzabili sulle risorse remote. Il sistema avvia automaticamente il controllo sincronizzazione per l'SQL database distribuito sul sistema sorgente nell'ora del collegamento se il programma SQL è in esecuzione con un'opzione del controllo sincronizzazione diversa da *NONE. Quando la prima risorsa remota viene posta sotto il controllo sincronizzazione, il sistema avvia il controllo sincronizzazione sul sistema di destinazione.

Salvare le considerazioni

Non è possibile effettuare un'operazione di salvataggio se il lavoro che esegue il salvataggio possiede una o più definizioni sincronizzazione attive con uno dei seguenti tipi di modifiche sincronizzabili:

- Una modifica di record su un file che risiede nella libreria salvata. Per i file logici, vengono controllati tutti i file fisici correlati.
- Qualsiasi modifica a livello oggetto all'interno di una libreria che viene salvata.
- Qualsiasi risorsa API aggiunta tramite l'API QTNADDCR (Aggiunta risorsa sincronizzazione) e con il campo Consentire elaborazione salvataggio normale impostato sul valore predefinito di N.

Questo impedisce il salvataggio delle modifiche del supporto magnetico di salvataggio dovute ad una transazione parziale.

I blocchi oggetto e record impediscono che le modifiche in sospeso dalle definizioni sincronizzazione in altri lavori, vengano salvate sul supporto magnetico di salvataggio. Questo è valido solo per le risorse sincronizzazione API se i blocchi vengono acquisiti quando le modifiche vengono apportate sull'oggetto o sugli oggetti associati alla risorsa sincronizzazione API.

Considerazioni e limitazioni varie

- Prima di potenziare il sistema con un nuovo rilascio, tutte le sincronizzazioni in sospeso devono essere completate o annullate. Per ulteriori dettagli, vedere l'argomento Accertarsi dell'integrità sincronizzazione a due fasi prima di installare il software.
- I valori COMMIT e ROLLBACK vengono visualizzati nel campo Funzione WRKACTJOB durante un commit o un rollback. Se la Funzione rimane COMMIT o ROLLBACK per lungo tempo, potrebbe verificarsi una delle seguenti situazioni:

- Un errore risorsa durante il commit o il rollback richiede la risincronizzazione. Il controllo non ritornerà all'applicazione fino a quando la risincronizzazione non verrà completata o annullata.
- Questo sistema ha fornito un'indicazione di sola lettura durante la sincronizzazione. Il controllo non ritornerà all'applicazione fino a quando il sistema che ha avviato la sincronizzazione invierà dati a questo sistema.
- Il sistema ha fornito un'indicazione OK per uscire durante la sincronizzazione. Il controllo non ritornerà all'applicazione fino a quando il sistema che ha avviato la sincronizzazione invierà dati a questo sistema.

Controllo sincronizzazione per applicazioni batch

Le applicazioni batch possono richiedere o meno il controllo sincronizzazione. In alcuni casi, un'applicazione batch può eseguire una singola funzione di lettura su un file di immissione e una funzione di aggiornamento di un file principale. Tuttavia, se è necessario effettuare il riavvio in seguito ad una fine anomala, è possibile utilizzare il controllo sincronizzazione per questo tipo di applicazione.

Il file di immissione è un file di aggiornamento con un codice nei record per indicare l'elaborazione di un record. Questo e qualsiasi altro file aggiornato viene posto sotto il controllo sincronizzazione. Quando il codice si trova nel file di immissione, rappresenta una transazione completata. Il programma effettua la lettura attraverso il file di immissione e inoltra eventuali record con codice completo. Questo consente l'utilizzo della stessa logica di programmazione per condizioni normali e di riavvio.

Se l'applicazione batch contiene i record di immissione dipendenti l'uno dall'altro e contiene commutatori o totali, è possibile utilizzare un oggetto di notifica per fornire informazioni sul riavvio. I valori contenuti nell'oggetto di notifica vengono utilizzati per riavviare l'elaborazione dall'ultima transazione sincronizzata nel file di immissione.

Se i record di immissione dipendono l'uno dall'altro, possono essere elaborati come una transazione. Un lavoro batch può bloccare un massimo di 500 000 000 di record. E' possibile ridurre questo limite tramite un QAQQINI (File opzioni interrogazioni). Utilizzare il parametro QRYOPTLIB del comando CHGQRYA (Modifica attributi query) per specificare un File opzioni interrogazioni per un lavoro da utilizzare. Utilizzare il valore COMMITMENT_CONTROL_LOCK_LEVEL nel File opzioni interrogazioni come limite blocco per il lavoro.

Qualsiasi ciclo di sincronizzazione che superi 2000 blocchi probabilmente rallenterà le prestazioni del sistema. In caso contrario, esistono le stesse considerazioni di blocco delle applicazioni interattive, ma il periodo di tempo durante il quale i record vengono bloccati in un'applicazione batch, è meno rilevante di quello delle applicazioni interattive.

Controllo sincronizzazione a due fasi

Il controllo sincronizzazione a due fasi garantisce che tutte le risorse sincronizzabili presenti su più sistemi rimangano sincronizzate. OS/400 supporta la sincronizzazione a due fasi in conformità con la struttura SNA LU 6.2. Per informazioni più dettagliate sui protocolli interni utilizzati dal sistema per la sincronizzazione a due fasi, fare riferimento a *SNA Transaction Programmer's Reference for LU Type 6.2, GC30-3084-05*. Tutti i rilasci supportati di OS/400 supportano i protocolli Presumed Nothing di SNA LU 6.2 e i protocolli Presumed Abort di SNA LU 6.2.

La sincronizzazione a due fasi viene, inoltre, supportata utilizzando TCP/IP come un protocollo DUW (Distributed Unit of Work) DRDA. Per utilizzare i collegamenti DUW TCP/IP, tutti i sistemi (sia il richiedente dell'applicazione sia il server dell'applicazione) devono essere V5R1M0 o versione più recente. Per ulteriori informazioni relative a DRDA vedere Open Group Technical Standard, *DRDA V2 Vol. 1: Distributed Relational Database Architecture* sul sito Web Open Group



Sotto la sincronizzazione a due fasi, il sistema esegue l'operazione di sincronizzazione in due onde:

- Durante l'**onda di preparazione**, un gestore risorse invia una richiesta di sincronizzazione al gestore transazioni. Il gestore transazioni comunica alle altre risorse che gestisce e agli altri gestori transazioni che la transazione può essere sincronizzata. Tutti i gestori risorse devono rispondere che possono eseguire la sincronizzazione. Ciò viene definito come **indicazione**.
- Durante l'**onda sincronizzata**, il gestore transazioni che avvia la richiesta di sincronizzazione decide cosa fare, in base al risultato dell'onda di preparazione. Se l'onda di preparazione viene completata con esito positivo e tutti i partecipanti indicano di essere pronti, il gestore transazioni comunica a tutte le risorse che gestisce e agli altri gestori transazioni di sincronizzare la transazione. Se l'onda di preparazione non viene completata con esito positivo, a tutti i gestori transazioni e ai gestori risorse viene ordinato di eseguire il rollback della transazione.

Operazioni di commit e rollback con risorse remote

Quando le risorse remote si trovano sotto il controllo sincronizzazione, l'iniziatore invia una richiesta di sincronizzazione a tutti gli agent remoti. La richiesta viene inviata attraverso la rete di programmi di transazione. Ogni agent risponde con i risultati dell'operazione di sincronizzazione.

Se si verificano degli errori durante l'onda di preparazione, l'iniziatore invia una richiesta di rollback a tutti gli agent. Se si verificano degli errori durante l'onda sincronizzata, il sistema tenta di portare il maggior numero possibile di ubicazioni sullo stato sincronizzato. Tali tentativi possono risultare in uno stato euristico misto. Consultare Stati di una transazione per il controllo sincronizzazione a due fasi per ulteriori informazioni sugli stati possibili.

Eventuali errori vengono inviati di nuovo all'iniziatore dove vengono segnalati all'utente. Se è stato specificato un giornale predefinito sul comando Avvio controllo sincronia (STRCMTCTL), vengono scritte le voci C LW. Se si verificano degli errori, queste voci vengono scritte anche se è stato specificato OMTJRNE(*LUWID). E' possibile utilizzare queste voci, insieme con i messaggi di errore e le informazioni sullo stato delle definizioni sincronizzazione, per tentare di sincronizzare manualmente le risorse sincronizzabili.

Quando le risorse remote si trovano sotto il controllo sincronizzazione, l'iniziatore invia una richiesta di rollback a tutti gli agent remoti. La richiesta viene inviata attraverso la rete di programmi di transazione. Ogni agent risponde con i risultati dell'operazione di rollback.

Per ulteriori informazioni vedere quanto segue:

- Ruoli nell'elaborazione della sincronizzazione
- Stati della transazione per il controllo sincronizzazione a due fasi
- Definizioni sincronizzazione per il controllo sincronizzazione a due fasi

Ruoli nell'elaborazione della sincronizzazione

Se la sincronizzazione di una transazione coinvolge più di un gestore risorse, ognuno di essi avrà un ruolo nella transazione. Un gestore risorse è responsabile per il commit e il rollback delle modifiche effettuate durante la transazione. I gestori risorse per tipo di risorsa sono:

FILE

Gestore database

DDM

Gestore database

DDL

Gestore database

DRDA

Programma di transazione comunicazioni

LU62

Programma di transazione comunicazioni

API

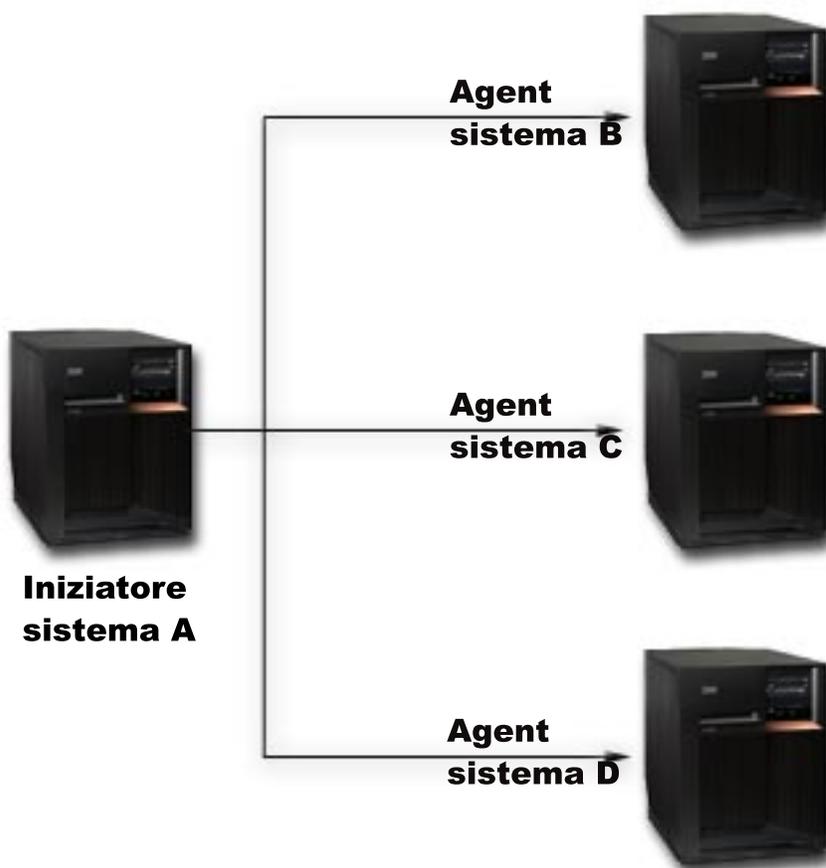
Programma di uscita API

Le seguenti figure descrivono i ruoli di base in una transazione. La struttura visualizzata nelle figure viene definita una **rete di programmi di transazione**. La struttura può trovarsi in un albero a livello unico e un albero a più livelli.

Ruoli nell'elaborazione della sincronizzazione a due fasi: albero a livello unico

Quando un'applicazione sul Sistema A emette una richiesta di sincronizzazione, il gestore risorse sul Sistema A diventa l'**iniziatore**. Per la DUW (distributed unit of work) DRDA su TCP/IP, l'iniziatore viene definito **coordinatore**.

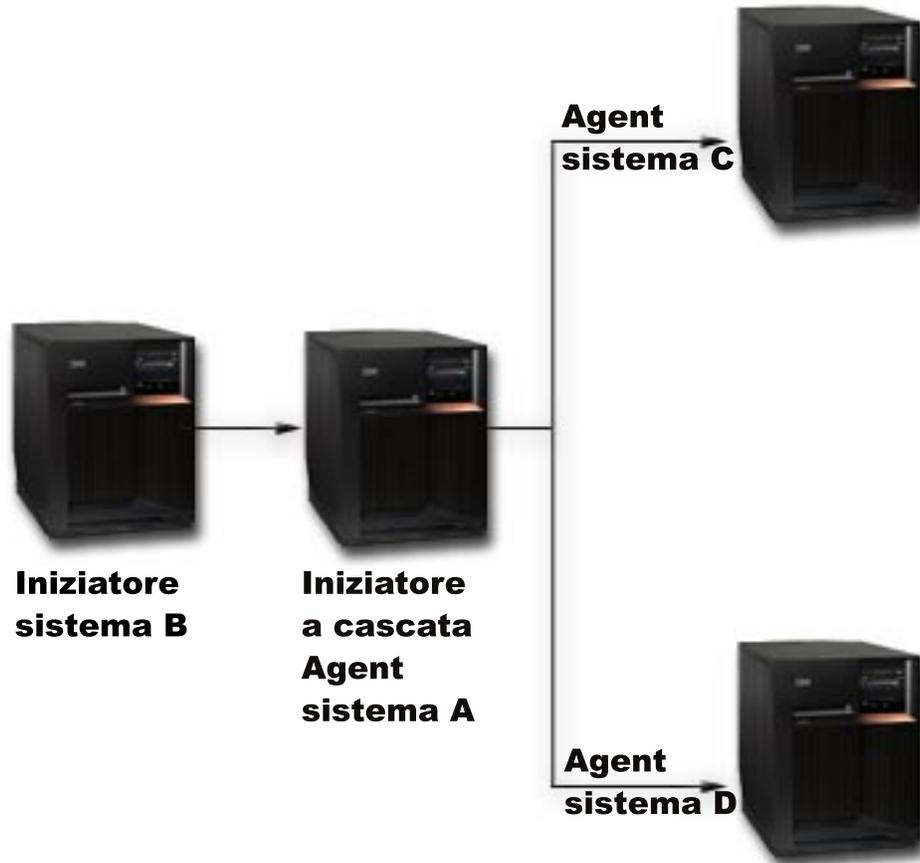
I gestori risorse per gli altri tre sistemi (B, C e D) diventano gli **agent** di questa transazione. Per la DUW (distributed unit of work) DRDA su TCP/IP, gli agent a volte vengono definiti **partecipanti**.



Ruoli nell'elaborazione della sincronizzazione a due fasi: albero a più livelli

Se l'applicazione sta utilizzando le comunicazioni APPC per eseguire la sincronizzazione a due fasi, il rapporto tra sistemi può variare da una transazione alla successiva. La seguente figura descrive gli stessi sistemi quando un'applicazione sul Sistema B emette una richiesta di sincronizzazione. Questa configurazione è un albero a più livelli.

I ruoli presenti in questa figura non si riferiscono ad una DUW (distributed unit of work) DRDA su TCP/IP poiché gli alberi delle transazioni a più livelli non sono supportati.



La rete di programmi di transazione possiede un altro livello poiché il Sistema B non comunica direttamente con il Sistema C e il Sistema D. Il gestore risorse nel Sistema A, ora, ha il ruolo di **agente e iniziatore a cascata**.

Per migliorare le prestazioni delle transazioni LU6.2 a due fasi, l'iniziatore può assegnare il ruolo di **ultimo agente** ad uno degli agent. L'ultimo agent non partecipa all'onda di preparazione. Nell'onda sincronizzata, l'ultimo agent esegue per primo la sincronizzazione. Se la sincronizzazione dell'ultimo agent non ha esito positivo, l'iniziatore ordina agli altri agent di eseguire il roll back.

Per la DUW (distributed unit of work) DRDA su TCP/IP, il coordinatore può assegnare il ruolo di **server risincronizzazione** ad un partecipante. Il server risincronizzazione è responsabile per la risincronizzazione degli altri partecipanti nel caso in cui si verifichi un errore nelle comunicazioni o nel caso in cui il coordinatore riceva errori di sistema.

Stati della transizione per il controllo sincronizzazione a due fasi

Viene stabilita una definizione sincronizzazione su ogni ubicazione che fa parte della rete di programmi di transazione. Per ogni definizione sincronizzazione, il sistema tiene traccia dello stato della relativa transazione corrente e di quella precedente. Il sistema utilizza lo stato per stabilire se eseguire il commit o il rollback nel caso in cui una transazione venga interrotta da un errore di sistema o nelle comunicazioni. Se più ubicazioni partecipano ad una transazione, gli stati delle transazioni su ogni ubicazione possono essere confrontati per stabilire la corretta azione (commit o rollback). Questo processo di comunicazione tra ubicazioni per stabilire la corretta azione viene definito **risincronizzazione**.

Nella seguente tabella vengono descritti:

- Gli stati principali che si possono verificare durante una transazione.

- Gli stati aggiuntivi che si possono verificare.
- Se uno stato richiede la risincronizzazione nel caso in cui la transazione viene interrotta da un errore di sistema o nelle comunicazioni. I valori possibili sono:

Non necessario

Ogni ubicazione può effettuare la corretta decisione indipendentemente.

Potrebbe essere necessario

Ogni ubicazione potrebbe effettuare la corretta decisione ma l'iniziatore potrebbe richiedere di essere informato della decisione.

Necessario

Lo stato di ogni ubicazione deve essere determinato prima di poter effettuare la corretta decisione.

- Azione eseguita da un errore di sistema o nelle comunicazioni.

| Nome stato | Descrizione | Risincronizzazione se la transazione viene interrotta | Azione eseguita da un errore di sistema o nelle comunicazioni. |
|---|--|---|--|
| Stati principali durante l'elaborazione della sincronizzazione a due fasi: | | | |
| Ripristino (RST) | Dal limite di sincronizzazione fino a quando un programma non emette una richiesta di commit o rollback. | Non necessaria. | Viene eseguito il rollback delle modifiche in sospeso. |
| Preparazione in corso (PIP) | L'iniziatore ha avviato l'onda di preparazione. Tutte le ubicazioni non hanno ancora fornito le indicazioni. | Potrebbe essere necessaria. | Viene eseguito il rollback delle modifiche in sospeso. |
| Preparato (PRP) | Questa ubicazione e tutte le ubicazioni successive nella rete di programmi di transazione hanno fornito l'indicazione di eseguire la sincronizzazione. Questa ubicazione non ha ancora ricevuto la notifica dall'iniziatore di eseguire la sincronizzazione. | Necessaria. | Incerta. Dipende dai risultati del processo di risincronizzazione. |
| Sincronizzazione in corso (CIP) | Tutte le ubicazioni hanno fornito l'indicazione di eseguire la sincronizzazione. L'iniziatore ha avviato l'onda sincronizzata. | Necessaria. | Le modifiche in sospeso vengono sincronizzate. Viene eseguita la risincronizzazione per assicurarsi che tutte le ubicazione vengano sincronizzate. Se viene riportato un rollback euristico da un'altra ubicazione, viene riportato un errore. |

| Nome stato | Descrizione | Risincronizzazione se la transazione viene interrotta | Azione eseguita da un errore di sistema o nelle comunicazioni. |
|---|---|---|--|
| Sincronizzato (CMT) | Tutti gli agent sono stati sincronizzati e hanno restituito una risposta a questo nodo. | Potrebbe essere necessaria. | Nessuna. |
| Stati aggiuntivi durante l'elaborazione della sincronizzazione a due fasi: | | | |
| Ultimo agent in sospenso (LAP) | Se viene selezionato l'ultimo agent, questo stato si verifica sull'inziatore tra lo stato PIP e lo stato CIP. L'inziatore ha ordinato all'ultimo agent di eseguire la sincronizzazione e non ha ancora ricevuto una risposta. | Necessaria | Incerta. Dipende dai risultati del processo di risincronizzazione. |
| Indicazione di sola lettura (VRO) | Questo agent ha risposto all'onda di preparazione indicando che non esistono modifiche in sospenso. Se lo stato indicazione di sola lettura è autorizzato, questo agent non viene incluso nell'onda sincronizzata. | Potrebbe essere necessaria. | Nessuna. |
| Rollback necessario (RBR) | Si verifica uno dei seguenti casi: <ul style="list-style-type: none"> • Un agent ha emesso una richiesta di rollback prima dell'operazione di commit. • Si è verificato un errore di transazione. • L'API QTNRBRQD è stata utilizzata per inserire la transazione in uno stato di rollback necessario. Il programma di transazione non può eseguire ulteriori modifiche sotto il controllo sincronizzazione. | Potrebbe essere necessaria. | Viene eseguito il rollback delle modifiche in sospenso. |
| Condizioni che si verificano a causa di azioni o errori dell'operatore: | | | |
| Rollback forzato | Questa ubicazione e tutte le ubicazioni successive nella rete di programmi di transazione, ad eccezione dell'ultimo agent, sono state sottoposte a rollback attraverso l'intervento dell'operatore. | Potrebbe essere necessaria | E' stato già eseguito il rollback delle modifiche in sospenso. |

| Nome stato | Descrizione | Risincronizzazione se la transazione viene interrotta | Azione eseguita da un errore di sistema o nelle comunicazioni. |
|--------------------------|---|---|--|
| Sincronizzazione forzata | Questa ubicazione e tutte le ubicazioni successive nella rete di programmi transazione, ad eccezione dell'ultimo agent, sono state sincronizzate attraverso l'intervento dell'operatore. | Potrebbe essere necessaria | E' stata già eseguita la sincronizzazione delle modifiche in sospenso. |
| Euristico misto (HRM) | Sono stati sincronizzati alcuni gestori risorse. Alcuni sono stati sottoposti a rollback. E' stato utilizzato l'intervento dell'operatore oppure si è verificato un errore di sistema. Non viene visualizzato Euristico misto come stato nei pannelli controllo sincronizzazione. Vengono inviati all'operatore i messaggi di notifica. | Potrebbe essere necessaria | L'operatore dovrebbe eseguire un'operazione di ripristino su tutte le ubicazioni partecipanti per riportare il database ad uno stato coerente. |

Definizioni sincronizzazione per controllo sincronizzazione a due fasi

Dopo aver avviato il controllo sincronizzazione, è possibile utilizzare l'API QTNCHGCO (Modifica opzioni sincronizzazione) per modificare le opzioni di sincronizzazione per la transazione. In base all'ambiente e alle applicazioni, la modifica delle opzioni di sincronizzazione può migliorare le prestazioni del sistema.

I seguenti collegamenti descrivono le opzioni di sincronizzazione e i motivi per utilizzarle:

- Consentire indicazione di sola lettura
- Non attendere risultato
- Indicare OK per uscire
- Non selezionare l'ultimo agent
- Indicazione attendibile

Se si sta utilizzando una DUW (distributed unit of work) DRDA sul collegamento TCP/IP, l'unica opzione che viene applicata è Consentire indicazione di sola lettura.

Definizione di sincronizzazione per la sincronizzazione a due fasi: consentire indicazione di sola lettura: Generalmente, il gestore transazioni partecipa ad entrambe le fasi di elaborazione della sincronizzazione. Per migliorare le prestazioni dell'elaborazione di sincronizzazione, è possibile impostare alcune o tutte le ubicazioni presenti in una transazione per consentire al gestore transazioni di fornire le indicazioni di sola lettura. Se l'ubicazione non dispone di modifiche sincronizzabili durante una transazione, il gestore transazioni fornisce le indicazioni di sola lettura durante l'onda di preparazione. L'ubicazione non partecipa all'onda sincronizzata. Questo consente un miglioramento delle prestazioni generali poiché i flussi di comunicazione, che solitamente si verificano durante l'onda sincronizzata, vengono eliminati durante le transazioni in cui non viene effettuato alcun aggiornamento su una o più ubicazioni remote.

Dopo aver avviato il controllo sincronizzazione, è possibile utilizzare l'API QTNCHGCO (Modifica opzioni di sincronizzazione) per modificare l'opzione Indicazione di sola lettura autorizzata in Y. Questo può essere effettuato nei seguenti casi:

- Uno o più sistemi remoti non dispongono di modifiche sincronizzabili per una transazione.

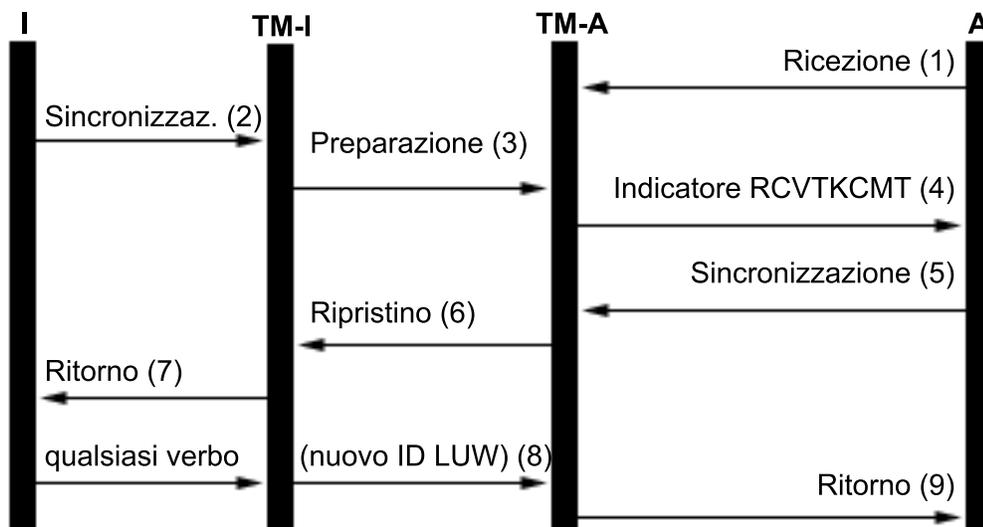
- Una transazione non dipende dal luogo in cui è stato impostato il cursore file (record successivo) dalla transazione precedente. Quando viene fornita un'indicazione di sola lettura tramite un'ubicazione, e se viene effettuato il rollback di una transazione, l'applicazione non viene mai notificata. L'ubicazione ha sincronizzato le operazioni di lettura con i file database e, quindi, ha spostato la posizione del cursore. La posizione del cursore file è importante solo se viene effettuata un'elaborazione sequenziale.

Se la definizione di sincronizzazione è impostata per consentire l'indicazione di sola lettura, l'applicazione attenderà il flusso di messaggi successivo da un'altra ubicazione.

L'opzione Indicazione di sola lettura autorizzata è valida per le applicazioni client/server. Se lo scopo di un programma A è solo quello di soddisfare le richieste provenienti dal programma I, e non per eseguire lavori indipendenti, è appropriato consentire l'opzione Indicazione di sola lettura per il programma A.

Flusso di elaborazione di sincronizzazione senza ottimizzazione ultimo agent quando l'agent fornisce indicazioni di sola lettura

La figura seguente illustra il flusso di messaggi tra i programmi dell'applicazione e i gestori transazioni quando un programma dell'applicazione emette un'istruzione di sincronizzazione senza ottimizzazione dell'ultimo agent, quando l'agent fornisce indicazioni di sola lettura. Sia il programma dell'applicazione dell'iniziatore sia quelli dell'agent ignorano l'elaborazione di sincronizzazione a due fasi. I numeri tra parentesi () nella figura corrispondono alle voci numerate nella descrizione che segue.



Legenda

- I = Iniziatore (Applicazione che avvia la rich. di sincronizzazione)
- TM-I = Gestore transazioni per l'iniziatore
- A = Agent (Applicazione che riceve la rich. di sincronizzazione)
- TM-A = Gestore transazioni per l'agent

Di seguito viene riportata una descrizione degli eventi per una normale elaborazione senza ottimizzazione ultimo agent quando l'agent fornisce indicazioni di sola lettura. Viene descritto un flusso di base. La sequenza di eventi può diventare molto più complessa quando la rete di programmi di transazione possiede diversi livelli o quando si verificano degli errori.

1. Il programma dell'applicazione A effettua una richiesta di ricezione per indicare che è pronto a ricevere una richiesta dal programma I.
2. L'applicazione dell'iniziatore (I) emette un'istruzione di sincronizzazione.

3. Il TM-I (gestore transazioni per l'iniziatore) assume il ruolo di iniziatore per questa transazione. Questo avvia l'onda di preparazione inviando un messaggio di preparazione a tutte le altre ubicazioni che partecipano alla transazione.
4. I gestori transazioni per tutte le altre ubicazioni assumono il ruolo di agent(TM-A). Il programma dell'applicazione A ha ricevuto da TM-A la notifica della ricezione di una richiesta di sincronizzazione. Per i file ICF, la notifica si trova nel modulo dell'indicatore ICF RCVTKCMT (Receive Take Commit) su cui è impostata.
5. Il programma dell'applicazione A risponde emettendo un'istruzione di commit (o un'istruzione di rollback). Questa rappresenta l'indicazione del programma dell'applicazione.
6. Se il programma dell'applicazione A ha utilizzato l'API QTNCHGCO (Modifica opzioni di sincronizzazione) per impostare l'opzione di sincronizzazione Indicazione di sola lettura autorizzata su Y, e non sono state apportate modifiche all'agent durante la transazione, l'agent (TM-A) risponde all'iniziatore (TM-I) con un messaggio di reimpostazione. Non esiste nessuna onda sincronizzata per l'agent.
7. Un messaggio di ritorno viene inviato al programma dell'applicazione (A) per indicare che la transazione sull'agent TM-A è completa.
8. La volta successiva che l'iniziatore (TM-I) emetterà dei messaggi per l'agent (TM-A), un flusso di dati o un'istruzione di sincronizzazione, l'ID transazione corrente verrà inviato con il messaggio. La ragione di ciò risiede nel fatto che, se si è verificato un errore nelle comunicazioni tra TM-I e altri sistemi durante l'operazione di sincronizzazione, potrebbe essere stato creato un nuovo ID transazione su TM-I.
9. Un messaggio di ritorno viene inviato al programma dell'applicazione (A) per indicare che la transazione sull'agent TM-A è completa. Il messaggio di ritorno viene ritardato fino alla ricezione del nuovo messaggio poiché è necessario che TM-I riceva un nuovo ID transazione, prima di poter avviare una nuova transazione dall'applicazione A.

Per ulteriori informazioni sul controllo sincronizzazione a due fasi, vedere Ruoli nell'elaborazione della sincronizzazione e Stati della transazione per il controllo sincronizzazione a due fasi.

Definizione sincronizzazione per sincronizzazione a due fasi: non attendere risultato: Quando si verifica un errore nelle comunicazioni o nei sistemi, durante un'operazione di sincronizzazione, tale da rendere necessaria la risincronizzazione, l'impostazione predefinita prevede di attendere il termine della risincronizzazione prima di completare l'operazione di sincronizzazione.

Nota: L'opzione Non attendere risultato non viene applicata se si sta utilizzando una DUW (distributed unit of work) DRDA sul collegamento TCP/IP. DUW (distributed unit of work) DRDA sul collegamento TCP/IP non attende mai il risultato.

E' necessario modificare questo comportamento se si verificano le seguenti condizioni:

- Ogni applicazione partecipante è indipendente dalle altre.
- La logica di programma non richiede i risultati delle transazioni precedenti per accertarsi che i file database rimangano sincronizzati.

Dopo aver avviato il controllo sincronizzazione, è possibile utilizzare l'API QTNCHGCO (Modifica opzioni sincronizzazione) per specificare che la definizione sincronizzazione non attende il risultato della risincronizzazione. Se viene specificato N (No) per l'opzione Attendere risultato, il sistema utilizza QDBSRVnn (lavoro server database) per gestire la risincronizzazione in modo asincrono.

Nota: Questi lavori server database vengono avviati durante il processo IPL. Se vengono modificate le opzioni per il controllo sincronizzazione, questo non influisce sul numero di lavori avviati dal sistema.

Questo argomento si riferisce solo ai due valori per l'opzione risolta Attendere risultato, Y (Sì) e N (No). Esistono altri due valori che è possibile specificare, L (Sì o eredita dall'iniziatore) e U (No o eredita dall'iniziatore). Quando si utilizzano questi valori, il valore attuale utilizzato durante ogni operazione di sincronizzazione viene risolto in Sì o No dal sistema. L'argomento API QTNCHGCO (Modifica opzioni sincronizzazione) dispone di ulteriori dettagli su questi valori.

Nota: Il valore dell'iniziatore può essere ereditato solo da un agent se sia l'iniziatore sia l'agent supportino l'interruzione presunta.

L'opzione WFO (Attendere risultato) non riguarda l'elaborazione sincronizzazione normale e priva di errori. Se si verifica un errore, l'opzione WFO stabilisce se l'applicazione attende o meno la risincronizzazione, tramite le seguenti condizioni:

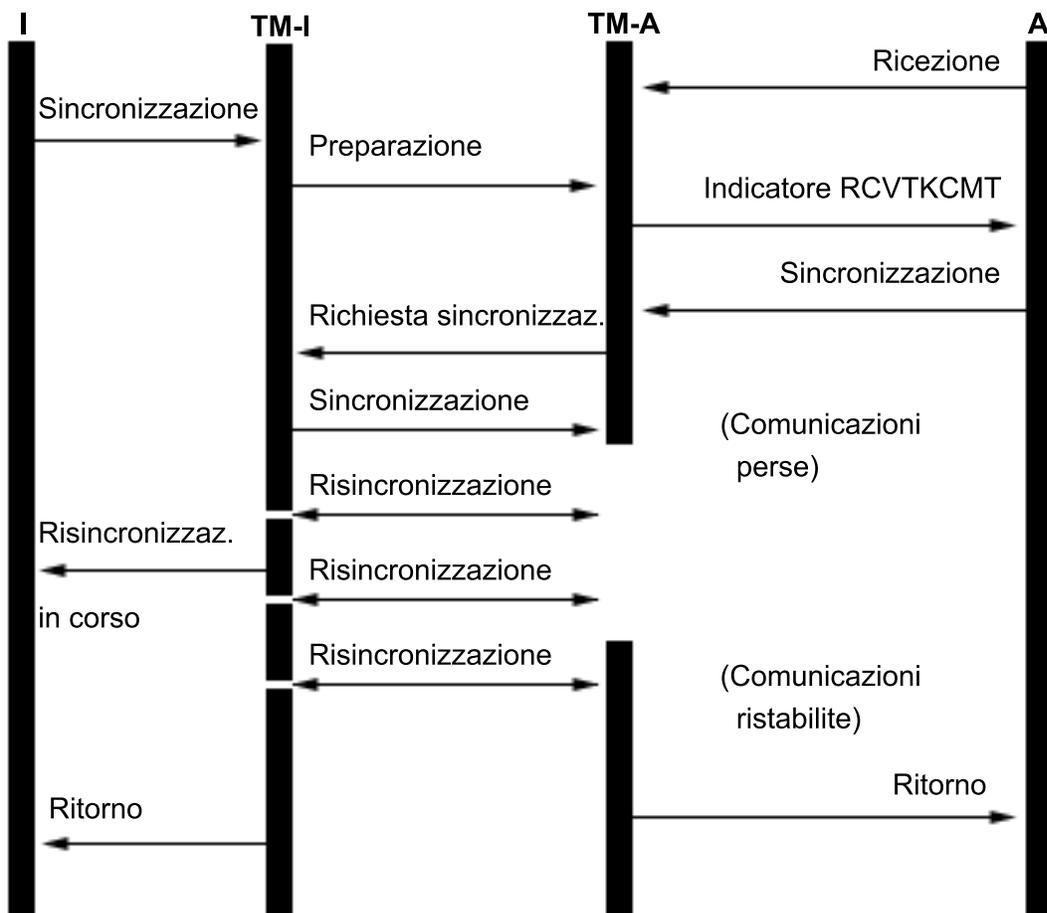
- Se l'opzione WFO risolta è Y (Sì), l'applicazione attende il risultato della risincronizzazione.
- Se l'opzione WFO risolta è N (No) e si verifica un errore di comunicazione durante l'onda di preparazione o il rollback di un'ubicazione che supporta i protocolli di presunta interruzione, la risincronizzazione non viene eseguita e viene effettuato il rollback della definizione sincronizzazione.
- Se la definizione sincronizzazione è incerta (lo stato transazione è preparato o Ultimo agent in sospenso), l'applicazione attenderà il risultato della risincronizzazione a prescindere dal valore WFO risolto. Per ulteriori informazioni sulla definizione sincronizzazione incerta, fare riferimento a Stati della transazione per controllo sincronizzazione a due fasi.
- Se l'opzione WFO risolta è N e non si presenta né la condizione due né la tre, il sistema tenta di effettuare nuovamente la risincronizzazione. Se non riesce, il sistema segnala il messaggio STATUS CPF83E6 all'applicazione per indicare che la risincronizzazione è in corso.

Poiché CPF83E6 è un messaggio STATUS, ha effetto solo se l'applicazione sta effettuando il monitoraggio. Generalmente, l'applicazione può trattare questo messaggio come un messaggio informativo. I sistemi che partecipano alla transazione tentano di risincronizzare la transazione fino a quando l'errore non viene riparato. Questi successivi tentativi di risincronizzazione vengono eseguiti nei lavori del server database. Se un successivo tentativo di risincronizzazione, eseguito in un lavoro del server database, non ha esito positivo, il messaggio CPI83D0 viene inviato a QSYSOPR.

Attendere risultato-Sì

Nella seguente figura, la definizione sincronizzazione per I (iniziatore) utilizza il valore predefinito di Y (Sì) per l'opzione Attendere risultato. Quando si perde la comunicazione tra TM-I e TM-A, sia l'applicazione

A sia l'applicazione I attendono fino a quando non viene risincronizzata la transazione.

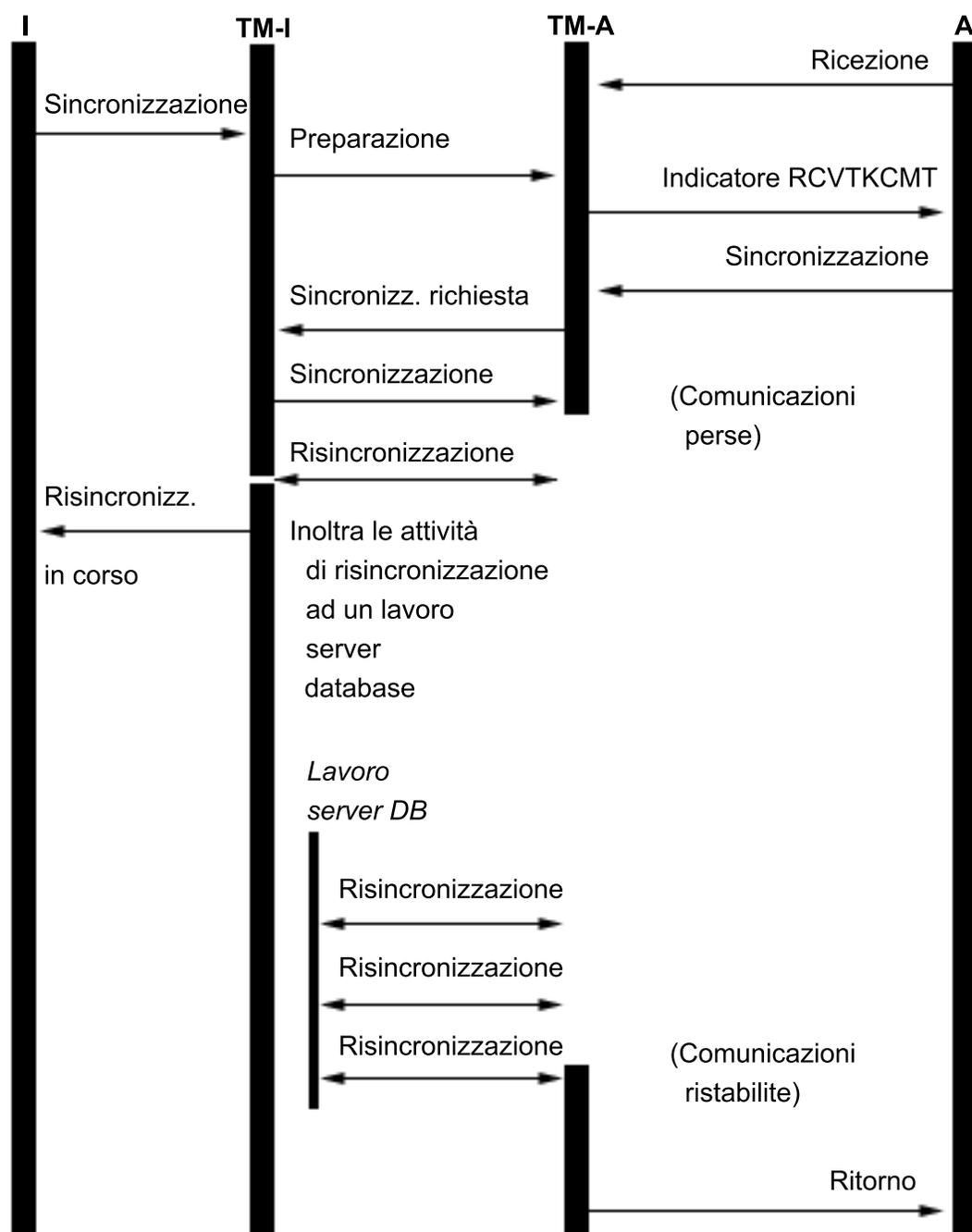


Attendere risultato-No

Nella seguente figura, la definizione sincronizzazione per l'iniziatore ha WFO impostata su N (No). TM-A risponde alla condizione 3 nella lista precedente, mentre TM-I risponde alla condizione 4. Il controllo viene restituito all'applicazione I in seguito ad un tentativo di risincronizzazione tramite TM-A. Un lavoro del server database tenta di effettuare la risincronizzazione. Quando il completamento della richiesta di sincronizzazione ha esito positivo, l'applicazione I non riceve l'indicatore di ritorno. Il controllo non viene restituito all'applicazione A (agent) fino a quando non vengono ristabilite le comunicazioni. Questo dipende dall'ora dell'errore. In questo caso, si verifica un errore nelle comunicazioni prima che venga ricevuto il messaggio di sincronizzazione dall'iniziatore, lasciando TM-A in dubbio se eseguire il commit o il rollback. Quando il gestore transazioni è in dubbio, mantiene il controllo fino al completamento della risincronizzazione, a prescindere dal valore WFO risolto su quel sistema.

Se si desidera che le applicazioni su tutti i sistemi continuino prima del completamento della sincronizzazione, è necessario modificare l'opzione WFO risolta su N (No) su tutti i sistemi oppure impostare l'iniziatore su N e il resto dei sistemi su U (No o eredita dall'iniziatore). Ma è necessario ricordarsi che l'opzione WFO risolta viene ignorata quando il gestore transazioni è in dubbio se eseguire il commit o il rollback e attende sempre il completamento della risincronizzazione prima di restituire il

controllo.



Quando viene effettuato un collegamento ad un RDB (relational database) remoto e non sono state avviate conversazioni protette, il sistema modifica implicitamente il valore Attendere risultato su N. La ragione risiede nel fatto che le prestazioni delle operazioni di sincronizzazione vengono migliorate quando il valore Attendere risultato è N e il sistema remoto supporta una presunta interruzione. Questa modifica implicita del valore Attendere risultato viene eseguita solo per le applicazioni DRDA e DDM. Le applicazioni APPC utilizzano il valore predefinito Attendere risultato su Y a meno che non richiamino l'API QTNCHGCO per modificarlo.

Definizione sincronizzazione per sincronizzazione a due fasi: indicare OK per uscire:

Generalmente, il gestore transazioni su ogni ubicazione nella rete di programmi di transazione partecipa a tutte le operazioni di commit o rollback. Per migliorare le prestazioni, è possibile impostare alcune o tutte

le ubicazioni in una transazione per consentire al gestore transazioni di indicare OK per uscire.

Nota:

L'opzione Indicare OK per uscire non viene applicata se si sta utilizzando una DUW (distributed unit of work) DRDA sul collegamento TCP/IP.

Se non vengono inviati flussi di comunicazioni sull'ubicazione durante una transazione, l'ubicazione viene abbandonata quando viene eseguita un'operazione di commit o rollback. Questo migliora le prestazioni generali poiché i flussi di comunicazioni, che normalmente si verificano durante il commit o il rollback, vengono eliminati durante le transazioni in cui non viene inviato alcun dato ad una o più ubicazioni remote.

Dopo aver avviato il controllo sincronizzazione, è possibile utilizzare l'API QTNCHGO (Modifica opzioni sincronizzazione) per modificare l'opzione OK per uscire su Y (Sì). Ciò è possibile se uno o più sistemi non sono coinvolti spesso in una transazione.

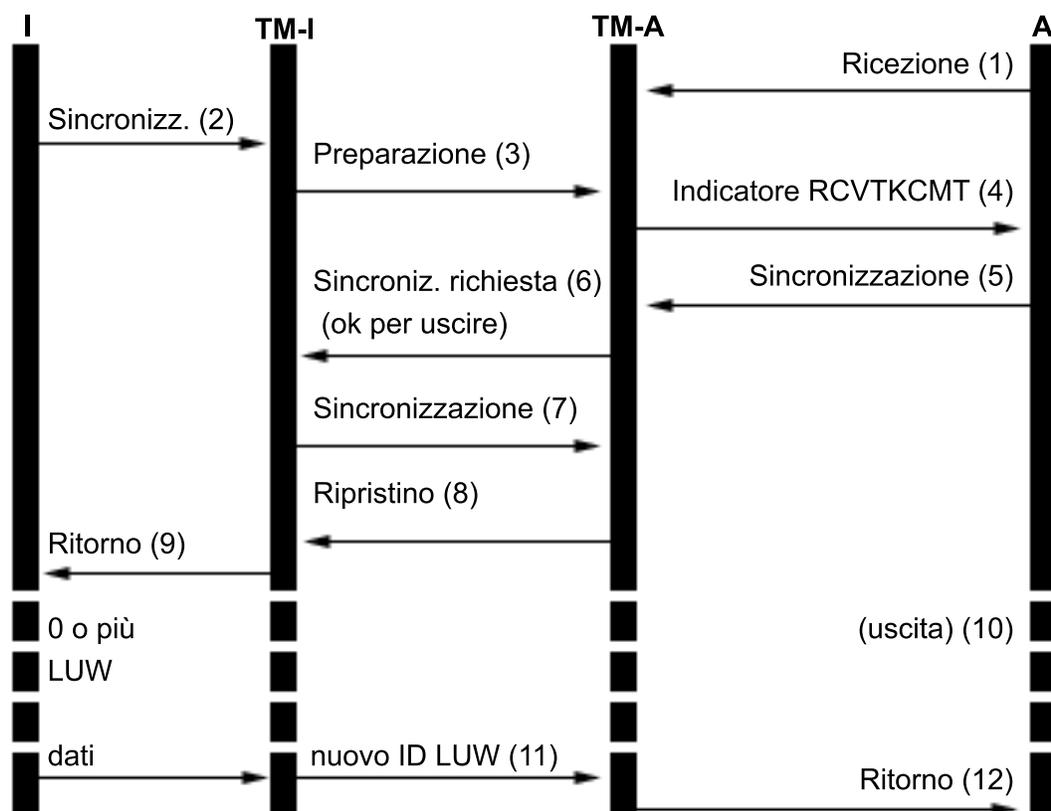
Se la definizione sincronizzazione è impostata su indicare OK per uscire, l'applicazione attende il successivo flusso di messaggi da un'altra ubicazione.

L'opzione OK per uscire è valida per le applicazioni client/server. Se lo scopo di un programma A è solo quello di soddisfare le richieste provenienti dal programma I e non di eseguire eventuali lavori indipendenti, è appropriato consentire l'opzione OK per uscire per il programma A.

Flusso di elaborazione di sincronizzazione senza ottimizzazione ultimo agent quando l'agent indica OK per uscire

Nella seguente figura viene visualizzato il flusso di messaggi tra i programmi dell'applicazione e i gestori transazioni quando un programma dell'applicazione emette un'istruzione di sincronizzazione senza ottimizzazione ultimo agent, quando l'agent indica OK per uscire. Sia il programma dell'applicazione dell'iniziatore sia quelli dell'agent ignorano l'elaborazione di sincronizzazione a due fasi. I numeri tra

parentesi () nella figura corrispondono alle voci numerate nella descrizione che segue.



Legenda

- I** = Iniziatore (Applicazione che avvia la richiesta di sincronizzazione)
- TM-I** = Gestore transazioni per l'iniziatore
- A** = Agent (Applicazione che riceve la richiesta di sincronizzazione)
- TM-A** = Gestore transazioni per l'agent

Di seguito viene riportata una descrizione degli eventi per l'elaborazione normale senza ottimizzazione ultimo agent quando l'agent indica OK per uscire. Viene descritto un flusso di base. La sequenza di eventi può diventare molto più complessa quando la rete di programmi di transazione possiede diversi livelli o quando si verificano degli errori.

1. Il programma dell'applicazione A effettua una richiesta di ricezione per indicare che è pronto a ricevere una richiesta dal programma I.
2. L'applicazione dell'iniziatore (I) emette un'istruzione di sincronizzazione.
3. Il TM-I (gestore transazioni per l'iniziatore) assume il ruolo di iniziatore per questa transazione. Questo avvia l'onda di preparazione inviando un messaggio di preparazione a tutte le altre ubicazioni che partecipano alla transazione.
4. I gestori transazioni per tutte le altre ubicazioni assumono il ruolo di agent (TM-A). Il programma dell'applicazione A ha ricevuto da TM-A la notifica della ricezione di una richiesta di sincronizzazione. Per i file ICF, la notifica si trova nel modulo dell'indicatore ICF RCVTKCMT (Receive Take Commit) su cui è impostata.
5. Il programma dell'applicazione A risponde emettendo un'istruzione di sincronizzazione (o un'istruzione di rollback). Questa rappresenta l'indicazione del programma dell'applicazione.
6. Se il programma dell'applicazione A ha utilizzato l'API QTNCHGCO (Modifica opzioni di sincronizzazione) per impostare l'opzione OK per lasciare la sincronizzazione su Y, viene inviato un indicatore quando TM-A (agent) risponde a TM-I (iniziatore) con un messaggio di sincronizzazione richiesta.

Nota:

Qualsiasi modifica apportata all'opzione OK per lasciare la sincronizzazione non ha effetto fino alla successiva operazione di sincronizzazione con esito positivo.

7. Quando TM-I (iniziatore) riceve tutte le indicazioni, invia un messaggio di sincronizzazione. Questo avvia l'onda sincronizzata.
8. Ogni TM-A (agent) esegue la sincronizzazione e risponde con un messaggio di ripristino.
9. Un messaggio di ritorno viene inviato al programma dell'applicazione (I) per indicare che la transazione sull'iniziatore è completa.
10. Su TM-I può verificarsi qualsiasi numero di transazioni, nessuna delle quali richiede delle modifiche su TM-A o richiede dati da TM-A. TM-A non è incluso in queste transazioni.
11. La volta successiva che TM-I (iniziatore) emetterà un messaggio per A (agent), un nuovo ID transazione verrà inviato con il messaggio. Se l'iniziatore esegue delle operazioni di commit o rollback prima di inviare un messaggio all'agent, nessun messaggio viene inviato all'agent durante tali operazioni (l'agent viene 'lasciato fuori' da tali commit o rollback). Poiché è possibile che una o più transazioni siano state sottoposte a commit o rollback sull'iniziatore mentre l'agent è lasciato fuori, l'iniziatore deve comunicare l'ID transazione corrente all'invio del successivo messaggio all'agent.
12. Un messaggio di ritorno viene inviato al programma dell'applicazione (A) per indicare che la sincronizzazione originale è completa e che partecipa alla transazione corrente.

Definizione sincronizzazione per sincronizzazione a due fasi: Non selezionare l'ultimo agent: Per impostazione predefinita, il gestore transazioni dell'iniziatore può selezionare qualsiasi agent come ultimo agent durante un'operazione di sincronizzazione.

Nota:

L'opzione Non selezionare ultimo agent non viene applicata se si sta utilizzando una DUW (distributed unit of work) DRDA su un collegamento TCP/IP.

In un albero a più livelli, qualsiasi agent selezionato come ultimo agent dal relativo iniziatore può selezionare un ultimo agent. Le prestazioni vengono migliorate quando viene selezionato l'ultimo agent durante l'operazione di sincronizzazione poiché due flussi di comunicazioni vengono eliminati tra un iniziatore e il relativo ultimo agent (la fase di preparazione viene eliminata per questi sistemi).

Tuttavia, dopo che l'iniziatore invia la sincronizzazione richiesta al relativo ultimo agent, deve attendere la ricezione dell'indicazione dell'ultimo agent prima di poter continuare. Questo a prescindere dal valore Attesa risultato della definizione sincronizzazione. Questo non rappresenta un problema durante un'elaborazione normale e priva di errori della sincronizzazione. Ma, se si verifica un errore in questa finestra, l'iniziatore non può continuare fino a quando non viene completata la risincronizzazione. Se l'applicazione dell'iniziatore gestisce le richieste da un utente al terminale, questa potrebbe essere una considerazione di utilizzabilità.

E' necessario considerare se le prestazioni migliorate durante le operazioni di sincronizzazione normali sono più rilevanti dell'impatto sull'utilizzabilità quando si verifica un tale errore. Notare che se l'errore si verifica prima che la sincronizzazione richiesta venga inviata all'ultimo agent, la LUW verrà immediatamente sottoposta a rollback e l'iniziatore non attenderà. Tuttavia, se la finestra, che viene visualizzata quando si verifica un errore che comporta l'attesa da parte dell'iniziatore, è ridotta, l'errore si verificherà raramente.

Se si stabilisce che l'impatto dell'utilizzabilità non è più rilevante delle prestazioni migliorate, è possibile modificare le definizioni sincronizzazione per non selezionare un ultimo agent. Dopo aver avviato il controllo sincronizzazione, è possibile utilizzare l'API QTNCHGCO (Modifica opzioni sincronizzazione) per modificare l'opzione Ultimo agent autorizzato su N.

Influenza dell'indicazione attendibile sul flusso di elaborazione della sincronizzazione:

L'indicazione attendibile è un'ottimizzazione che consente di migliorare le prestazioni ritornando all'applicazione dell'iniziatore dopo un'operazione di sincronizzazione ed eliminando un messaggio durante un'operazione di sincronizzazione. Non esiste nessuna ottimizzazione esplicita dell'indicazione attendibile per DUW (distributed unit of work) DRDA su TCP/IP. Tuttavia, OS/400 non richiede mai una conferma di ripristino per i collegamenti TCP/IP. Quindi, viene sempre implicato un ripristino per i collegamenti TCP/IP.

Dopo aver avviato il controllo sincronizzazione, è possibile utilizzare l'API QTNCHGCO (Modifica opzioni sincronizzazione) per modificare l'opzione Accettazione indicazione attendibile in Y.

L'indicazione attendibile può essere considerata come una promessa effettuata dall'agent al relativo iniziatore, che nessuna decisione euristica verrà effettuata sull'agent se si verifica un errore nelle comunicazioni mentre l'agent è incerto. Un agent che sta utilizzando un'ottimizzazione dell'indicazione attendibile invia un indicatore all'iniziatore durante l'onda di preparazione della sincronizzazione. Se anche l'iniziatore sta utilizzando l'ottimizzazione dell'indicazione attendibile, invia un indicatore all'agent in cui comunica che non è necessario alcun ripristino in risposta al messaggio di sincronizzazione. In questo modo viene eliminato il messaggio di ripristino e ciò consente al gestore transazioni di ritornare all'applicazione sull'iniziatore non appena viene inviato il messaggio di sincronizzazione.

E' necessario utilizzare l'ottimizzazione dell'indicazione attendibile se si verificano le seguenti condizioni:

- E' improbabile che venga effettuata una decisione euristica su un agent incerto nel caso in cui si verifichi un errore nei sistemi o nelle comunicazioni a meno che non sia possibile riparare tale errore.
- La logica di programma non richiede i risultati delle transazioni precedenti per accertarsiche i file database rimangono sincronizzati.

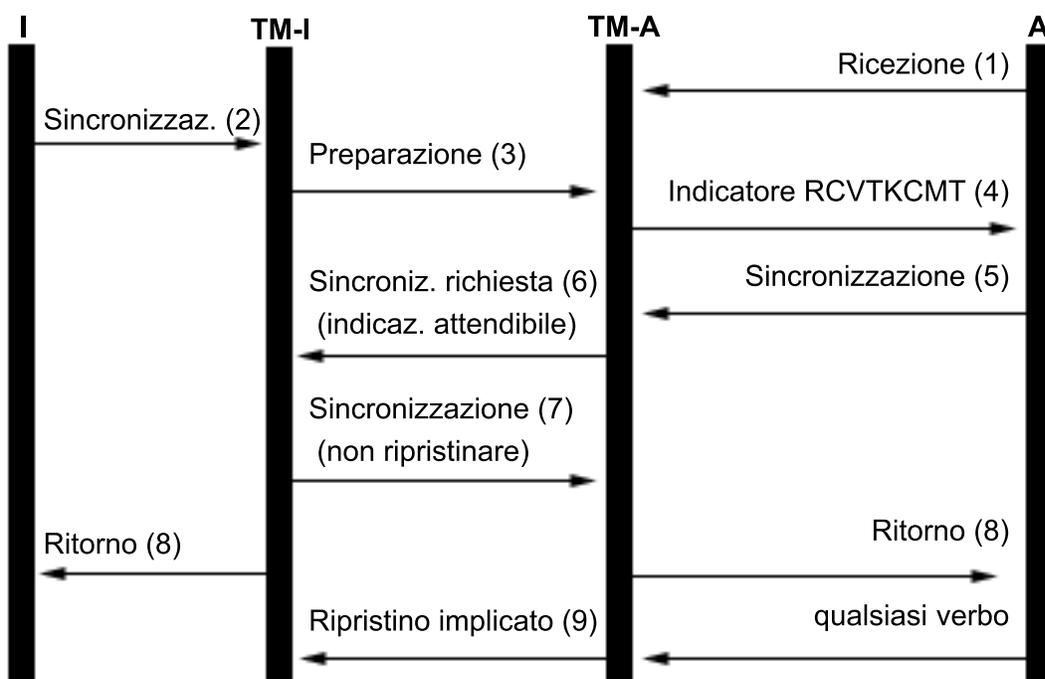
L'ottimizzazione dell'indicazione attendibile verrà utilizzata da OS/400 solo se si verificano tutte le seguenti situazioni:

- Le ubicazioni dell'iniziatore e dell'agent supportano il livello di interruzione presunta del controllo sincronizzazione.
- L'ubicazione dell'iniziatore accetta l'indicazione attendibile proveniente dall'agent. Sugli iniziatori OS/400, ciò dipende dal valore delle due opzioni di sincronizzazione:
 - Il valore dell'opzione di sincronizzazione Attendere risultato deve essere No (Sì come valore predefinito).
 - Il valore dell'opzione di sincronizzazione Accettazione indicazione attendibile deve essere Sì (No come valore predefinito).
- L'ubicazione dell'agent fornisce l'indicazione attendibile durante l'onda di preparazione. Gli agent OS/400 forniscono sempre l'indicazione attendibile. Ciò si verifica poiché le decisioni euristiche possono essere effettuate solo tramite una procedura manuale che segnala i possibili effetti negativi di una decisione euristica.

Flusso di elaborazione della sincronizzazione con ottimizzazione dell'indicazione attendibile

La seguente figura illustra il flusso di messaggi tra i programmi dell'applicazione e i gestori transazioni quando viene utilizzata l'ottimizzazione dell'indicazione attendibile. Sia il programma dell'applicazione dell'iniziatore sia quelli dell'agent ignorano l'elaborazione di sincronizzazione a due fasi. I numeri tra

parentesi () nella figura corrispondono alle voci numerate nella descrizione che segue.



Legenda

I = Iniziatore (Applicazione che avvia la richiesta di sincronizzazione)

TM-I = Gestore transazioni per l'iniziatore

A = Agent (Applicazione che riceve la richiesta di sincronizzazione)

TM-A = Gestore transazioni per l'agent

Di seguito viene riportata una descrizione degli eventi per l'elaborazione normale senza ottimizzazione dell'ultimo agent quando l'agent fornisce un'indicazione attendibile. Viene descritto un flusso di base. La sequenza di eventi può diventare molto più complessa quando la rete di programmi di transazione possiede diversi livelli o quando si verificano degli errori.

1. Il programma dell'applicazione A effettua una richiesta di ricezione per indicare che è pronto a ricevere una richiesta dal programma I.
2. L'applicazione dell'iniziatore (I) emette un'istruzione di sincronizzazione.
3. Il TM-I (gestore transazioni per l'iniziatore) assume il ruolo di iniziatore per questa transazione. Questo avvia l'onda di preparazione inviando un messaggio di preparazione a tutte le altre ubicazioni che partecipano alla transazione.
4. I gestori transazioni per tutte le altre ubicazioni assumono il ruolo di agent(TM-A). Il programma dell'applicazione A ha ricevuto da TM-A la notifica della ricezione di una richiesta di sincronizzazione. Per i file ICF, la notifica si trova nel modulo dell'indicatore ICF RCVTKCMT (Receive Take Commit) su cui è impostata.
5. Il programma dell'applicazione A risponde emettendo un'istruzione di sincronizzazione (o un'istruzione di rollback). Questa rappresenta l'indicazione del programma dell'applicazione.
6. Il TM-A (agent) risponde al TM-I (iniziatore) con un messaggio di sincronizzazione richiesta. I sistemi OS/400 inviano un indicatore di indicazione attendibile con la sincronizzazione richiesta.
7. Quando TM-I (iniziatore) riceve tutte le indicazioni, invia un messaggio di sincronizzazione. Se l'opzione di sincronizzazione Attendere risultato è N (No) e l'opzione di sincronizzazione Accettazione indicazione attendibile è Y (Si), viene inviato un indicatore di ripristino con il messaggio di sincronizzazione. Questo indica all'agent che non è necessario il messaggio di ripristino in risposta alla sincronizzazione.

8. La transazione è completa. Un messaggio di ritorno viene inviato ai programmi dell'applicazione (I e A). Questo indica che l'operazione di sincronizzazione è avvenuta con esito positivo. Se si verifica un danno euristico al sistema A a causa di una decisione euristica effettuata prima della ricezione di un messaggio sincronizzato, l'applicazione I non verrà informata. Al contrario, verrà inviato un messaggio sulla coda messaggi QSYSOPR. Tuttavia, l'applicazione A **riceverà** l'indicazione del danno euristico.
9. La volta successiva che TM-A (agent) invia dei messaggi al TM-I (iniziatore), un flusso di dati o un'istruzione di sincronizzazione, viene inviato un indicatore di ripristino implicato con il messaggio per comunicare a TM-I che TM-A ha completato la sincronizzazione con esito positivo. La ragione di questo è che il TM-I deve conservare le informazioni relative alla transazione completata fino a quando non viene confermato che TM-A ha ricevuto il messaggio di sincronizzazione nella fase 7.

Supporto transazioni XA per il controllo sincronizzazione

DB2 UDB for iSeries può partecipare alle transazioni globali X/Open. L'Open Group ha definito un modello standard industriale per il lavoro di transazione che consente che le modifiche effettuate sulle risorse non correlate facciano parte di una transazione globale. Un esempio di questo è rappresentato dalle modifiche al database fornite da due fornitori diversi. Questo modello viene definito modello X/Open Distributed Transaction Processing. Le seguenti pubblicazioni descrivono il modello X/Open Distributed Transaction Processing nel dettaglio:

- X/Open Guide, February 1996, Distributed Transaction Processing: Reference Model, Version 3 (ISBN:1-85912-170-5, G504), The Open Group.
- X/Open CAE Specification, December 1991, Distributed Transaction Processing: The XA Specification (ISBN:1-872630-24-3, C193 or XO/CAE/91/300), The Open Group.
- X/Open CAE Specification, April 1995, Distributed Transaction Processing: The TX (Transaction Demarcation) Specification (ISBN:1-85912-094-6, C504), The Open Group.

E' necessario conoscere le informazioni presenti in questi manuali, in particolare The XA Specification, prima di tentare di utilizzare il supporto transazioni XA fornito da DB2 UDB for iSeries. E' possibile trovare questi manuali sul sito Web Open Group



Esistono cinque componenti sul modello DTP:

Programma dell'applicazione (AP, Application Program)

Implementa la funzione desiderata dell'utente finale specificando una sequenza di operazioni che coinvolgono le risorse come ad esempio un database. Definisce l'avvio e la fine delle transazioni globali, accede alle risorse all'interno dei limiti di transazione e solitamente decide se eseguire il commit o il rollback di ogni transazione.

Gestore transazioni (TM, Transaction Manager)

Gestisce le transazioni e coordina la decisione di avviarle e di eseguire il commit o il rollback per assicurare il completamento della transazione atomica. Il TM inoltre coordina le attività di ripristino negli RM dopo un errore nel componente.

Gestore risorse (RM, Resource Manager)

Gestisce una parte definita delle risorse condivise del computer, come ad esempio un sistema di gestione database. L'AP utilizza le interfacce definite da ogni RM sulle interfacce fornite da RM per effettuare il completamento della transazione.

Gestore risorse di comunicazione (CRM, Communications Resource Manager)

Consente un'istanza del modello per accedere ad un'altra istanza all'intero o all'esterno del dominio TM corrente. I CRM sono esterni all'ambito di DB2 UDB for iSeries e non verranno trattati qui.

Protocollo comunicazione

I protocolli utilizzati dai CRM per comunicare l'uno con l'altro. Il protocollo è esterno all'ambito di DB2 UDB for iSeries e non verrà trattato qui.

The XA Specification è la parte del modello DTP che descrive una serie di interfacce utilizzate dai componenti TM e RM del modello DTP. DB2 UDB for iSeries implementa queste interfacce come una serie di API di stile UNIX e di programmi di uscita. Consultare API XA per la documentazione dettagliata di queste AOI e per ulteriori informazioni su come utilizzare DB2 UDB for iSeries come un RM.

iSeries Navigator e transazioni XA

iSeries Navigator supporta la gestione delle transazioni XA come **Transazioni globali**

Una transazione globale può contenere modifiche sia all'esterno sia all'interno di DB2 UDB for iSeries. Una transazione globale è coordinata da un Gestore transazioni esterno tramite la struttura XA Open Group o un'altra struttura simile. Un'applicazione esegue il commit o il rollback di una transazione globale utilizzando le interfacce fornite dal Gestore transazioni. Il TM utilizza i protocolli di sincronizzazione definiti dalla struttura XA o un'altra struttura, per completare la transazione. DB2 UDB for iSeries, quando partecipa ad una transazione globale, agisce con un Gestore risorse XA. Esistono due tipi di transazioni globali:

- **Blocchi nell'ambito della transazione:** i blocchi ottenuti per la transazione vengono inseriti nell'ambito della transazione. La transazione si può spostare da un lavoro o un sottoprocesso ad un altro.
- **Blocchi nell'ambito del lavoro:** i blocchi ottenuti per la transazione vengono inseriti nell'ambito del lavoro. La transazione non può spostarsi da un lavoro che la ha avviata.

Se si stanno eseguendo transazioni XA su un database che risiede sul sistema locale, è necessario utilizzare le API XA per i blocchi nell'ambito della transazione. Queste API possiedono menolimitazioni delle API XA per i blocchi nell'ambito del lavoro e forniscono delle prestazioni migliori nelle seguenti situazioni:

- Se vengono sempre utilizzati più collegamenti SQL per lavorare su un ramo transazione XA singolo.
- Se viene utilizzato un collegamento SQL singolo per lavorare su più rami transazione XA simultanei.

In queste situazioni, deve essere avviato un lavoro separato per eseguire i rami transazione XA quando vengono utilizzate le API XA per i Blocchi nell'ambito del lavoro.

Se si è in esecuzione su un database che risiede su un sistema remoto, è necessario utilizzare le API XA per i Blocchi nell'ambito del lavoro.

Considerazioni relative alle transazioni XA

E' necessario conoscere le seguenti informazioni e limitazioni prima di utilizzare DB2 UDB for iSeries come un RM. Il termine "sottoprocesso" si riferisce ad un lavoro che non ha la funzione di sottoprocesso o ad un singolo sottoprocesso all'interno di un lavoro con funzione di sottoprocesso.

Le seguenti considerazioni si riferiscono sia alle transazioni con blocchi nell'ambito della transazione sia alle transazioni con blocchi nell'ambito del lavoro a meno che non indicato diversamente.

Considerazioni relative a DB2 UDB for iSeries

- Le transazioni XA possono essere eseguite solo nei lavori che sono in esecuzione in modalità server SQL. Un effetto di ciò è che le applicazioni vengono limitate alle interfacce SQL quando vengono effettuate le modifiche su DB2 UDB for iSeries durante una transazione XA. Se viene utilizzata l'API

db2xa_open() in un lavoro che ancora non è in esecuzione in modalità server SQL, tale modalità verrà avviata implicitamente. Fare riferimento a Modalità server SQL e transazioni nell'ambito del sottoprocesso per il controllo sincronizzazione

- Eventuali errori rilevati da DB2 UDB for iSeries durante i richiami API XA vengono riportati attraverso i codici di ritorno in base alla specifica XA. I messaggi di diagnostica vengono lasciati nella registrazione lavori quando il significato dell'errore non è reso chiaro dal codice di ritorno.

Considerazioni relative a SQL incorporato

- Per utilizzare un collegamento SQL (Structured Query Language) per le transazioni XA, è necessario utilizzare l'API (application program interface) db2xa_open() prima di effettuare un collegamento SQL. L'RDB (relational database) che verrà collegato deve essere inoltrato all'API db2xa_open() tramite il parametro Xainfo. Il profilo utente e la parola d'ordine devono essere utilizzati nel lavoro in cui è instradato il collegamento per essere inoltrato all'API db2xa_open(). Se non viene inoltrato, il profilo sarà predefinito su uno specificato o fornito come valore predefinito durante il tentativo di collegamento.
- Se viene utilizzato SQL incorporato per eseguire le transazioni XA, il lavoro eseguito per ogni collegamento viene instradato su un lavoro differente, anche se i collegamenti sono eseguiti nello stesso sottoprocesso. Questo è diverso dalla modalità server SQL senza XA, in cui il lavoro eseguito per tutti i collegamenti in un sottoprocesso singolo viene instradato sullo stesso lavoro. Ciò si verifica poiché la specifica XA richiede una richiesta di preparazione, commit o rollback separata, per ogni istanza del gestore risorse.

Nota: la seguente considerazione si riferisce solo alle transazioni con blocchi nell'ambito del lavoro.

- Se l'SQL incorporato viene utilizzato per eseguire le transazioni XA, può essere eseguito solo un collegamento per RDB (relational database) per sottoprocesso. Quando il sottoprocesso non è associato in modo attivo ad un ramo transazione, il lavoro richiesto su uno dei collegamenti del sottoprocesso causerà l'utilizzo del programma di uscita ax_reg() del TM da parte di RM per stabilire se un lavoro deve avviare, riavviare o collegare un ramo transazione.

Se il lavoro deve avviare un ramo transazione, viene eseguito sul collegamento del sottoprocesso al RDB (relational database) corrispondente.

Se il lavoro deve collegare un ramo transazione, viene reinstradato sul collegamento al RDB (relational database) corrispondente eseguito nel sottoprocesso che ha avviato il ramo transazione. Notare che il sistema non impone che il profilo utente per quel collegamento sia lo stesso di quello sul collegamento del sottoprocesso da collegare. Il TM deve accertarsi che questo non sia un interesse della sicurezza. I TM tipici utilizzano lo stesso profilo utente per tutti i collegamenti. Questo profilo utente è autorizzato a tutti i dati gestiti dal TM. Un'ulteriore sicurezza di accesso a questi dati viene gestita da TM o AP invece di utilizzare i meccanismi di sicurezza iSeries standard.

- Se il lavoro deve riavviare un ramo transazione, il collegamento che viene utilizzato varia in base all'associazione ramo transazione sospesa che è stata stabilita avviando o collegando il ramo transazione.

Il lavoro successivo viene eseguito sullo stesso collegamento fino a quando l'API db2xa_end() viene utilizzata per sospendere o terminare l'associazione del sottoprocesso con il ramo transazione.

Considerazioni relative a CLI

- Se la CLI viene utilizzata per eseguire le transazioni XA, è possibile effettuare più di un collegamento nello stesso sottoprocesso dopo aver utilizzato l'API db2xa_open(). I collegamenti possono essere utilizzati in altri sottoprocessi per eseguire le transazioni XA, fino a quando gli altri sottoprocessi utilizzano l'API db2xa_open() con lo stesso valore di parametro Xainfo.

Nota: la seguente considerazione si riferisce solo alle transazioni con blocchi nell'ambito del lavoro.

- Se la CLI viene utilizzata per eseguire le transazioni XA, il collegamento utilizzato per avviare un ramo transazione deve essere utilizzato per tutto il lavoro su tale ramo transazione. Se un altro sottoprocesso deve collegare il ramo transazione, l'handle di collegamento per il collegamento utilizzato per avviare il ramo transazione deve essere inoltrato al sottoprocesso da collegare in modo tale che possa eseguire il lavoro sullo stesso collegamento. Allo stesso modo, se un sottoprocesso deve riavviare il ramo transazione, è necessario utilizzare lo stesso collegamento.

Nota: quanto segue si riferisce alle transazioni con blocchi nell'ambito della transazione e nell'ambito del lavoro.

Poiché gli handle di collegamento CLI non possono essere utilizzati in un lavoro differente, la funzione di collegamento è limitata ai sottoprocessi in esecuzione nello stesso lavoro che ha avviato il ramo transazione quando è stata utilizzata la CLI.

Considerazioni sul RDB (Relational database) remoto

Nota: queste considerazioni per un RDB (relational database) remoto si riferiscono solo alle transazioni con blocchi nell'ambito del lavoro.

- I collegamenti XA ad un RDB (relational database) remoto sono supportati solo se l'RDB risiede su un sistema che supporta i collegamenti DRDA DUW (Distributed Unit of Work). Ciò include i sistemi che eseguono DRDA su conversazioni SNA LU6.2 e i sistemi che utilizzano V5R1 quando eseguono DRDA tramite collegamenti TCP/IP.
- Prima di utilizzare la funzione di collegamento XA, è necessario utilizzare l'API `db2xa_open()` nel sottoprocesso da collegare. E' necessario specificare lo stesso nome RDB (relational database) e RMID sull'API `db2xa_open()` sia nel sottoprocesso che ha avviato il ramo transazione, sia nel sottoprocesso da collegare. Se il ramo transazione è attivo quando viene tentato un collegamento, il sottoprocesso da collegare è bloccato. Il sottoprocesso da collegare, rimane bloccato fino a quando il sottoprocesso attivo sospende o termina la relativa associazione con il ramo transazione.

Considerazioni relative al ripristino

- Il supporto di commit e rollback euristico manuale, fornito per tutte le definizioni di sincronizzazione può essere utilizzato se diventa necessario forzare un ramo transazione per eseguire il commit o il rollback mentre si trova in uno stato preparato. Fare riferimento a Quando forzare i commit e i rollback e quando annullare la risincronizzazione per i dettagli.

Considerazioni relative al ramo transazione

- Le informazioni relative ai rami transazione XA vengono visualizzate come parte delle informazioni sul controllo sincronizzazione visualizzate da iSeries Navigator e dai comandi `WRKJOB` (Gestione lavoro), `DSPJOB` (Visualizzazione lavoro) e `WRKCMTDFN` (Gestione definizione sincronizzazione). Vengono visualizzati il nome TM, lo stato ramo transazione, l'identificativo transazione e il qualificatore del ramo. Le definizioni sincronizzazione correlate a tutte le transazioni XA attive simultaneamente possono essere visualizzate tramite il comando `WRKCMTDFN JOB(*ALL) STATUS(*XOPEN)` o visualizzando le **Transazioni globali** in iSeries Navigator.

Nota: le seguenti voci si riferiscono solo alle transazioni con blocchi nell'ambito del lavoro.

- Se un'associazione tra un sottoprocesso e un ramo transazione esistente è sospesa o terminata tramite l'API `db2xa_end()`, il sottoprocesso può avviare un nuovo ramo transazione. Se il collegamento utilizzato per avviare il nuovo ramo transazione è stato utilizzato prima di avviare un ramo transazione differente e l'associazione del sottoprocesso con tale ramo transazione è stata terminata o sospesa dall'API `db2xa_end()`, è possibile avviare un nuovo lavoro server SQL. Un nuovo lavoro server SQL è necessario solo se il primo ramo transazione non è stato ancora completato dall'API `db2xa_commit()` o `db2xa_rollback()`. In questo caso, un altro messaggio di completamento SQL7908 viene inviato alla registrazione lavori che identifica il nuovo lavoro server SQL, mentre il lavoro server SQL originale del collegamento è stato identificato quando è stato stabilito il collegamento. Tutte le richieste SQL per il nuovo ramo transazione vengono instradate sul nuovo lavoro server SQL. Quando il ramo transazione viene completato tramite l'API `db2xa_commit()` o `db2xa_rollback()`, il nuovo lavoro server SQL viene riciclato e restituito al lotto lavori di preavvio.
- Un ramo transazione sarà contrassegnato per Solo rollback dal sistema quando si verificano le seguenti situazioni:
 - Un sottoprocesso termina quando è ancora associato al ramo transazione.

- L'API `db2xa_close()` viene utilizzato in un sottoprocesso che possiede un'associazione attiva con il ramo transazione.
- Un ramo transazione verrà sottoposto a rollback dal sistema se i sottoprocessi si sono ancora associati ad esso quando si verifica una delle seguenti situazioni:
 - Il collegamento correlato al ramo transazione viene chiuso
 - Il lavoro che ha avviato il ramo transazione è terminato.
 - Il sistema non funziona.
- Esiste una situazione in cui un ramo transazione verrà sottoposto a rollback dal sistema, senza considerare se esistono sottoprocessi ancora associati. Ciò si verifica quando il lavoro server SQL, su cui il lavoro del collegamento è instradato, viene terminato. Ciò può verificarsi solo quando viene utilizzato il comando `CL ENDJOB` (Fine lavoro) su tale lavoro.

Nota: le seguenti voci si riferiscono solo alle transazioni con blocchi nell'ambito del lavoro.

- Un ramo transazione non sarà influenzato se nessun sottoprocesso possiede un'associazione attiva con esso quando si verifica una delle seguenti situazioni. Il TM può eseguire il commit o il rollback del ramo transazione da qualsiasi sottoprocesso che ha utilizzato l'API `db2xa_open()` con lo stesso valore di parametro `Xainfo` specificato nel sottoprocesso che ha avviato il ramo transazione.
 - Il collegamento correlato al ramo transazione viene chiuso
 - Un sottoprocesso o un lavoro che ha eseguito il lavoro per il ramo transazione ma non possiede più un'associazione attiva con esso, utilizza l'API `db2xa_close()`.
 - Un sottoprocesso o un lavoro che ha eseguito il lavoro per il ramo transazione ma non possiede più un'associazione attiva con esso, utilizza l'API `db2xa_close()`.
 - Il sistema non funziona. In questo caso, il ramo transazione non viene influenzato solo se si trova in stato preparato. Se si trova in stato inattivo, il sistema esegue il rollback.

Modalità server SQL e transazioni nell'ambito del sottoprocesso per il controllo sincronizzazione

Le definizioni sincronizzazione con blocchi nell'ambito del lavoro, solitamente si trovano nell'ambito di un gruppo attivazione. Se un lavoro è a più sottoprocessi, tutti i sottoprocessi nel lavoro hanno accesso alla definizione sincronizzazione e le modifiche effettuate per una transazione particolare possono essere distribuite su più sottoprocessi. Ossia, tutti i sottoprocessi i cui programmi sono in esecuzione nello stesso gruppo attivazione partecipano ad un'unica transazione.

Esistono dei casi in cui è consigliabile che il lavoro di transazione si trovi nell'ambito del sottoprocesso piuttosto che nel gruppo attivazione. In altre parole, ogni sottoprocesso dovrebbe avere la propria definizione sincronizzazione e il lavoro di transazione per ogni definizione sincronizzazione dovrebbe essere indipendente dal lavoro eseguito in altri sottoprocessi.

Questo viene supportato da DB2 UDB for iSeries utilizzando l'API `QWTCHGJB` (Modifica lavoro) per modificare il lavoro eseguito in modalità server SQL. Quando nella modalità server SQL viene richiesto un collegamento SQL, viene instradato su un lavoro separato. Tutte le successive operazioni SQL eseguite per tale collegamento vengono instradate su quel lavoro. Quando viene effettuato il collegamento, il messaggio di completamento `SQL7908` viene inviato alla registrazione lavori del lavoro modalità server SQL che indica su quale lavoro vengono instradate le richieste SQL. La definizione sincronizzazione appartiene al lavoro indicato in questo messaggio. Se si verificano degli errori, potrebbe essere necessario controllare le registrazioni lavori per entrambi i lavori per comprendere l'origine del problema poiché non è stato effettuato nessun lavoro reale nel lavoro che ha eseguito le istruzioni SQL.

Durante l'esecuzione in modalità server SQL, è possibile utilizzare solo le interfacce SQL per eseguire il lavoro sotto il controllo sincronizzazione. E' possibile utilizzare le SQL o CLI (Call Level Interface) incorporate. Tutti i collegamenti effettuati attraverso SQL incorporata in un unico sottoprocesso vengono instradati sullo stesso lavoro back-end. Questo consente ad un'unica richiesta di sincronizzazione di

sincronizzare il lavoro per tutti i collegamenti, allo stesso modo di come avverrebbe in un lavoro non in esecuzione in modalità server SQL. Ogni collegamento effettuato attraverso la CLI viene instradato su un lavoro separato. La CLI richiede che il lavoro eseguito per ogni collegamento venga sottoposto a commit o rollback indipendentemente.

Non è possibile eseguire le seguenti operazioni sotto il controllo sincronizzazione durante l'esecuzione in modalità server SQL:

- Modifiche record effettuate con interfacce diverse dalle interfacce SQL
- Modifiche sui file DDM
- Modifiche sulle risorse sincronizzazione API

Non è possibile avviare il controllo sincronizzazione direttamente in un lavoro in esecuzione in modalità server SQL. Per ulteriori informazioni sulla modalità server SQL, fare riferimento alle seguenti pagine nell'argomento Database:

- Perché eseguire DB2 CLI in modalità server SQL
- Avviare DB2 CLI in modalità server SQL
- Limitazioni per l'esecuzione DB2 CLI in modalità server

Avviare il controllo sincronizzazione

Per avviare il controllo sincronizzazione, utilizzare il comando Avvio controllo sincronia (STRCMTCTL).

Nota: il controllo sincronizzazione non deve essere avviato dalle applicazioni SQL. SQL avvia automaticamente il controllo sincronizzazione all'ora di collegamento quando il livello di isolamento SQL è diverso da *NONE.

Quando viene utilizzato il comando STRCMTCTL è possibile specificare quanto segue:

Tali parametri vengono spiegati negli argomenti riportati di seguito.

Livello di blocco sincronizzazione

Specificare il livello di blocco tramite il parametro LCKLVL sul comando STRCMTCTL. Il livello specificato diventa il livello predefinito di blocco record per i file database aperti e inseriti sotto il controllo sincronizzazione per la definizione sincronizzazione. Per ulteriori informazioni, consultare Livello blocco sincronizzazione.

Oggetto di notifica sincronizzazione

Utilizzare il parametro NTFY per specificare l'oggetto di notifica. Un oggetto di notifica è una coda messaggi, un'area dati o un file database contenente le informazioni che identificano l'ultima transazione completata con esito positivo per una particolare definizione sincronizzazione, nel caso in cui tale definizione non sia terminata normalmente. Per ulteriori informazioni, consultare Oggetto di notifica sincronizzazione

Parametro ambito di sincronizzazione

Utilizzare il parametro CMTSCOPE per specificare l'ambito di sincronizzazione. Quando viene avviato il controllo sincronizzazione, il sistema crea una definizione sincronizzazione. Il parametro dell'ambito di sincronizzazione identifica l'ambito per la definizione sincronizzazione. Il valore predefinito prevede che la definizione sincronizzazione venga inserita nell'ambito del gruppo attivazione del programma che effettua la richiesta di avvio del controllo sincronizzazione. L'ambito alternativo è il lavoro.

Parametro giornale predefinito

E' possibile specificare un giornale predefinito quando viene avviato il controllo sincronizzazione. E' possibile utilizzare un giornale predefinito per questi motivi:

- Si desidera raccogliere le voci di giornale transazioni. Queste voci consentono di analizzare la cronologia delle risorse associate ad una transazione. Non vengono utilizzate per applicare ed eliminare le modifiche registrate su giornale. Il parametro OMTJRNE (Omissione voci di giornale) stabilisce se il sistema scrive le voci di transazione.
- Si desidera migliorare le prestazioni dei lavori che chiudono e riaprono i file all'interno di una fase di instradamento. Se vengono chiusi tutti i file assegnati ad un giornale che non è il giornale predefinito, tutte le informazioni di sistema relative al giornale vengono eliminate dalla fase di instradamento. Se un file assegnato a quel giornale viene aperto successivamente, tutte le informazioni relative al giornale devono essere ricreate. Il sistema conserva le informazioni relative al giornale predefinito con la definizione sincronizzazione, sia se le risorse assegnate a tale giornale sono attive sia se non sono attive.

Parametro testo di sincronizzazione

Utilizzare il parametro di testo TEXT per identificare il testo specifico da associare ad una definizione sincronizzazione durante la visualizzazione delle informazioni relative alle definizioni sincronizzazione avviate per un lavoro. Se non viene specificato alcun testo, il sistema fornisce una descrizione testo predefinita.

Parametro omissione voci di giornale

Se viene specificato un giornale predefinito per migliorare le prestazioni, è possibile utilizzare il parametro OMTJRNE per impedire che il sistema scriva le voci di giornale della transazione. Quando il sistema scrive le voci transazione, la dimensione del ricevitore del giornale aumenta in modo significativo e le prestazioni durante le operazioni di commit e rollback peggiorano.

Le voci transazione sono utili quando si sta impostando o controllando l'ambiente del controllo sincronizzazione o una nuova applicazione.

Le voci transazione vengono scritte sul giornale predefinito, a prescindere dal valore del parametro OMTJRNE, in queste condizioni:

- Si verifica un errore di sistema durante un'operazione di commit o rollback.
- Viene effettuata una modifica manuale su una risorsa che ha partecipato ad una transazione e la modifica ha causato una condizione euristica mista. Consultare Stati di una transazione per il controllo sincronizzazione a due fasi per una descrizione della condizione euristica mista. Questo tipo di modifica manuale viene definita una decisione euristica.

E' possibile utilizzare le informazioni relative alle risorse che hanno partecipato alla transazione per stabilire l'azione da eseguire in queste situazioni.

Nell'argomento Parte di lunghezza-variabile della voce di giornale, Tabella 15 tramite Tabella 21 vengono visualizzati i layout per i dati specifici della voce per le voci di giornale della transazione.

Gli argomenti seguenti forniscono ulteriori informazioni sull'avvio del controllo sincronizzazione:

- Oggetto di notifica sincronizzazione
- Livello blocco sincronizzazione

Oggetto di notifica sincronizzazione

Un **oggetto di notifica** è una coda messaggi, un'area dati o un file database contenente le informazioni che identificano l'ultima transazione completata con esito positivo per una particolare definizione sincronizzazione nel caso in cui tale definizione sincronizzazione non sia terminata normalmente. Le informazioni utilizzate per identificare l'ultima transazione riuscita per una definizione sincronizzazione vengono fornite dall'**identificazione di sincronizzazione** che associa un'operazione di sincronizzazione con un insieme specifico delle modifiche risorse sincronizzabili.

L'identificazione di sincronizzazione dell'ultima transazione riuscita per una definizione sincronizzazione, viene ubicata nell'oggetto di notifica solo se la definizione non termina normalmente. E' possibile utilizzare tali informazioni per stabilire dove si è interrotta l'elaborazione di un'applicazione, in modo tale che possa essere riavviata.

Per i lotti dischi indipendenti, l'oggetto di notifica deve risiedere sullo stesso lotto dischi indipendente o sullo stesso gruppo di lotti dischi indipendenti della definizione sincronizzazione. Se la definizione sincronizzazione viene spostata su un altro lotto dischi indipendente o un altro gruppo lotto dischi indipendente, è necessario che l'oggetto di notifica risieda sull'altro lotto o sull'altro gruppo. L'oggetto di notifica sull'altro lotto dischi indipendente o gruppo di lotti dischi indipendente viene aggiornato se la definizione sincronizzazione termina in modo anomalo. Se l'oggetto di notifica non viene trovato sull'altro lotto dischi indipendente o gruppo di lotti dischi indipendente, l'aggiornamento non avrà esito positivo e causerà un messaggio CPF8358.

Se le risorse registrate su giornale partecipano alla transazione corrente e viene eseguita un'operazione di sincronizzazione con un'identificazione di sincronizzazione, quest'ultima deve essere ubicata nella voce giornale di sincronizzazione (codice giornale e tipo di voce di C CM) che identifica quella particolare transazione come sincronizzata. Una voce giornale di sincronizzazione contenente l'identificazione di sincronizzazione viene inviata ad ogni giornale associato alle risorse che hanno partecipato alla transazione.

Nella seguente tabella viene visualizzato come specificare l'identificazione di sincronizzazione e la relativa dimensione massima. Se l'identificazione di sincronizzazione supera la relativa dimensione massima, viene troncata quando è scritta sull'oggetto di notifica.

| Linguaggio | Operazione | Numero massimo di caratteri nell'identificazione di sincronizzazione |
|---|--------------------------|--|
| CL | Comando COMMIT | 3000 ¹ |
| ILE RPG* | Codice operazione COMMIT | 4000 ¹ |
| PLI. | Sottoroutine PLICOMMIT | 4000 ¹ |
| ILE C* | Funzione _Rcommit | 4000 ¹ |
| ILE COBOL* | Verbo COMMIT | Non supportato |
| SQL | Istruzione COMMIT | Non supportato |
| Nota: | | |
| ¹ Se l'oggetto di notifica è un'area dati, la dimensione massima è 2000 caratteri. | | |

Quando un oggetto di notifica viene aggiornato con l'identificazione di sincronizzazione, l'aggiornamento avviene nel seguente modo:

File database

Se un file database viene utilizzato come l'oggetto di notifica, l'identificazione di sincronizzazione viene aggiunta alla fine del file. Qualsiasi record esistente viene lasciato nel file. Poiché alcuni utenti o lavori possono modificare i record nello stesso momento, ogni definizione sincronizzazione presente nel file dovrebbe contenere informazioni univoche per associare i dati con il lavoro e la definizione sincronizzazione non riuscita. E' possibile registrare su giornale il file da utilizzare.

Area dati

Se un'area dati viene utilizzata come oggetto di notifica, l'intero contenuto viene sostituito quando l'identificazione di sincronizzazione si trova nell'area dati. Se più di un utente o un lavoro sta utilizzando lo stesso programma, nell'area dati sarà presente solo l'identificazione di sincronizzazione dall'ultima definizione sincronizzazione non terminata normalmente. Di conseguenza, un singolo oggetto di notifica dell'area dati non può fornire le informazioni corrette per il riavvio dei programmi

delle applicazioni. Per risolvere questo problema, utilizzare un'area dati separata per ogni definizione sincronizzazione di ogni utente stazione di lavoro o di ogni lavoro.

Coda messaggi

Se una coda messaggi viene utilizzata come un oggetto di notifica, viene inviato il messaggio CPI8399 alla coda messaggi. L'identificazione di sincronizzazione viene ubicata nel testo di secondo-livello del messaggio CPI8399. Come con l'utilizzo di un file database per l'oggetto di notifica, i contenuti di ogni identificazione di sincronizzazione, dovrebbero identificare in modo univoco una definizione sincronizzazione particolare per un lavoro, in modo tale che sia possibile riavviare un programma dell'applicazione.

Vedere Esempio: utilizzare un oggetto di notifica per avviare un'applicazione, per esempi relativi all'utilizzo di un oggetto di notifica.

Livello blocco sincronizzazione

Il valore specificato per il parametro LCKLVL sul comando Avvio controllo sincronia (STRCMTCTL) diventa il livello predefinito del blocco record per i file database aperti e posti sotto il controllo sincronizzazione per la definizione sincronizzazione. Il livello predefinito del blocco record non può essere sovrascritto durante l'apertura dei file database locali. Tuttavia, i file database a cui si è avuto accesso tramite SQL utilizzano il livello di isolamento SQL corrente in vigore durante l'emissione della prima istruzione SQL. In Considerazioni e limitazioni relative al controllo sincronizzazione vengono riportate le considerazioni relative allo stato per le modifiche livello oggetto e record

Il livello di blocco dovrebbe essere specificato in base alle necessità, ai periodi di attesa consentiti e alle procedure di rilascio più spesso utilizzate.

Le seguenti descrizioni si riferiscono solo ai file aperti sotto il controllo sincronizzazione:

Livello blocco *CHG

Utilizzare questo valore per proteggere i record modificati dalle variazioni effettuate da altri lavori in esecuzione nello stesso tempo. Per i file aperti sotto il controllo sincronizzazione, il blocco viene mantenuto per la durata della transazione. Per i file non aperti sotto il controllo sincronizzazione, il blocco sul record viene mantenuto solo a partire dal momento in cui viene letto il record fino al completamento dell'operazione di aggiornamento.

Livello blocco *CS

Utilizzare questo valore per proteggere sia i record modificati sia i record richiamati, dalle variazioni effettuate da altri lavori in esecuzione nello stesso tempo. I record richiamati che non sono modificati, sono protetti fino a quando vengono rilasciati o fino a quando viene richiamato un record differente.

Il livello blocco *CS garantisce che gli altri lavori non possano leggere un record per l'aggiornamento di ciò che viene letto da tale lavoro. Inoltre, il programma non può leggere i record, per l'aggiornamento, che sono stati bloccati con un tipo di blocco record di *UPDATE in un altro lavoro, fino a quando tale lavoro non accede ad un record differente.

Livello blocco *ALL

Utilizzare questo valore per proteggere i record modificati e i record richiamati, che si trovano sotto il controllo sincronizzazione, dalle modifiche effettuate da altri lavori in esecuzione sotto il controllo sincronizzazione nello stesso tempo. I record richiamati o modificati vengono protetti fino alla successiva operazione di commit o rollback.

Il livello blocco *ALL garantisce che gli altri lavori non possano accedere al record per l'aggiornamento di ciò che viene letto da tale lavoro. Questo è diverso dal protocollo di blocco

normale. Quando il livello di blocco viene specificato come *ALL, non è possibile accedere anche ad un record che non viene letto per l'aggiornamento se viene bloccato con un tipo di blocco record di *UPDATE in un altro lavoro.

Nella seguente tabella viene illustrata la durata dei blocchi record per i file sotto il controllo sincronizzazione o i file non sotto il controllo sincronizzazione.

| Richiesta | Parametro LCKLVL | Durata del blocco | Tipo di blocco |
|---|--|---|---|
| Sola lettura | Nessun controllo sincronizzazione | Nessun blocco | Nessuno |
| | *CHG | Nessun blocco | Nessuno |
| | *CS | Dalla lettura a successiva lettura, commit o rollback | *READ |
| | *ALL | Dalla lettura a commit o rollback | *READ |
| Leggere per aggiornamento quindi aggiornare o cancellare ¹ | Nessun controllo sincronizzazione | Dalla lettura all'aggiornamento o alla cancellazione | *UPDATE |
| | *CHG | Dalla lettura all'aggiornamento o alla cancellazione | *UPDATE |
| | | Dall'aggiornamento o cancellazione al commit o rollback successivi ² | *UPDATE |
| | *CS | Dalla lettura all'aggiornamento o alla cancellazione | *UPDATE |
| | | Dall'aggiornamento o cancellazione al commit o rollback successivi ² | *UPDATE |
| | *ALL | Dalla lettura all'aggiornamento o alla cancellazione | *UPDATE |
| | | Dall'aggiornamento o cancellazione al commit o rollback successivi ² | |
| | Leggere per aggiornamento quindi rilasciare ¹ | Nessun controllo sincronizzazione | Dalla lettura al rilascio |
| *CHG | | Dalla lettura al rilascio | *UPDATE |
| *CS | | Dalla lettura al rilascio, al commit o al rollback | *UPDATE |
| | | Dal rilascio a successiva lettura, commit o rollback | *UPDATE |
| Dalla lettura al rilascio, al commit o al rollback | | *UPDATE | Dal rilascio a successivo commit o rollback |
| *UPDATE | | | |

| Richiesta | Parametro LCKLVL | Durata del blocco | Tipo di blocco |
|-----------------------|-----------------------------------|--|----------------|
| Aggiungere | Nessun controllo sincronizzazione | Nessun blocco | Nessuno |
| | *CHG | Da aggiunta a commit o rollback | *UPDATE |
| | *CS | Da aggiunta a commit o rollback | *UPDATE |
| | *ALL | Da aggiunta a commit o rollback | *UPDATE |
| Scrivere direttamente | Nessun controllo sincronizzazione | Per la durata della scrittura diretta | *UPDATE |
| | *CHG | Da scrittura diretta a commit o rollback | *UPDATE |
| | *CS | Da scrittura diretta a commit o rollback | *UPDATE |
| | *ALL | Da scrittura diretta a commit o rollback | *UPDATE |

Note:

¹Se un'operazione di commit o rollback viene eseguita dopo un'operazione di lettura-per-aggiornamento, ma prima che il record venga aggiornato, cancellato o rilasciato, il record viene sbloccato durante l'operazione di commit o rollback. Il record non sarà più protetto quando il commit o il rollback verrà completato.

²Se un record viene cancellato ma il commit o il rollback per la transazione non sono stati ancora emessi, il record cancellato non rimane bloccato. Se lo stesso lavoro o un lavoro differente tenta di leggere il record cancellato tramite chiave, il lavoro riceve un'indicazione non trovata del record. Tuttavia, se esiste un percorso di accesso munito di chiave univoca sul file, un altro lavoro non può inserire o aggiornare un record con lo stesso valore di chiave univoca di quello del record cancellato, fino a quando non viene sincronizzata la transazione. Se un record è bloccato ma il commit o il rollback per la transazione non sono stati ancora emessi, il record cancellato non rimane bloccato. Se lo stesso lavoro o un lavoro differente tenta di leggere il record cancellato tramite chiave, il lavoro riceve un'indicazione non trovata del record. Tuttavia, se esiste un percorso di accesso munito di chiave univoca sul file, un altro lavoro non può inserire o aggiornare un record con lo stesso valore di chiave univoca di quello del record cancellato, fino a quando non viene sincronizzata la transazione.

Un tipo di blocco record di *READ si ottiene sui record che non sono di lettura per l'aggiornamento quando il livello blocco è *CS o *ALL. Questo tipo di blocco impedisce agli altri lavori di leggere i record per l'aggiornamento ma non impedisce l'accesso ai record dall'operazione di sola lettura.

Un tipo di blocco record di *UPDATE si ottiene sui record che vengono aggiornati, cancellati, aggiunti o letti per l'aggiornamento. Questo tipo di blocco impedisce ad altri lavori di leggere i record per l'aggiornamento ed impedisce ai lavori, in esecuzione sotto il controllo sincronizzazione con un livello blocco record di *CS o *ALL, di accedere ai record anche per un'operazione di sola lettura.

I programmi che non stanno utilizzando il controllo sincronizzazione, possono leggere i record bloccati da un altro lavoro, ma non possono leggere i record per l'aggiornamento, a prescindere dal valore specificato per il parametro LCKLVL.

Il livello di blocco specificato per una definizione sincronizzazione quando viene avviato il controllo sincronizzazione per un gruppo attivazione o per un lavoro, si riferisce solo ai lavori aperti associati ad una definizione sincronizzazione particolare.

Nota:

I valori a livello di blocco *CS e *ALL impediscono di richiamare un record che attualmente possiede una modifica in sospeso da un lavoro differente. Tuttavia, i valori a livello di blocco *CS e *ALL *non* impediscono di richiamare un record, che utilizza un programma in esecuzione in un gruppo attivazione, che attualmente possiede una modifica in sospeso da un programma in esecuzione in un gruppo attivazione differente all'interno dello stesso lavoro.

All'interno dello stesso lavoro, un programma può modificare un record, già modificato nella transazione corrente, fino a quando l'accesso al record viene effettuato tramite la stessa definizione sincronizzazione. Quando viene utilizzata la definizione sincronizzazione alivello del lavoro, l'accesso al record modificato può essere effettuato da un programma, in esecuzione in qualsiasi gruppo attivazione, che utilizza la definizione sincronizzazione alivello del lavoro.

Terminare controllo sincronizzazione

E' possibile terminare il controllo sincronizzazione per la definizione sincronizzazione a livello del lavoro o a livello del gruppo attivazione, tramite il comando ENDCMTCTL (Fine controllo sincronizzazione). L'emissione del comando ENDCMTCTL indica al sistema che la definizione sincronizzazione, utilizzata dal programma che effettua la richiesta, deve essere terminata. Il comando ENDCMTCTL termina solo una definizione sincronizzazione per il lavoro e tutte le altre definizioni sincronizzazione per il lavoro rimangono invariate.

Se la definizione sincronizzazione è terminata, i programmi in esecuzione in un gruppo attivazione non possono effettuare modifiche sotto il controllo sincronizzazione, a meno che la definizione sincronizzazione per il lavoro sia già avviata. Se la definizione sincronizzazione a livello del lavoro è attiva, viene resa immediatamente disponibile per i programmi in esecuzione nel gruppo attivazione che ha appena terminato il controllo sincronizzazione.

Se la definizione sincronizzazione a livello del lavoro è terminata, qualsiasi programma in esecuzione nel lavoro che stava utilizzando la definizione sincronizzazione a livello del lavoro, non può più effettuare modifiche sotto il controllo sincronizzazione senza prima riavviare il controllo sincronizzazione con il comando STRCMTCTL.

Prima di emettere il comando ENDCMTCTL, è necessario effettuare quanto segue per terminare la definizione sincronizzazione:

- Tutti i file aperti sotto il controllo sincronizzazione per la definizione sincronizzazione da terminare, devono essere chiusi. La fine della definizione sincronizzazione a livello del lavoro include tutti i file sotto il controllo sincronizzazione da qualsiasi programma in esecuzione in qualsiasi gruppo attivazione che sta utilizzando la definizione sincronizzazione a livello del lavoro.
- Tutte le risorse sincronizzazione API per la definizione sincronizzazione da terminare, devono essere prima eliminate tramite l'API QTNRMVCR. La fine della definizione sincronizzazione a livello del lavoro include tutte le risorse sincronizzazione API aggiunte da qualsiasi programma in esecuzione in un gruppo attivazione che sta utilizzando la definizione sincronizzazione a livello del lavoro.
- E' necessario scollegare un database remoto associato alla definizione sincronizzazione da terminare.
- Tutte le conversazioni protette associate alla definizione sincronizzazione devono essere terminate normalmente tramite il livello di sincronizzazione corretto.

Se il controllo sincronizzazione è stato terminato in una lavoro interattivo e una o più risorse sincronizzabili associate alla definizione sincronizzazione hanno delle modifiche in sospeso, il messaggio di interrogazione CPA8350 viene inviato all'utente che domanda se eseguire il commit delle modifiche in sospeso, se eseguire il rollback delle modifiche o annullare la richiesta ENDCMTCTL.

Se il controllo sincronizzazione è stato terminato in un lavoro batch, e uno o più file chiusi associati alla definizione sincronizzazione da terminare, hanno ancora delle modifiche in sospeso, le modifiche sono sottoposte a rollback e viene inviato un messaggio:

- CPF8356 se vengono registrate solo le risorse locali
- CPF835C se vengono registrate solo le risorse remote
- CPF83E4 se vengono registrate sia le risorse locali sia quelle remote

Se un oggetto di notifica viene definito per la definizione sincronizzazione terminata, può essere aggiornato. Vedere "Aggiornamenti effettuati sull'oggetto di notifica" alle pagine 4-13 per ulteriori informazioni relative all'aggiornamento di un oggetto di notifica effettuato dal sistema.

Quando viene terminato un gruppo attivazione che possiede un'API registrata come ultimo agent, il programma di uscita per l'API viene richiamato per ricevere la decisione di commit o rollback. In questo caso, anche se il gruppo attivazione sta terminando normalmente, una richiesta di rollback potrebbe essere restituita dal programma di uscita API. Quindi, non sarebbe possibile eseguire l'operazione di sincronizzazione implicita.

Al termine della definizione sincronizzazione, viene eseguito il ripristino necessario. Non viene eseguito nessun ripristino aggiuntivo per le risorse sincronizzazione associate alla definizione appena terminata.

Al termine della definizione sincronizzazione, è possibile riavviare la definizione a livello del lavoro o a livello del gruppo attivazione per i programmi in esecuzione nel gruppo attivazione. E' possibile avviare la definizione sincronizzazione a livello del lavoro solo se non è già avviata per il lavoro.

Sebbene le definizioni sincronizzazione possano essere avviate e terminate molte volte dai programmi in esecuzione in un gruppo attivazione, la quantità di risorse di sistema necessarie per le ripetute operazioni di avvio e fine, possono causare una riduzione nelle prestazioni del lavoro e nelle prestazioni generali del sistema. Quindi, si consiglia di lasciare attiva una definizione sincronizzazione se un programma da richiamare successivamente la utilizzerà.

Termine del controllo sincronizzazione avviato dal sistema

Il sistema può terminare il controllo sincronizzazione o eseguire un'operazione implicita di commit o rollback. A volte il termine del controllo sincronizzazione avviato dal sistema è normale. Altre volte, il controllo sincronizzazione termina con una fine anomala del sistema o del lavoro.

Le seguenti pagine descrivono le situazioni in cui il sistema termina implicitamente il controllo sincronizzazione e quali azioni effettuare, se possibile:

- Controllo sincronizzazione durante la fine del gruppo attivazione
- Operazioni di commit e rollback implicite
- Controllo sincronizzazione durante la fine normale della fase di instradamento
- Controllo sincronizzazione durante la fine anomala di un sistema o di un lavoro
- Aggiornamenti sull'oggetto di notifica dopo la fine del controllo sincronizzazione
- Ripristino controllo sincronizzazione durante l'IPL (initial program load)

Controllo sincronizzazione durante la fine del gruppo attivazione

Il sistema interrompe automaticamente la definizione di sincronizzazione a livello del gruppo attivazione quando un gruppo attivazione termina. Se esistono delle modifiche in sospeso per la definizione di sincronizzazione a livello del gruppo attivazione e il gruppo attivazione termina in modo normale, il sistema esegue un'implicita operazione di sincronizzazione per la relativa definizione prima della chiusura. In caso contrario, viene eseguita l'operazione di rollback implicita per la definizione di sincronizzazione a livello del gruppo attivazione prima della chiusura se il gruppo attivazione viene terminato in modo anomalo o se vengono riscontrati degli errori dal sistema durante la chiusura dei file aperti sotto il controllo

sincronizzazione nell'ambito del gruppo attivazione.

Nota:

Un'implicita operazione di commit o rollback non viene *mai* eseguita durante l'elaborazione della chiusura del gruppo attivazione per le definizioni di sincronizzazione *JOB o *DFACTGRP. Questo perché le definizioni di sincronizzazione *JOB e *DFACTGRP non vengono mai terminate in seguito alla chiusura di un gruppo attivazione. Invece, queste definizioni di sincronizzazione vengono terminate esplicitamente tramite un comando ENDCMTCTL o tramite il sistema al termine del lavoro.

Il sistema chiude automaticamente i file nell'ambito del gruppo attivazione quando questo termina. Questo include qualsiasi file database nell'ambito del gruppo attivazione aperto sotto il controllo sincronizzazione. La chiusura di tali file si verifica prima di qualsiasi operazione di sincronizzazione, che è possibile eseguire per la definizione di sincronizzazione a livello del gruppo attivazione. Quindi, qualsiasi record che risiede in un buffer I/E viene prima forzato nel database prima dell'esecuzione di qualsiasi operazione di sincronizzazione implicita.

Come parte dell'operazione di commit o rollback implicita che può essere eseguita, viene effettuata una chiamata al programma di uscita commit e rollback API per ogni risorsa di sincronizzazione API associata alla definizione di sincronizzazione a livello del gruppo attivazione. Il programma di uscita deve completare la relativa elaborazione entro 5 minuti. Dopo aver richiamato il programma di uscita commit e rollback API, il sistema elimina automaticamente la risorsa di sincronizzazione API.

Se viene eseguita un'operazione di rollback implicita per una definizione di sincronizzazione terminata a causa della fine di un gruppo attivazione, l'oggetto di notifica, se definito per la definizione di sincronizzazione, può essere aggiornato. Per ulteriori informazioni relative all'aggiornamento di un oggetto di notifica tramite il sistema, vedere Aggiornamenti dell'oggetto di notifica.

Operazioni di commit e rollback implicite

Generalmente, un'operazione di commit o rollback viene avviata da un programma dell'applicazione tramite uno dei linguaggi di programmazione disponibili che supporta il controllo sincronizzazione. Questi tipi di operazioni di commit e rollback sono conosciute come **richieste di commit e rollback esplicite**. Tuttavia, in alcune istanze il sistema avvia un'operazione di commit e rollback per una definizione di sincronizzazione. Le operazioni di commit e rollback avviate dal sistema sono conosciute come **richieste di commit e rollback implicite**.

Le due tabelle seguenti mostrano come si comporta il sistema quando si verificano alcuni eventi correlati ad una definizione di sincronizzazione che possiede delle modifiche in sospeso. Una definizione di sincronizzazione possiede delle modifiche in sospeso in uno dei seguenti casi:

- Qualsiasi risorsa sincronizzabile è stata aggiornata.
- Un file database aperto sotto il controllo di sincronizzazione è stato letto poiché la lettura di un file modifica la relativa posizione.
- La definizione di sincronizzazione possiede una risorsa API. Poiché le modifiche sulle risorse API vengono effettuate da un programma utente, il sistema deve supporre che tutte le risorse API abbiano delle modifiche in sospeso.

La voce di giornale C CM (operazione commit) e C RB (operazione rollback) indica se l'operazione è stata esplicita o implicita.

La seguente tabella mostra le azioni che il sistema esegue quando un lavoro termina, sia in modo anomalo sia in modo normale, in base a quanto riportato di seguito:

- Lo stato della transazione.

- Il valore di azione-in-caso-di-fine-lavoro per la definizione sincronizzazione.
- Se la risorsa API è o meno l'ultimo agent.

| Stato | API ultimo agent | Opzione Azione in caso di fine lavoro ¹ | Operazione di commit o rollback |
|-------|------------------|--|---|
| RST | N/A | N/A | <p>Se la definizione sincronizzazione non è associata ad una transazione globale X/Open, viene eseguito un rollback implicito.</p> <p>Se la definizione sincronizzazione è associata ad una transazione globale X/Open, si verifica quanto segue:</p> <ul style="list-style-type: none"> • Se lo stato ramo transazione non è Attivo (S1), non viene eseguita nessuna azione e il ramo transazione viene lasciato nello stesso stato. • Se lo stato ramo transazione è Attivo (S1), viene eseguito un rollback implicito. |
| PIP | N/A | N/A | <p>Se la definizione sincronizzazione non è associata ad una transazione globale X/Open, viene eseguito un rollback implicito.</p> <p>Se la definizione sincronizzazione è associata ad una transazione globale X/Open, il ramo transazione si trova nello stato Inattivo (S2) e viene lasciato in tale stato.</p> |

| Stato | API ultimo agent | Opzione Azione in caso di fine lavoro ¹ | Operazione di commit o rollback |
|-------|------------------|--|--|
| PRP | N/A | WAIT | <p>Se la definizione sincronizzazione non è associata ad una transazione globale X/Open², si verifica quanto segue:</p> <ul style="list-style-type: none"> • Viene avviata la risincronizzazione per ricevere la decisione dall'iniziatore dell'operazione di sincronizzazione. • Viene eseguita la decisione restituita di eseguire il commit o il rollback. E questa viene considerata un'operazione esplicita. |

| Stato | API ultimo agent | Opzione Azione in caso di fine lavoro ¹ | Operazione di commit o rollback |
|-------|------------------|--|--|
| PRP | N/A | C | Se la definizione sincronizzazione non è associata ad una transazione globale X/Open ² , viene eseguita un'operazione di sincronizzazione implicita. |
| | | R | <p>Se la definizione sincronizzazione non è associata ad una transazione globale X/Open, viene eseguita un'operazione di rollback implicita.</p> <p>Se la definizione sincronizzazione è associata ad una transazione globale X/Open, si verifica quanto segue:</p> <ul style="list-style-type: none"> • Se il lavoro che ha avviato la transazione termina, la transazione viene lasciata in uno stato preparato fino a quando TM XA esegue il commit o il rollback. In questo caso lo stato ramo transazione XA verrà lasciato in Preparato (S3). • Se il lavoro server SQL, su cui è instradato il lavoro della transazione, è terminato, viene eseguito implicitamente un rollback forzato. In questo caso, lo stato ramo transazione XA verrà modificato in Completato euristicamente (S5). |
| CIP | N/A | N/A | Viene eseguita un'operazione di sincronizzazione esplicita. |

| Stato | API ultimo agent | Opzione Azione in caso di fine lavoro ¹ | Operazione di commit o rollback |
|-------|------------------|--|---|
| LAP | NO | WAIT | 1. La risincronizzazione sull'ultimo agent viene utilizzata per richiamare la decisione di eseguire il commit o il roll back. |
| | | | 2. Viene eseguita la decisione restituita di eseguire il commit o il roll back. E questa viene considerata un'operazione esplicita. |
| LAP | Si | WAIT | 1. Viene chiamata l'API dell'ultimo agent per richiamare la decisione di eseguire il commit o il rollback. |
| | | | 2. Viene eseguita l'operazione di commit o di rollback. E questa viene considerata un'operazione esplicita. |
| LAP | N/A | C | Viene eseguita un'operazione di sincronizzazione implicita. |
| | | R | Viene eseguita un'operazione di rollback implicita. |
| CMT | N/A | N/A | E' stata già completata un'operazione di sincronizzazione per questa definizione sincronizzazione e qualsiasi ubicazione dipendente. L'operazione di sincronizzazione è completa. |
| VRO | N/A | N/A | Gli agent locali e remoti forniscono indicazioni di sola lettura. Tutti gli agent dipendenti devono fornire indicazioni di sola lettura. Non è necessaria nessuna azione. |
| RBR | N/A | N/A | E' necessaria un'operazione di rollback. Viene eseguita un'operazione di rollback esplicita. |

| Stato | API ultimo agent | Opzione Azione in caso di fine lavoro ¹ | Operazione di commit o rollback |
|--|------------------|--|---------------------------------|
| <p>Nota: ¹ E' possibile modificare l'opzione Azione in caso di fine lavoro con l'API QTNCHGCO (Modifica opzioni di sincronizzazione).</p> <p>²Se la definizione sincronizzazione è associata con una transazione globale X/Open, si verifica quanto segue:</p> <ul style="list-style-type: none"> • Se il lavoro che ha avviato la transazione termina, la transazione viene lasciata in uno stato preparato fino a quando TM XA esegue il commit o il rollback. In questo caso lo stato ramo transazione XA verrà lasciato in Preparato (S3). • Solo per i blocchi nell'ambito della transazione, il lavoro server SQL, su cui è instradato il lavoro della transazione, è terminato, viene eseguito implicitamente un rollback forzato. In questo caso, lo stato ramo transazione XA transaction verrà modificato in Completato euristicamente (S5). | | | |

La seguente tabella mostra le azioni che il sistema esegue quando un gruppo attivazione termina e viene applicato solo alle transazioni con blocchi nell'ambito della transazione. Le azione del sistema sono basate su:

- Lo stato della transazione. (Viene sempre ripristinato (RST) quando un gruppo attivazione termina).
- Come termina il gruppo attivazione, in modo anomalo o normale.
- Se una risorsa API è l'ultimo agent.

Nota: Se una risorsa API è registrata come ultimo agent, questo fornisce il controllo della decisione di commit o rollback all'ultimo agent. Il risultato della decisione è considerata un'operazione esplicita

| Stato | API ultimo agent | Tipo di fine | Operazione di commit o rollback |
|-------|------------------|--------------|--|
| RST | No | Normale | Viene eseguita un'operazione di sincronizzazione implicita. Se esistono conversazioni protette, la definizione sincronizzazione diventerà l'iniziatore principale dell'operazione di sincronizzazione. |
| RST | No | Anomalo | Viene eseguito un rollback implicito. |
| RST | Sì | Normale | Viene chiamato il programma di uscita API. L'operazione di commit o rollback viene determinata dall'API. |
| RST | Sì | Anomalo | Viene chiamato il programma di uscita API. L'operazione di commit o rollback viene determinata dall'API. |

Controllo sincronizzazione durante la fine normale della fase di instradamento

Il sistema termina tutte le definizioni sincronizzazione per un lavoro quando la fase di instradamento termina normalmente.

Nota:

Quanto segue si riferisce solo alle definizioni sincronizzazione con blocchi nell'ambito del lavoro.

Una fase di instradamento termina normalmente in uno dei seguenti casi:

- Fine normale per un lavoro batch.
- Disattivazione normale di un lavoro interattivo.
- I comandi RRTJOB (Reinstradamento lavoro), TFRJOB (Trasferimento lavoro) o TFRBCHJOB (Trasferimento lavoro batch) terminano la fase di instradamento corrente e ne avviano una nuova.

Qualsiasi altra fine di una fase di instradamento viene considerata anomala ed è riconosciuta tramite un codice di completamento diverso da zero nel messaggio di completamento lavoro CPF1164 nella registrazione lavori.

Prima di terminare una definizione sincronizzazione durante la fine di una fase di instradamento, se la definizione sincronizzazione possiede delle modifiche in sospeso, il sistema esegue un'implicita operazione di rollback. Ciò comporta il richiamo del programma di uscita commit o rollback API per ogni risorsa sincronizzazione API associata alla definizione sincronizzazione. Il programma di uscita deve completare la relativa elaborazione entro 5 minuti. Dopo aver richiamato il programma di uscita commit e rollback API, il sistema elimina automaticamente la risorsa di sincronizzazione API.

Se per la definizione di sincronizzazione viene definito un oggetto di notifica, è possibile aggiornarlo. Per ulteriori informazioni relative all'aggiornamento di un oggetto di notifica tramite il sistema, consultare Aggiornamenti sull'oggetto di notifica.

Controllo sincronizzazione durante la fine anomala di un sistema o di un lavoro

Questo argomento si riferisce solo alle definizioni di sincronizzazione con blocchi nell'ambito del lavoro. Il sistema termina tutte le definizioni di sincronizzazione per un lavoro che termina in modo anomalo. Tali definizioni vengono terminate durante l'elaborazione del lavoro finale. Se il sistema termina in modo anomalo, tutte le definizioni di sincronizzazione avviate e utilizzate da tutti i lavori attivi al momento della fine anomala del sistema, vengono terminate. Le definizioni di sincronizzazione vengono terminate come parte dell'elaborazione di ripristino del database, eseguita durante il successivo IPL in seguito alla fine anomala del sistema.

Attenzione:

Il ripristino delle definizioni di sincronizzazione si riferisce ad una fine anomala del sistema o di un lavoro dovuta ad un errore di alimentazione, un errore hardware o ad un errore verificatosi nel sistema operativo o nel codice interno su licenza. Non è necessario utilizzare il comando ENDJOBABN (Fine anomala lavoro) per forzare la fine di un lavoro in modo anomalo. La fine anomala può determinare delle modifiche in sospeso da sottoporre a commit o rollback parziale, per le transazioni attive del lavoro che si sta terminando. L'IPL successivo potrebbe effettuare un tentativo di ripristino di eventuali transazioni parziali per il lavoro terminato tramite il comando ENDJOBABN.

Il risultato del ripristino del controllo sincronizzazione effettuato dal sistema durante l'IPL di un lavoro terminato tramite il comando ENDJOBABN, è incerto. Tale incertezza è dovuta al fatto che, quando il lavoro termina in modo anomalo, tutti i blocchi delle risorse di sincronizzazione vengono rilasciati. Tutte le modifiche in sospeso dovute alle transazioni parziali, vengono rese disponibili per altri lavori

ed è possibile che causino ulteriori modifiche errate del database, effettuate da altri programmi delle applicazioni. Allo stesso modo, qualsiasi ripristino IPL eseguito successivamente, può influire negativamente sulle modifiche apportate dalle applicazioni in seguito alla fine anomala del lavoro. Ad esempio, è possibile che una tabella SQL venga rilasciata durante il ripristino IPL come azione rollback per una tabella di creazione in sospeso. Tuttavia, potrebbero essere state già inserite da altre applicazioni alcune righe all'interno della tabella, dopo la fine anomala del lavoro.

Il sistema effettua le seguenti operazioni per le definizioni di sincronizzazione terminate durante la fine anomala di un lavoro o durante l'IPL successivo alla fine anomala del sistema:

- Prima di terminare una definizione di sincronizzazione, se quest'ultima possiede delle modifiche in sospeso, il sistema esegue un'implicita operazione di rollback, a meno che l'elaborazione di tale definizione non sia stata interrotta durante l'operazione di sincronizzazione. Se è stata terminata durante un'operazione di sincronizzazione, la transazione viene sottoposta a rollback, risincronizzazione o sincronizzazione, in base al relativo stato. Vedere Operazioni di commit e rollback implicite. Il processo per l'esecuzione dell'operazione di rollback implicita o per il completamento dell'operazione di commit, include il richiamo del programma di uscita commit e rollback API per ogni risorsa sincronizzazione API associata alla definizione di sincronizzazione. Dopo aver richiamato il programma di uscita commit e rollback API, il sistema elimina automaticamente la risorsa di sincronizzazione API.

Attenzione:

Chiudere un lavoro quando una transazione è incerta (lo stato della transazione è LAP o PRP) può causare delle incoerenze nel database (le modifiche potrebbero essere sincronizzate su uno o più sistemi e potrebbero essere sottoposte a roll back su altri sistemi).

- Se l'opzione di sincronizzazione *Azione in caso di fine lavoro* è COMMIT, le modifiche su questo sistema vengono sincronizzate se il lavoro è terminato, senza considerare se le modifiche su altri sistemi che partecipano alla transazione sono sottoposte a commit o rollback.
- Se l'opzione di sincronizzazione *Azione in caso di fine lavoro* è ROLLBACK, le modifiche su questo sistema vengono sottoposte a roll back se il lavoro è terminato, senza considerare se le modifiche su altri sistemi che partecipano alla transazione sono sottoposte a commit o rollback.
- Se l'opzione di sincronizzazione *Azione in caso di fine lavoro* è WAIT, il lavoro non terminerà fino a quando la risincronizzazione non sarà completata sul sistema che possiede la decisione di commit o roll back. Per eseguire la fine del lavoro prima del completamento della risincronizzazione, è necessario effettuare una decisione euristica ed annullare la risincronizzazione.

Non è consigliabile chiudere il lavoro o il sistema in modo anomalo durante l'esecuzione del rollback. Questo determinerà un altro rollback durante la chiusura del lavoro (o durante l'IPL successivo se il sistema è chiuso). Il rollback successivo ripeterà il lavoro eseguito dal rollback originale e richiederà maggior tempo per l'esecuzione.

- Se per la definizione di sincronizzazione viene definito un oggetto di notifica, è possibile aggiornarlo. Vedere Aggiornamenti dell'oggetto di notifica per ulteriori informazioni sul sistema tramite il quale viene aggiornato un oggetto di notifica.

Se un processo termina prima della fine del controllo sincronizzazione e le conversazioni protette sono ancora attive, la definizione di sincronizzazione potrebbe richiedere il commit o il rollback. L'azione è basata sull'opzione Stato o Azione in caso di fine lavoro per la definizione di sincronizzazione.

Aggiornamenti sull'oggetto di notifica

Per scopi dell'oggetto di notifica, le seguenti sono considerate modifiche non sincronizzate:

- Un aggiornamento su un record effettuato sotto il controllo sincronizzazione.
- Un record cancellato sotto il controllo sincronizzazione.
- Una modifica livello oggetto effettuata su un oggetto DDL locale sotto il controllo sincronizzazione.
- Un'operazione di lettura eseguita per un file database aperto sotto il controllo sincronizzazione. Ciò si verifica poiché la posizione del file viene riportata sull'ultimo limite di sincronizzazione quando viene

eseguita un'operazione di rollback. Se viene eseguita un'operazione di lettura sotto il controllo sincronizzazione, la posizione del file viene modificata ed esiste una modifica non sincronizzata per la definizione sincronizzazione.

- Una definizione sincronizzazione con una delle seguenti risorse aggiunte, ha sempre delle modifiche non sincronizzate:
 - Una risorsa sincronizzazione API
 - Una risorsa remota DRDA* (Distributed Relational Database Architecture)
 - Una risorsa DDM (Distributed Database Management Architecture)
 - Una risorsa LU 6.2

Ciò si verifica poiché il sistema non è al corrente di quando viene effettuata una modifica reale sull'oggetto o sugli oggetti associati a questi tipi di risorse. In Tipi di risorse sincronizzabili è possibile trovare ulteriori informazioni su come aggiungere e gestire questi tipi di risorse.

Il sistema effettua gli aggiornamenti sull'oggetto di notifica e tali aggiornamenti dipendono da uno dei seguenti modi in cui una definizione sincronizzazione può terminare:

- Se un lavoro termina normalmente e non esistono modifiche non sincronizzate, il sistema *non* inserisce nell'oggetto di notifica l'identificazione della sincronizzazione dell'ultima operazione di sincronizzazione avvenuta con esito positivo.
- Se viene eseguita un'operazione di sincronizzazione implicita per una definizione sincronizzazione a livello del gruppo attivazione quando viene terminato il gruppo attivazione, il sistema *non* inserisce nell'oggetto di notifica l'identificazione sincronizzazione dell'ultima operazione di sincronizzazione avvenuta con esito positivo.

Nota:

Le operazioni di sincronizzazione implicite non vengono *mai* eseguite per la definizione sincronizzazione *DFACTGRP o *JOB

- Se il sistema, il lavoro o un gruppo attivazione termina in modo anomalo prima della prima operazione di sincronizzazione con esito positivo per una definizione sincronizzazione, il sistema *non* aggiorna l'oggetto di notifica poiché non esiste l'ultima identificazione di sincronizzazione. Per effettuare una distinzione tra questa condizione e il normale completamento del programma, il programma deve aggiornare l'oggetto di notifica con una voce specifica prima di completare la prima operazione di sincronizzazione con esito positivo per la definizione sincronizzazione.
- Se si verifica una fine anomala del lavoro o del sistema dopo almeno un'operazione di sincronizzazione con esito positivo, il sistema inserisce l'identificazione di sincronizzazione di tale operazione di sincronizzazione nell'oggetto di notifica. Se l'ultima operazione di sincronizzazione con esito positivo non ha specificato un'identificazione di sincronizzazione, l'oggetto di notifica non viene aggiornato. Per una fine anomala del lavoro, questo oggetto di notifica viene eseguito per ogni definizione sincronizzazione attiva per il lavoro. Per una fine anomala del sistema, l'elaborazione di questo oggetto di notifica viene eseguita per ogni definizione di sincronizzazione attiva per tutti i lavori sul sistema.
- Il sistema aggiorna l'oggetto di notifica con l'identificazione di sincronizzazione dell'ultima operazione di sincronizzazione con esito positivo per quella definizione sincronizzazione se si verifica uno dei seguenti casi:
 - Un gruppo attivazione non predefinito termina.
 - Viene eseguita un'operazione di rollback implicita per la definizione sincronizzazione a livello del gruppo attivazione.
 - E' stata eseguita almeno un'operazione di sincronizzazione con esito positivo per quella definizione sincronizzazione.

Se l'ultima operazione di sincronizzazione con esito positivo non ha specificato un'identificazione di sincronizzazione, l'oggetto di notifica non viene aggiornato. Un'operazione di rollback implicita viene eseguita per una definizione sincronizzazione a livello del gruppo attivazione se il gruppo attivazione

termina in modo anomalo o se si verificano degli errori alla chiusura dei file aperti sotto il controllo sincronizzazione nell'ambito di tale gruppo attivazione. Per ulteriori informazioni relative all'inserimento dei file database nell'ambito dei gruppi attivazione e a come è possibile terminare i gruppi attivazione, vedere il manuale di riferimento per il linguaggio ILE che si sta utilizzando.

- Se esistono delle modifiche non sincronizzate quando un lavoro termina normalmente ed è stata eseguita almeno un'operazione di sincronizzazione con esito positivo, l'identificazione di sincronizzazione dell'ultima operazione di sincronizzazione con esito positivo viene inserita nell'oggetto di notifica e viene eseguito il rollback delle modifiche non sincronizzate. Se l'ultima operazione di sincronizzazione con esito positivo non ha specificato un'identificazione di sincronizzazione, l'oggetto di notifica non viene aggiornato. L'elaborazione di questo oggetto di notifica viene eseguita per ogni definizione sincronizzazione attiva per il lavoro al termine di quest'ultimo. Per ulteriori informazioni sulle funzioni che vengono eseguite durante una fine normale di un lavoro, vedere l'argomento Controllo sincronizzazione durante la fine normale della fase di instradamento.
- Se esistono delle modifiche non sincronizzate quando viene eseguito il comando ENDCMTCTL, l'oggetto di notifica viene aggiornato solo se l'ultima operazione di sincronizzazione con esito positivo ha specificato un'identificazione di sincronizzazione:
 - Per un lavoro batch, viene eseguito il rollback delle modifiche non sincronizzate e l'identificazione di sincronizzazione dell'ultima operazione di sincronizzazione viene inserita nell'oggetto di notifica.
 - Per un lavoro interattivo, se la risposta al messaggio di interrogazione CPA8350 è di eseguire il rollback delle modifiche, viene eseguito il rollback delle modifiche non sincronizzate e l'identificazione di sincronizzazione dell'ultima operazione di sincronizzazione con esito positivo viene inserita nell'oggetto di notifica.
 - Per un lavoro interattivo, se la risposta al messaggio di interrogazione CPA8350 è di sincronizzare le modifiche, il sistema richiede di utilizzare un'identificazione di sincronizzazione e le modifiche vengono sincronizzate. L'identificazione di sincronizzazione immessa sul pannello richiesta viene inserita nell'oggetto di notifica.
 - Per un lavoro interattivo, se la risposta al messaggio di interrogazione CPA8350 è di annullare la richiesta ENDCMTCTL, le modifiche in sospeso rimangono e l'oggetto di notifica *non* viene aggiornato.

Ripristino controllo sincronizzazione durante l'IPL (initial program load) in seguito ad una fine anomala.

Quando viene eseguito un IPL (initial program load) dopo la fine anomala di un sistema, questo tenta di ripristinare tutte le definizioni di sincronizzazione precedentemente attive. Allo stesso modo, quando viene attivato un lotto dischi indipendente, il sistema tenta di ripristinare tutte le definizioni di sincronizzazione correlate al lotto dischi indipendente, attive prima della disattivazione o della fine anomala. Il ripristino viene eseguito tramite i lavori server database avviati dal sistema durante l'IPL. I lavori server database vengono avviati dal sistema per gestire il lavoro che non è possibile o che non dovrebbe essere eseguito da altri lavori.

I lavori server database sono definiti QDBSRVnn, dove nn è un numero a due cifre. Il numero di lavori server database dipende dalla dimensione del sistema. Allo stesso modo, il nome del lavoro server database per un lotto dischi indipendente o un gruppo lotto dischi indipendente è QDBSxxxVnn, dove xxx è il numero del lotto dischi indipendente e nn è un numero a due cifre. Ad esempio, QDBS035V02 può essere il nome del lavoro server database per il lotto dischi indipendente 35.

In Stati della transazione per il controllo di sincronizzazione a due fasi vengono indicate le azioni eseguite dal sistema, in base allo stato della transazione quando si verifica l'errore. Per due stati, PRP e LAP, l'azione del sistema è incerta.

Nota:

- Quanto segue si riferisce solo alle definizioni sincronizzazione con blocchi nell'ambito del lavoro.
- Il gestore transazioni ripristina le definizioni sincronizzazione associate alle transazioni XA (sia se i relativi blocchi si trovano o meno nell'ambito del lavoro o nell'ambito della transazione) tramite le API XA, non il processo di risincronizzazione descritto in questo argomento.

Il sistema non può determinare le successive operazioni fino a quando non viene eseguita la risincronizzazione con altre ubicazioni che hanno partecipato alla transazione. Questa risincronizzazione viene eseguita in seguito al completamento dell'IPL o dell'operazione di attivazione.

Per eseguire la risincronizzazione, il sistema utilizza i lavori server database. Le definizioni sincronizzazione da ripristinare sono associate ai lavori server database. Durante l'IPL, il sistema acquisisce tutti i blocchi record e altri blocchi oggetti contenuti dalla definizione sincronizzazione prima della fine del sistema. Questi blocchi sono necessari per proteggere le risorse sincronizzazione locali fino al completamento della risincronizzazione e le risorse possono essere sottoposte a commit o a roll back.

I messaggi vengono inviati alle registrazioni lavori dei lavori server database per indicare lo stato della risincronizzazione con le ubicazioni remote. Se la transazione è incerta, è necessario completare la risincronizzazione con l'ubicazione che possiede la decisione per la transazione, prima di poter eseguire il commit o il rollback delle risorse locali.

Quando viene effettuata una decisione per la transazione, potrebbero essere inviati i seguenti messaggi alla registrazione lavori per il lavoro server database.

CPI8351

&1 modifica in sospeso da sottoporre a rollback

CPC8355

Ripristino post-IPL della definizione sincronizzazione &8 per lavoro &19/&18/&17 completato.

CPD835F

Ripristino IPL della definizione sincronizzazione &8 per lavoro &19/&18/&17 non riuscito.

Inoltre, potrebbero essere inviati altri messaggi correlati al ripristino. Questi messaggi vengono inviati alla registrazione QHST (cronologica). Se si verificano errori, i messaggi vengono anche inviati alla coda messaggi QSYSOPR.

E' possibile stabilire il corso del ripristino tramite iSeries Navigator, visualizzando la registrazione lavori per il lavoro server database oppure utilizzando il comando Gestione definizione sincronia (WRKCMTDFN). Sebbene iSeries Navigator e il pannello Gestione definizione sincronia consentono di forzare il sistema per eseguire il commit o il rollback, utilizzare ciò solo come ultima soluzione. Se si prevede che tutte le ubicazioni che hanno partecipato alla transazione saranno riportate in funzione, è necessario consentire al sistema di risincronizzarle. Ciò garantisce l'integrità dei database.

Gestire le transazioni e il controllo sincronizzazione

Le seguenti sono le attività possibili per gestire il controllo sincronizzazione.

Visualizzare le informazioni relative al controllo sincronizzazione

Queste informazioni contengono le attività possibili per visualizzare le informazioni relative a tutte le transazioni sul sistema e per visualizzare le informazioni relative ai lavori associati alle transazioni.

Ottimizzare le prestazioni per il controllo sincronizzazione

Queste informazioni contengono le attività possibili per ridurre gli effetti del controllo sincronizzazione sulle prestazioni del sistema.

Visualizzare informazioni sul controllo sincronizzazione

E' possibile utilizzare iSeries Navigator per visualizzare le informazioni relative a tutte le transazioni (unità logiche di lavoro) sul sistema. E' possibile, inoltre, controllare le informazioni relative al lavoro, se esistente, associato ad una transazione.

Nota: Queste operazioni di visualizzazione non visualizzano il livello di isolamento delle applicazioni SQL.

Per visualizzare le informazioni, procedere nel seguente modo:

1. Nella finestra **iSeries Navigator**, espandere il server che si desidera utilizzare.
2. Espandere **Database**.
3. Espandere il sistema che si desidera gestire.
4. Espandere **Transazioni**.

Nota: Per visualizzare una transazione associata ad una transazione globale X/Open, espandere **Transazioni globali**. Per visualizzare una transazione gestita DB2 UDB, espandere **Transazioni database**.

5. Espandere **Transazioni globali** o **Transazioni database**.

Questo pannello visualizza quanto segue:

- ID unità di lavoro
- Stato unità di lavoro
- Lavoro
- Utente
- Numero
- Risincronizzazione in corso
- Definizione sincronizzazione

La guida in linea fornisce le informazioni relative a tutti i pannelli di stato e ai campi su ogni pannello.

Inoltre, è possibile utilizzare iSeries Navigator per visualizzare le seguenti informazioni:

- Visualizzare oggetti bloccati per una transazione
- Visualizzare i lavori associati ad una transazione
- Visualizzare stato risorsa di una transazione
- Visualizzare proprietà transazione

Visualizzare oggetti bloccati per una transazione

E' possibile visualizzare gli oggetti bloccati per le transazioni globali solo con blocchi nell'ambito della transazione.

Per visualizzare gli oggetti bloccati per una transazione:

1. Nella finestra **iSeries Navigator**, espandere il server che si desidera utilizzare.
2. Espandere **Database**.
3. Espandere il sistema che si desidera gestire
4. Espandere **Transazioni**.

5. Espandere **Transazioni globali**.
6. Fare clic con il tasto destro del mouse sulla transazione da gestire e selezionare **Oggetti bloccati**.

Visualizzare lavori associati ad una transazione

Per visualizzare i lavori associati ad una transazione:

1. Nella finestra **iSeries Navigator**, espandere il server che si desidera utilizzare.
2. Espandere **Database**.
3. Espandere il sistema che si desidera gestire
4. Espandere **Transazioni**.
5. Espandere **Transazioni globali** o **Transazioni database**.
6. Fare clic con il tasto destro del mouse sulla transazione da gestire e selezionare **Lavori**.

Per transazioni database e globali con blocchi nell'ambito del lavoro, viene visualizzata una lista dei lavori associati alla transazione.

Per le transazioni globali con blocchi nell'ambito della transazione, viene visualizzata una lista di lavori con questo oggetto transazioni allegato o in attesa di allegare questo oggetto di transazione

Visualizzare stato risorsa di una transazione

Per visualizzare lo stato risorsa di una transazione:

1. Nella finestra **iSeries Navigator**, espandere il server che si desidera utilizzare.
2. Espandere **Database**.
3. Espandere il sistema che si desidera gestire
4. Espandere **Transazioni**.
5. Espandere **Transazioni globali** o **Transazioni database**.
6. Fare clic con il tasto destro del mouse sulla transazione da gestire e selezionare **Stato risorsa**.

Visualizzare proprietà transazione

Per visualizzare le proprietà transazione:

1. Nella finestra **iSeries Navigator**, espandere il server che si desidera utilizzare.
2. Espandere **Database**.
3. Espandere il sistema che si desidera gestire
4. Espandere **Transazioni**.
5. Espandere **Transazioni globali** o **Transazioni database**.
6. Fare clic con il tasto destro del mouse sulla transazione da gestire e selezionare **Proprietà**.

Ottimizzare le prestazioni per il controllo sincronizzazione

L'utilizzo del controllo sincronizzazione richiede delle risorse che possano influenzare le prestazioni del sistema. Alcuni fattori influenzano le prestazioni del sistema che riguardano il controllo sincronizzazione. Di seguito vengono riportati fattori che non influenzano le prestazioni, fattori che le peggiorano e fattori che le migliorano.

Fattori che non influenzano le prestazioni

Apertura di un file

Se viene aperto un file senza specificare l'opzione di apertura sincronizzazione, non viene utilizzata nessuna ulteriore risorsa di sistema, anche se è stata avviata una definizione sincronizzazione. Per ulteriori informazioni sulla specifica dell'opzione di apertura sincronizzazione, vedere l'adeguato manuale di riferimento linguaggio ad alto livello.

Fattori che peggiorano le prestazioni

Registrazione su giornale

La registrazione su giornale di un file richiede le risorse di sistema. Tuttavia, nella maggior parte dei casi, la registrazione su giornale viene eseguita in modo migliore con il controllo sincronizzazione piuttosto che senza. Se vengono specificate solo le immagini successive, il controllo sincronizzazione la modifica sia sulle immagini precedenti sia sulle immagini successive mentre il controllo sincronizzazione è in funzione. Solitamente questa è una considerazione sullo spazio e non sulle prestazioni. Vedere l'argomento Gestione giornale per ulteriori informazioni sulla registrazione su giornale.

Operazione di commit

Se vengono apportate delle modifiche alle risorse registrate su giornale durante la transazione, ogni sincronizzazione (commit) di una transazione aggiunge due voci ad ogni giornale correlato a queste risorse. Il numero di voci aggiunte può aumentare in modo significativo per una grande quantità di piccole transazioni. E' possibile inserire i ricevitori del giornale in un lotto dischi separato dai giornali.

Operazione di rollback

Poiché il controllo sincronizzazione deve eseguire il rollback delle modifiche in sospeso registrate nel database, sono necessarie delle risorse di sistema aggiuntive ogni volta che si verifica un rollback. Inoltre, se le modifiche del record sono in sospeso, l'operazione di rollback causa l'aggiunta di ulteriori voci al giornale.

Comandi Avvio controllo sincronia (STRCMTCTL) e ENDCMTCTL (Fine controllo sincronizzazione)

Il sistema incorre in un ulteriore sovraccarico ogni volta che viene avviata o terminata la definizione sincronizzazione tramite, rispettivamente, i comandi STRCMTCTL e ENDCMTCTL. Evitare di utilizzare i comandi STRCMTCTL e ENDCMTCTL per ogni transazione. Utilizzarli solo se necessario. E' possibile stabilire una definizione sincronizzazione all'inizio di un lavoro interattivo e utilizzarla per tutta la durata del lavoro.

Utilizzo di più di un giornale per le transazioni del controllo sincronizzazione

Con la sincronizzazione a due fasi, i file aperti sotto il controllo sincronizzazione possono essere registrati su più di un giornale. Tuttavia, l'utilizzo di più di un giornale comporta l'utilizzo di ulteriori risorse di sistema per gestire la definizione sincronizzazione. L'utilizzo di più di un giornale, inoltre, può rendere più complicato il ripristino.

Blocco record

Il blocco record può interessare altre applicazioni. Il numero dei record bloccati all'interno di un particolare lavoro, aumenta le risorse di sistema generali utilizzate per il lavoro. Per accedere allo stesso record le applicazioni devono attendere la fine della transazione.

Richiesta di SEQONLY(*YES)

Se viene richiesta l'opzione SEQONLY(*YES) (tramite il comando OVRDBF o tramite il tentativo implicito del programma dell'applicazione di utilizzare SEQONLY(*YES)) e il file viene aperto per l'immissione solo sotto il controllo sincronizzazione con LCKLVL(*ALL), l'opzione viene modificata in SEQONLY(*NO). Questa opzione può influire sulle prestazioni dei file di immissione poiché i record non verranno bloccati.

Richiesta di una modifica a livello record per un file database quando è attiva l'elaborazione salva-mentre-attivo

Una richiesta di effettuare una modifica a livello record sotto il controllo sincronizzazione per un file database può essere ritardata se la definizione sincronizzazione si trova su un limite sincronizzazione e l'operazione salva-mentre-attivo è in esecuzione su un lavoro differente. Questo può verificarsi quando un file viene registrato sullo stesso giornale di alcuni oggetti sulla richiesta di salvataggio.

Nota:

Nella colonna Stato sul pannello Gestione lavori attivi (comando WRKACTJOB) viene visualizzato CMTW (attesa sincronizzazione) quando un lavoro viene congelato a causa dell'elaborazione del punto di controllo salva-mentre-attivo.

Commit o rollback delle modifiche quando è attiva l'elaborazione salva-mentre-attivo

Un'operazione di commit o rollback può essere ritardata su un limite sincronizzazione quando un'operazione salva-mentre-attivo è in esecuzione su un lavoro differente. Questo può verificarsi quando una risorsa sincronizzazione API è stata aggiunta precedentemente alla definizione sincronizzazione, a meno che la risorsa API sia stata aggiunta tramite l'API QTNADDCR e il campo *Consentire elaborazione di salvataggio normale* ha un valore di Y.

Poiché il lavoro viene congelato durante la richiesta di commit o rollback e poiché una richiesta di commit o rollback può essere eseguita solo per una singola definizione sincronizzazione alla volta, i lavori con più di una definizione sincronizzazione con risorse sincronizzazione API aggiunte, impediscono sempre il completamento di un'operazione salva-mentre-attivo.

Richiesta di una modifica a livello oggetto quando è attiva l'elaborazione salva-mentre-attivo

Una richiesta di effettuare una modifica a livello oggetto sotto il controllo sincronizzazione può essere ritardata se la definizione sincronizzazione si trova su un limite sincronizzazione e l'operazione salva-mentre-attivo è in esecuzione su un lavoro differente. Questo può verificarsi quando una modifica a livello oggetto viene effettuata mentre l'operazione salva-mentre-attivo è in esecuzione sulla libreria in cui si trova l'oggetto. Ad esempio, l'operazione di creazione tabella SQL sotto il controllo sincronizzazione per la tabella MYTBL nella libreria MYSQLLIB può essere ritardata quando un'operazione salva-mentre-attivo è in esecuzione sulla libreria MYSQLLIB.

Nota:

Se il tempo di attesa supera i 60 secondi, viene inviato un messaggio di interrogazione CPA8351 per domandare all'utente se continuare o annullare l'operazione.

Aggiunta di una risorsa API tramite l'API QTNADDCR

Una richiesta di aggiungere una risorsa sincronizzazione API tramite API QTNADDCR può essere ritardata se le definizioni sincronizzazione del lavoro si trovano su un limite sincronizzazione e l'operazione salva-mentre-attivo è in esecuzione su un lavoro differente.

Note:

1. Se il tempo di attesa supera i 60 secondi, viene inviato un messaggio di interrogazione CPA8351 per domandare all'utente se continuare o annullare l'operazione.
2. Questo non viene applicato alle risorse API aggiunte tramite l'API QTNADDCR se il campo *Consentire elaborazione di salvataggio normale* ha un valore di Y.

Fattori che migliorano le prestazioni**Utilizzo di un giornale predefinito**

L'utilizzo di un giornale predefinito può migliorare le prestazioni se tutti i file sotto il controllo sincronizzazione vengono chiusi e riaperti mentre la definizione sincronizzazione è attiva. Tuttavia, l'utilizzo di un giornale predefinito con OMTJRNE(*NONE) peggiora le prestazioni delle operazioni di commit e rollback.

Selezione dell'ultimo agent

Le prestazioni vengono migliorate quando viene selezionato un ultimo agent poiché sono necessarie meno interazioni tra il sistema e l'ultimo agent durante un'operazione di sincronizzazione. Tuttavia, se si verifica un errore nelle comunicazioni durante un'operazione di sincronizzazione, tale operazione

non verrà completata fino a quando non terminerà la risincronizzazione, a prescindere dal valore dell'opzione attesa risultato. Tale errore è raro ma questa opzione consente al programma di scrittura dell'applicazione di considerare l'impatto negativo dell'attesa indefinita dell'utente per il completamento della risincronizzazione quando si verifica un errore. Ciò dovrebbe essere valutato in base al miglioramento delle prestazioni fornito dall'ottimizzazione dell'ultimo agent durante le operazioni di sincronizzazione con esito positivo. Questa considerazione è più importante per i lavori interattivi che per i lavori batch.

L'impostazione predefinita consente all'ultimo agent di essere selezionato dal sistema ma l'utente può modificare questo valore tramite l'API QTCHGCO.

Non utilizzare l'opzione di attesa risultato

Quando le risorse remote sono sotto il controllo sincronizzazione, le prestazioni vengono migliorate quando l'opzione Attendere risultato è impostata su N (No) e tutti i sistemi remoti supportano la presunta interruzione. L'opzione Attendere risultato viene impostata su N dal sistema per l'applicazione DRDA e DDM quando viene effettuato il primo collegamento ad un sistema remoto. Le applicazioni APPC devono essere impostate esplicitamente sull'opzione Attendere risultato oppure verrà utilizzato il valore predefinito di Y.

Selezione dell'opzione OK per uscire

Le prestazioni vengono migliorate quando viene selezionata l'opzione OK per uscire. Per ulteriori informazioni su questa opzione, vedere Definizione sincronizzazione per sincronizzazione a due fasi: indicare OK per uscire.

Selezione dell'opzione Indicazione di sola lettura

Le prestazioni vengono migliorate quando viene selezionata l'opzione Indicazione di sola lettura. Per ulteriori informazioni su questa opzione, vedere Definizione sincronizzazione per sincronizzazione a due fasi: consentire indicazione di sola lettura.

Le attività possibili per migliorare le prestazioni sono:

- Ridurre i blocchi
- Gestire la dimensione della transazione

Ridurre i blocchi

Un modo tipico di ridurre i blocchi dei record è rilasciare il blocco record. (Questa tecnica non funziona se non viene specificato LCKLVL(*ALL).) Ad esempio, una singola applicazione di manutenzione file generalmente effettua quanto segue:

- Visualizza una richiesta per un'identificazione del record da modificare.
- Richiama il record richiesto.
- Visualizza il record.
- Consente all'utente stazione di lavoro di effettuare la modifica.
- Aggiorna il record.

Nella maggior parte dei casi, il record viene bloccato dall'accesso del record richiesto attraverso l'aggiornamento. Il tempo di attesa del record può essere superato per un altro lavoro in attesa del record. Per evitare il blocco di un record mentre l'utente stazione di lavoro sta effettuando una modifica, rilasciare il record dopo averlo richiamato dal database (prima che il pannello del record venga visualizzato). E' necessario accedere nuovamente al record prima dell'aggiornamento. Se il record è stato modificato tra l'ora in cui è stato rilasciato a quella del nuovo accesso, è necessario informare l'utente stazione di lavoro. Il programma può determinare se il record è stato modificato salvando uno o più campi del record originale e confrontandoli ai campi nello stesso record dopo che è stato richiamato, nel seguente modo:

- Utilizzare un campo conto aggiornamento nel record e aggiungere 1 al campo prima di un aggiornamento. Il programma salva il valore originale e lo confronta con il valore presente nel campo quando il record viene richiamato. Se si è verificata una modifica, l'utente stazione di lavoro viene

informato e il record viene rivisualizzato. Il campo conto aggiornamento viene modificato solo se si verifica un aggiornamento. Il record viene rilasciato mentre l'utente stazione di lavoro sta effettuando una modifica. Se si utilizza questa tecnica, è necessario utilizzarla in tutti i programmi che aggiornano il file.

- Salvare i contenuti del record dati completo e confrontarlo con il record al successivo richiamo.

In entrambi i casi precedenti, la sequenza delle operazioni impedisce il semplice utilizzo dei dati descritti esternamente in RPG laddove gli stessi nomi campi vengono utilizzati nel record principale e nel file di visualizzazione. L'utilizzo degli stessi nomi campi (in RPG) non funziona poiché le modifiche dell'utente stazione di lavoro vengono sostituite quando il record viene richiamato.

E' possibile risolvere questo problema spostando i dati record su una struttura dati o continuare ad utilizzare i dati descritti esternamente se viene utilizzata la parola chiave RTNDDTA DDS. La parola chiave RTNDDTA consente al programma di rileggere i dati sul pannello senza che il sistema operativo sposti i dati dal pannello al programma. Ciò consente al programma di effettuare quanto segue:

1. Richiedere l'identificazione record.
2. Richiamare il record richiesto dal database.
3. Rilasciare il record.
4. Salvare il campo o i campi utilizzati per stabilire se il record è stato modificato.
5. Visualizzare il record e attendere la risposta dell'utente stazione di lavoro.

Se l'utente stazione di lavoro modifica il record sul pannello, il programma utilizza la seguente sequenza:

1. Richiama nuovamente il record dal database.
2. Confronta i campi salvati per determinare se il record database è stato modificato. Se è stato modificato, il programma rilascia il record e invia un messaggio quando viene visualizzato il record.
3. Richiama il record dal pannello eseguendo un'operazione di lettura tramite la parola chiave RTNDDTA e aggiorna il record nel record del database.
4. Passa alla successiva richiesta logica poiché non esistono record aggiuntivi da rilasciare se l'utente stazione di lavoro annulla la richiesta.

LCKLVL(*CHG) e LCKLVL(*CS) funzionano in questa situazione. Se viene utilizzato LCKLVL(*ALL), è necessario rilasciare il blocco record tramite un'operazione di commit o rollback.

Per ulteriori informazioni sui blocchi, vedere Rilevare stalli.

Gestire la dimensione della transazione

Per questo argomento, la transazione è interattiva. (Il controllo sincronizzazione può essere utilizzato anche per le applicazioni batch, che spesso possono essere considerate una serie di transazioni. Molte delle stesse considerazioni si riferiscono alle applicazioni batch, trattate in Controllo sincronizzazione per le applicazioni batch.)

E' possibile bloccare un massimo di 500 000 000 di record durante una transazione per ogni giornale associato alla transazione. E' possibile ridurre questo limite tramite un QAQQINI (File opzioni interrogazioni). Utilizzare il parametro QRYOPTLIB del comando CHGQRYA (Modifica attributi interrogazioni) per specificare un File opzioni interrogazioni per un lavoro da utilizzare. Utilizzare il valore COMMITMENT_CONTROL_LOCK_LEVEL nel File opzioni interrogazioni come limite blocco per il lavoro.

Quando viene scelto il livello di blocco per i record, considerare la dimensione della transazione. La dimensione dovrebbe determinare per quanto tempo i record vengono bloccati prima della fine di una transazione. E' necessario decidere se un'operazione di commit o rollback per il controllo sincronizzazione è limitata al singolo utilizzo del tasto Invio o se la transazione è costituita da molti utilizzi del tasto Invio.

Nota:

Più è breve la transazione e prima il lavoro in attesa di avviare l'elaborazione del punto di controllo salva-mentre-attivo potrà continuare ed essere completato.

Ad esempio, per un'applicazione di immissione dell'ordine, un cliente potrebbe ordinare alcune voci in un unico ordine richiedendo un record dettagliato dell'ordine e un aggiornamento record principale inventario per ogni voce presente nell'ordine. Se la transazione viene definita come ordine completo ed ogni utilizzo del tasto Invio ordina una voce, tutti i record coinvolti nell'ordine vengono bloccati per tutta la durata dell'ordine completo. Quindi, i record utilizzati di frequente (come ad esempio i record principali inventario) potrebbero essere bloccati per lunghi periodi di tempo, impedendo l'elaborazione di altri lavori. Se tutte le voci vengono immesse con un singolo tasto Invio tramite un file secondario, la durata dei blocchi per l'ordine completo viene ridotta.

In generale, il numero e la durata dei blocchi dovrebbero essere ridotti in modo tale che alcuni utenti stazione di lavoro possano accedere agli stessi dati senza lunghi periodi di attesa. Ciò è possibile evitando di porre dei blocchi mentre l'utente immette i dati sul pannello. Alcune applicazioni non richiedono che più di un utente stazione di lavoro acceda agli stessi dati. Ad esempio, in un'applicazione di invio contanti con molti record voci aperti per cliente, l'approccio tipico è quello di bloccare tutti i record e ritardarli fino a quando un utente stazione di lavoro non completa l'invio di contanti per una ricevuta specificata.

Se l'utente stazione di lavoro preme il tasto Invio alcune volte per una transazione, è possibile eseguire la transazione in alcuni segmenti. Ad esempio:

- Il primo segmento è un'interrogazione in cui l'utente stazione di lavoro richiede le informazioni.
- Il secondo segmento è una conferma della possibilità dell'utente stazione di lavoro di completare l'intera transazione.
- Il terzo segmento è il richiamo e l'aggiornamento dei record interessati.

Questo approccio consente la limitazione del blocco record ad un unico utilizzo di Invio.

Questo approccio prima-interrogazione normalmente viene utilizzato nelle applicazioni in cui una decisione risulta dalle informazioni visualizzate. Ad esempio, in un'applicazione di prenotazione di biglietti aerei, un cliente potrebbe desiderare gli orari dei voli, le coincidenze e i posti disponibili prima di decidere quale volo prendere. Una volta presa questa decisione viene immessa la transazione. Se la transazione non riesce (il volo è completo), è possibile utilizzare la funzione di rollback e immettere una differente richiesta. Se i record sono stati bloccati dalla prima richiesta fino a quando è stata effettuata la decisione, un altro addetto alle prenotazioni attenderà fino al completamento della transazione.

Scenari ed esempi: controllo sincronizzazione

Di seguito vengono riportati uno scenario e gli esempi di controllo sincronizzazione. Lo scenario descrive ad alto livello come la società di giocattoli JKL implementa un controllo sincronizzazione per tracciare le transazioni sui relativi database locali.

Questi esempi forniscono il codice di esempio per il controllo sincronizzazione. Il Problema pratico è un programma RPG che implementa il controllo sincronizzazione. Comprende un flusso logico che descrive ciò che accade durante ogni fase.

I tre esempi seguenti descrivono gli usi del controllo sincronizzazione per avviare le applicazioni dopo una fine anomala del sistema.

Scenario

- Scenario: controllo sincronizzazione

Esempi

- Problema pratico per il controllo sincronizzazione
- Flusso logico del problema pratico per il controllo sincronizzazione
- Esempio: utilizzare un file di registrazione transazione per avviare un'applicazione
- Esempio: utilizzare un oggetto di notifica per avviare un'applicazione
- Esempio: utilizzare un programma di elaborazione standard per avviare un'applicazione

Nota: leggere la clausola di esonero della responsabilità inerente all'esempio di codice per quanto riguarda le informazioni legali rilevanti.

Scenario: controllo sincronizzazione

La società di giocattoli JKL utilizza il controllo sincronizzazione per proteggere i record database per la fabbricazione e l'inventario. Questo scenario, visualizza ad un alto livello il modo in cui la società di giocattoli JKL utilizza il controllo sincronizzazione quando una parte viene trasferita dal dipartimento inventario al dipartimento fabbricazione.

Vedere Scenario: gestione giornale per la descrizione dell'ambiente di rete della società di giocattoli JKL. Il seguente scenario descrive il modo in cui il controllo sincronizzazione funziona sul server produzione, JKLPROD.

Questo scenario illustra i vantaggi dell'utilizzo del controllo sincronizzazione nei due esempi. Il primo esempio descrive come il programma di inventario dell'azienda, Programma A, potrebbe funzionare senza il controllo sincronizzazione e i possibili problemi che si potrebbero verificare. Il secondo esempio descrive il modo in cui funziona il programma con il controllo sincronizzazione.

La società di giocattoli JKL utilizza un programma dell'applicazione inventario, Programma A, sul server JKLPROD. Il programma A utilizza due record. Un record traccia gli elementi che sono memorizzati nel magazzino. Un altro record mantiene la traccia degli elementi eliminati dal magazzino e utilizzati nella produzione.

Programma A senza controllo sincronizzazione

Il seguente programma dell'applicazione non utilizza il controllo sincronizzazione. Il sistema blocca i record letti per l'aggiornamento. Le seguenti fasi descrivono il modo in cui il programma dell'applicazione traccia il diodo quando viene eliminato dal magazzino e trasferito sul conto corrente:

- Il programma A blocca e recupera il record magazzino. (Questa azione potrebbe richiedere del tempo di attesa se il record viene bloccato da un altro programma.)
- Il programma A blocca e richiama il record produzione. (Questo potrebbe richiedere del tempo di attesa.) Entrambi i record del programma A sono bloccati e non possono essere modificati da nessun altro programma.
- Il programma A aggiorna il record magazzino. Questo causa il rilascio del record, rendendolo disponibile per essere letto per l'aggiornamento da qualsiasi altro programma.
- Il programma A aggiorna il record produzione, causando il rilascio del record e rendendolo disponibile per essere letto per l'aggiornamento da qualsiasi altro programma.

Senza l'utilizzo del controllo sincronizzazione, è necessario risolvere un problema per consentire il corretto funzionamento del programma in tutte le circostanze. Ad esempio, un problema si verifica se il programma A non aggiorna entrambi i record a causa di un errore di lavoro o di sistema. In questo caso, i due file non sono coerenti, e i diodi sono eliminati dal record magazzino ma non vengono aggiunti al record produzione. L'utilizzo del controllo sincronizzazione garantisce che tutte le modifiche coinvolte nella transazione vengano completate o che i file vengano restituiti allo stato originale nel caso in cui l'elaborazione della transazione venga interrotta.

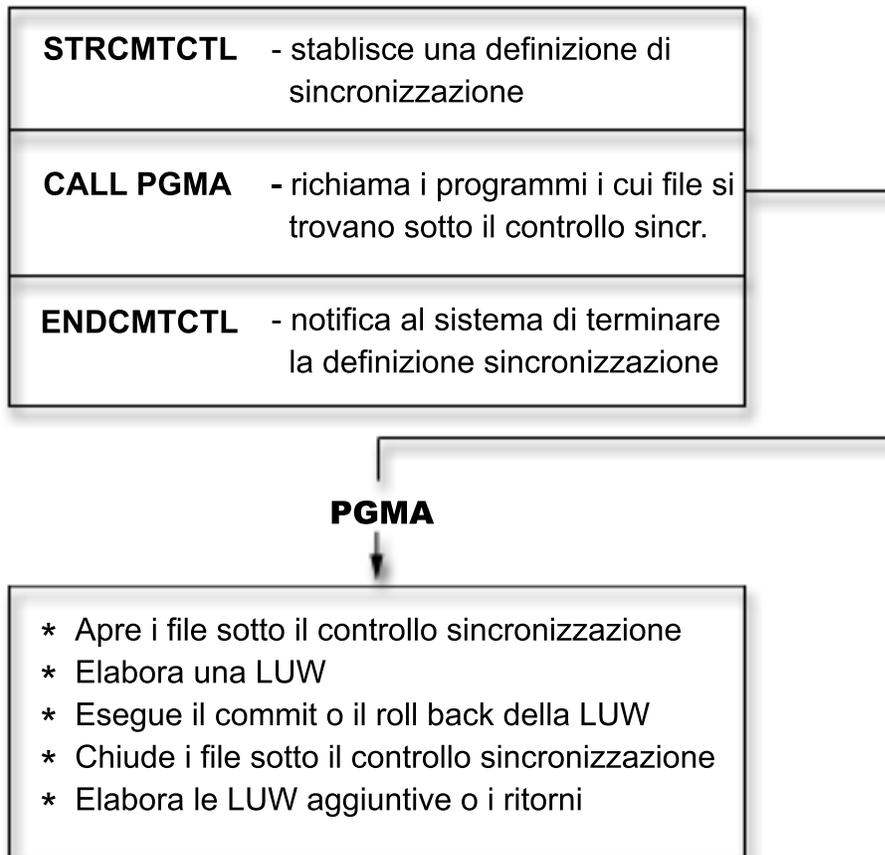
Programma A con controllo sincronizzazione

Se viene utilizzato il controllo sincronizzazione, l'esempio precedente viene modificato nel seguente modo:

1. Il controllo sincronizzazione viene avviato.
2. Il programma A blocca e recupera il record magazzino. (Quest'azione potrebbe richiedere del tempo di attesa se il record viene bloccato da un altro programma.)
3. Il programma A blocca e richiama il record produzione. (Questo potrebbe richiedere del tempo di attesa.) Entrambi i record del programma A sono bloccati e non possono essere modificati da nessun altro programma.
4. Il programma A aggiorna il record magazzino e il controllo sincronizzazione conserva il blocco sul record.
5. Il programma A aggiorna il record produzione, e il controllo sincronizzazione conserva il blocco sul record.
6. Il programma A sincronizza la transazione. Le modifiche apportate al record magazzino e al record produzione vengono rese permanenti nei file. Tali modifiche vengono registrate nel giornale, e verranno visualizzate sul disco. Il controllo sincronizzazione rilascia i blocchi su entrambi i record. I record sono ora disponibili per essere letti per l'aggiornamento da qualsiasi altro programma.

Poiché i blocchi su entrambi i record vengono mantenuti dal controllo sincronizzazione fino a quando la transazione non viene sincronizzata, non può verificarsi una situazione in cui un record viene aggiornato e l'altro no. Se si verifica un errore nella fase di instradamento o un errore di sistema prima che la transazione venga sincronizzata, il sistema elimina (rollback) le modifiche apportate, in modo tale che i file vengano aggiornati fino al punto in cui è stata sincronizzata l'ultima transazione.

Per ogni fase di instradamento in cui i file devono trovarsi sotto il controllo sincronizzazione, si verificano le fasi visualizzate nella seguente figura:



Le operazioni eseguite sotto il controllo sincronizzazione vengono registrate su giornale. La voce giornale controllo sincronizzazione di avvio viene visualizzata dopo la prima voce di apertura file sotto il controllo sincronizzazione. Questo perché la prima voce di apertura file determina quale giornale viene utilizzato per il controllo sincronizzazione. La voce di giornale proveniente dalla prima operazione di apertura viene, quindi, utilizzata per controllare successive operazioni di apertura per assicurarsi che tutti i file stiano utilizzando lo stesso giornale.

Quando si verifica un errore di lavoro o di sistema, le risorse sotto il controllo sincronizzazione vengono aggiornate su un limite di sincronizzazione. Se una transazione viene avviata ma non completata prima della fine di una fase di instradamento, il sistema esegue il rollback della transazione e dopo la fine della fase di instradamento la transazione non viene visualizzata nel file. Se il sistema termina in modo anomalo prima che una transazione venga completata, il sistema esegue il rollback della transazione e quest'ultima non viene visualizzata nel file in seguito ad un successivo IPL (initial program load) del codice interno su licenza con esito positivo. Ogni volta che si verifica un rollback, le voci di inversione vengono ubicate nel giornale.

Ad esempio, si supponga che la società JKL possieda 100 diodi in magazzino. La fabbricazione ne preleva 20 dal magazzino, per un nuovo equilibrio di 80. L'aggiornamento database determina le voci giornale immagine precedente (100) e immagine successiva (80).

Si supponga che il sistema termina in modo anomalo dopo la registrazione su giornale delle voci, ma prima di aver raggiunto il punto di commit o di rollback. In seguito all'IPL, il sistema legge la voce giornale

e aggiorna il record database corrispondente. Questo aggiornamento produce due voci giornale che invertono l'aggiornamento: la prima voce è immagine precedente (80) e la seconda è immagine successiva (100).

Quando l'IPL viene completato con esito positivo, il sistema elimina (rollback) eventuali modifiche database non sincronizzate. Nell'esempio precedente, il sistema elimina le modifiche dal record magazzino poiché nel giornale non esiste un'operazione sincronizzazione per tale transazione. In questo caso, l'immagine precedente del record magazzino viene ubicata nel file. Il giornale contiene le modifiche sottoposte a rollback, e un'indicazione che si è verificata un'operazione di rollback.

Problema pratico per il controllo sincronizzazione

Questo problema pratico faciliterà la comprensione del controllo sincronizzazione e dei relativi requisiti. Le seguenti operazioni suppongono la conoscenza del programma su licenza OS/400 e della DFU (Data file utility) e la lettura di questo argomento. Il Flusso logico consentirà un'ulteriore comprensione del programma pratico per il controllo sincronizzazione.

Nota: leggere la clausola di esonero della responsabilità inerente all'esempio di codice per quanto riguarda le informazioni legali rilevanti.

Prima di iniziare, effettuare le seguenti operazioni:

- Creare una libreria speciale per questo problema pratico. Nelle istruzioni, la libreria viene denominata CMTLIB. Sostituire il nome della libreria dove viene visualizzato CMTLIB.
- Creare i file sorgente e una descrizione del lavoro.

Effettuare le seguenti operazioni:

1. Creare un file fisico denominato ITMP (file master voci). La DDS (data description specification) per questo file è:

```
10  A  R  ITMR
20  A  ITEM      2
30  A  ONHAND   5  0
40  A  K  ITEM
```

2. Creare un file fisico denominato TRNP (file di transazione). Questo file viene utilizzato come file di registrazione transazione. La DDS per questo file è:

```
10  A  R  TRNR
20  A  QTY      5  0
30  A  ITEM      2
40  A  USER     10
```

3. Creare un file logico denominato TRNL (transazione logica). Questo file viene utilizzato per l'assistenza durante il riavvio dell'applicazione. Il campo *USER* è la sequenza di tipo LIFO. La DDS per questo file è:

```
10                                LIFO
20  A  R  TRNR      PFILE (TRNP)
30  A  K  USER
```

4. Immettere il comando STRDFU e creare un'applicazione DFU denominata ITMU per il file ITMP. Accettare le impostazioni predefinite offerte da DFU durante la definizione dell'applicazione.
5. Digitare il comando CHGDTA ITMU ed immettere i seguenti record per il file ITMP:

| Elemento | Disponibilità |
|----------|---------------|
| AA | 450 |
| BB | 375 |
| CC | 4000 |

6. Chiudere il programma tramite F3. Questa voce fornisce alcuni dati sui quali agisce il programma.
7. Creare il programma CL ITMPCSC (Item Process) nel seguente modo:

```

PGM
DCL &USER *CHAR LEN(10)
RTVJOBA USER(&USER)
CALL ITMPCS PARM(&USER)
ENDPGM

```

E' il programma di controllo che richiama il programma ITMPCS. Richiama il nome utente e lo inoltra al programma di elaborazione. Questa applicazione implica l'utilizzo di nomi utenti univoci.

- Creare un file di visualizzazione denominato ITMPCSD dalla DDS come segue.

Esistono due formati, il primo per la visualizzazione delle richieste di base e il secondo per consentire all'operatore di rivisualizzare l'ultima transazione immessa. Questo file di visualizzazione viene utilizzato dal programma ITMPCS.

```
SEQNBR *... .. 1 ... .. 2 ... .. 3 ... .. 4 ... .. 5 ... .. 6 ... .. 7 ..
```

```

1.00      A          R PROMPT
2.00      A
3.00      A          CA03(93 'End of program')
4.00      A          CA04(94 'Review last')
5.00      A          SETOFF(64 'No rcd to rvw')
6.00      A          1 2'INVENTORY TRANSACTIONS'
7.00      A          3 2'Quantity'
8.00      A 61      QTY          5 0I  +1
9.00      A          ERRMSG('Invalid +
10.00     A          quantity' 61)
11.00     A          +5'ITEM'
12.00     A 62      ITEM          2   I  +1
13.00     A          ERRMSG('Invalid +
14.00     A 63      Item number' 62)
15.00     A          ERRMSG('Rollback +
16.00     A 64      occurred' 63)
17.00     A          24 2'CF4 was pressed and +
18.00     A          there are no +
19.00     A          transactions for +
20.00     A          this user'
21.00     A          DSPATR(HI)
22.00     A          23 2'CF4 Review last +
23.00     A          transaction'
24.00     A          R REVW
25.00     A          1 2'INVENTORY TRANSACTIONS'
26.00     A          +5'REVIEW LAST TRANSACTION'
27.00     A          3 2'Quantity'
28.00     A          QTY          5 0  +1EDTCDE(Z)
29.00     A          ITEM          2   +1

```

- Studiare il flusso logico fornito in Flusso logico del programma pratico per il controllo sincronizzazione.

- Immettere il comando STRSEU e digitare il sorgente nel seguente modo:

```
SEQNBR *... .. 1 ... .. 2 ... .. 3 ... .. 4 ... .. 5 ... .. 6 ... .. 7 ..
```

```

1.00      FITMP    UF E          K          DISK
2.00      F*
3.00      FTRNP   0 E          DISK
4.00      F*
5.00      FTRNL   IF E          K          DISK
6.00      F        TRNR
7.00      FITMPCSD CF E          WORKSTN
8.00      C* Enter parameter with User name for -TRNP- file
9.00      C          *ENTRY    PLIST
10.00     C          PARM        USER    10
11.00     C          LOOP      TAG
12.00     C          EXFMTPROMPT
13.00     C* Check for CF3 for end of program
14.00     C 93      DO
15.00     C          SETON      LR

```

```

16.00 C RETRN
17.00 C END
18.00 C* Check for CF4 for review last transaction
19.00 C 94 DO Review last
20.00 C* Check for existence of a record for this user in -TRNL- file
21.00 C USER CHAINTRNR1 64 Not found
22.00 C 64 GOTO LOOP
23.00 C EXFMTREVV
24.00 C GOTO LOOP
25.00 C END
26.00 C* Access Item record
27.00 C ITEM CHAINITMR 62 Not found
28.00 C* Handle -not found- Condition
29.00 C 62 GOTO LOOP
30.00 C* Does sufficient quantity exist
31.00 C ONHAND SUB QTY TEST 50 61 Minus
32.00 C* Handle insufficient quantity
33.00 C 61 DO
34.00 C* Release Item record which was locked by the CHAIN for update
35.00 C EXCPTRLSITM
36.00 C GOTO LOOP
37.00 C END
38.00 C* Change ONHAND and update the Item record
39.00 C Z-ADDTEST ONHAND
40.00 C UPDATITMR
41.00 C* Test for Special Simulation Conditions
42.00 C ITEM IFEQ 'CC'
43.00 C* Simulate program need for rollback
44.00 C QTY IFEQ 100
45.00 C SETON 63 Simult Rlback
46.00 C* ROLBK
47.00 C GOTO LOOP
48.00 C END
49.00 C* Simulate an abnormal program cancellation by Div by zero
50.00 C* Operator Should respond -C- to inquiry message
51.00 C QTY IFEQ 101
52.00 C Z-ADD0 ZERO 30
53.00 C TESTZ DIV ZERO TESTZ 30 Msg occurs
54.00 C END
55.00 C* Simulate an abnormal job cancellation by DSPLY.
56.00 C* Operator Should System Request to another job
57.00 C* and cancel this one with OPTION(*IMMED)
58.00 C QTY IFEQ 102
59.00 C 'CC=102' DSPLY Msg occurs
60.00 C END
61.00 C END ITEM=CC
62.00 C* Write the -TRNP- file
63.00 C WRITETRNR
64.00 C* Commit the update to -ITMP- and write to -TRNP-
65.00 C* COMMIT
66.00 C GOTO LOOP
67.00 OITMR E RLSITM

```

11. Immettere il comando CRTRPGPGM per creare il programma ITMPCS dal sorgente immesso nell'operazione precedente.
12. Digitare il comando CALL ITMPCSC, premere Invio e premere F4. Dovrebbe essere visualizzato un messaggio che indica che non esistono voci per questo operatore.
13. Immettere i seguenti dati per visualizzare se il programma funziona correttamente:

| Quantità | Elemento |
|----------|----------|
| 3 | AA |
| 4 | BB |

14. Premere F4. Il pannello di rivisualizzazione comparirà con l'ultima voce BB immessa. Immettere i seguenti dati:

| Quantità | Elemento |
|----------|--|
| 5 | FF (Dovrebbe essere visualizzato un messaggio di numero elementi non valido.) |
| 9000 | BB (Dovrebbe essere visualizzato un messaggio di errore quantità insufficiente.) |
| 100 | CC (Dovrebbe essere visualizzato un messaggio di rollback.) |
| 102 | CC (Dovrebbe verificarsi un'operazione RPG DSPLY. Premere il tasto Invio.) |
| 101 | CC (Il programma dovrebbe visualizzare un messaggio di interrogazione indicando che si è verificata o è terminata una condizione di divisione per zero, in base all'impostazione dell'attributo lavoro INQMSGRPY. Se viene visualizzato il messaggio di interrogazione, immettere C per annullare il programma RPG e, successivamente, C per annullare il programma CL sull'interrogazione successiva. In questo modo viene simulata una condizione di errore imprevisto.) |

15. Digitare il comando DSPDTA (Visualizzazione dati) ITMP.

Controllare se i record AA e BB sono stati aggiornati correttamente. I valori devono essere AA = 447, BB = 371 e CC = 3697. Notare che le quantità sottratte da CC sono state visualizzate ma i record delle transazioni non sono stati scritti.

16. Creare un ricevitore giornale per il controllo sincronizzazione. Utilizzare il comando CRTJRNRCV (Creazione ricevitore giornale) denominato RCVR1 nella libreria CMRLIB. Specificare una soglia di almeno 5000KB. Una soglia maggiore è consigliata se il sistema ha spazio sufficiente per aumentare il tempo tra la generazione di nuovi ricevitori giornale per ridurre gli impatti sulle prestazioni dei giornali di modifiche frequenti.

17. Creare un giornale per il controllo sincronizzazione. Utilizzare il comando CRTJRN (Creazione giornale) per creare un giornale denominato JRNTEST nella libreria CMTLIB. Poiché questo giornale viene utilizzato solo per il controllo sincronizzazione, specificare MNGRCV(*SYSTEM) DLTRCV(*YES). Per il parametro JRNRCV, specificare il ricevitore del giornale creato nella fase 16 (Vedere pagina 79).

18. Utilizzare il comando STRJRNPF (Avvio file fisico del giornale) con i parametri FILE(CMTLIB/ITMP CMTLIB/TRNP) JRN(CMTLIB/JRNTEST) per registrare su giornale i file da utilizzare per il controllo sincronizzazione.

Il parametro IMAGES utilizza l'impostazione predefinita di *AFTER, che significa che vengono visualizzate nel giornale solo le modifiche delle immagini successive dei record. I file ITMP e TRNP hanno avviato ora la registrazione su giornale.

Solitamente, i file vengono salvati dopo l'avvio della registrazione su giornale. Non è possibile applicare le modifiche registrate su giornale ad un file ripristinato che non possiede lo stesso JID delle voci di giornale. Poiché questo problema pratico non richiede l'applicazione delle modifiche registrate su giornale, è possibile evitare di salvare i file registrati.

19. Digitare il comando CALL ITMPCSC ed immettere le seguenti transazioni:

| Quantità | Elemento |
|----------|----------|
| 5 | AA |
| 6 | BB |

Chiudere il programma premendo F3.

20. Digitare il comando di Visualizzazione giornale: DSPJRN CMTLIB/JRNTEST.

Notare le voci visualizzate sul giornale. La stessa sequenza di voci (R UP = aggiornamento di ITMP seguito da R PT = record aggiunto a TRNP) viene visualizzata nel giornale mentre veniva eseguita dal programma. Questo si verifica poiché un file logico viene definito sul file fisico TRNP e il sistema sostituisce il valore predefinito RPG. Se non fosse esistito nessun file logico, sarebbe stata utilizzata

la supposizione RPG di SEQONLY(*YES) e sarebbe stato visualizzato un blocco delle voci PT poiché i record sarebbero rimasti nel buffer fino al riempimento del blocco.

21. Modificare il programma CL ITMPCSC nel seguente modo (le nuove istruzioni sono indicate con un asterisco):

```

PGM
DCL &USER *CHAR LEN(10)
RTVJOBA USER(&USER)
*   STRCMTCTL LCKLVL(*CHG)
   CALL ITMPCS PARM(&USER)
*   MONMSG MSGID(RPG9001) EXEC(ROLLBACK)
*   ENDCMTCTL
ENDPGM

```

Il comando STRCMTCTL imposta l'ambiente di controllo sincronizzazione. La parola LCKLVL specifica che i record letti per l'aggiornamento ma non aggiornati, possono essere rilasciati durante la transazione. Il comando MONMSG gestisce qualsiasi messaggio di uscita RPG ed esegue un ROLLBACK in caso di fine anomala del programma RPG. Il comando ENDCMTCTL chiude l'ambiente di controllo sincronizzazione.

22. Cancellare il programma ITMPCSC esistente e ricrearlo.
23. Modificare il programma RPG per eliminare i simboli dei commenti sulle istruzioni 2.00, 4.00, 46.00 e 65.00. Ora, è possibile utilizzare il sorgente con il controllo sincronizzazione.
24. Cancellare il programma ITMPCS esistente e ricrearlo. Il programma, ora, può essere in funzione sotto il controllo sincronizzazione.
25. Digitare il comando CALL ITMPCSC ed immettere le seguenti transazioni:

| Quantità | Elemento |
|----------|----------|
| 7 | AA |
| 8 | BB |

26. Utilizzare Richiesta di sistema e richiedere l'opzione per visualizzare il lavoro corrente. Quando viene visualizzato il pannello Visualizzazione lavoro, selezionare l'opzione 16 per richiedere il pannello dello stato del controllo sincronizzazione.

Notare i valori sul pannello. Dovrebbero esistere due sincronizzazioni poiché sono state eseguite due istruzioni di sincronizzazione nel programma.

27. Premere F9 per visualizzare una lista di file sotto il controllo sincronizzazione e il numero di attività per ogni file.
28. Ritornare al programma e chiuderlo premendo F3.
29. Digitare DSPJRN CMTLIB/JRNTEST e notare le voci per i file e le voci giornale speciali per il controllo sincronizzazione:

| | |
|------|---|
| C BC | E' stato eseguito il comando STRCMTCTL. |
| C SC | Avvio ciclo di sincronizzazione. Ciò si verifica ogni volta che la prima operazione database nella transazione determina l'inserimento di un record, il relativo aggiornamento o la relativa cancellazione come parte del controllo sincronizzazione. |
| C CM | Si è verificata un'operazione di sincronizzazione. |
| C EC | E' stato eseguito il comando ENDCMTCTL. |

Le immagini precedenti e le immagini successive del controllo sincronizzazione (tipi R UB e R UP) si verificano automaticamente anche se originariamente è stato richiesto IMAGES(*AFTER) per il giornale.

30. Digitare il comando CALL ITMPCSC ed immettere le seguenti transazioni:

| Quantità | Elemento |
|-----------------|---|
| 12 | AA |
| 100 | CC (E' la condizione per simulare la necessità di un'applicazione di utilizzare il rollback. Il record CC nel file ITMP, aggiornato dall'istruzione RPG 40.00 viene sottoposto a rollback.) |

31. Premere F4 per determinare l'ultima transazione immessa.

L'ultima transazione sincronizzata è la voce per l'elemento AA.

32. Utilizzare Richiesta di sistema e richiedere l'opzione Visualizzazione lavoro corrente. Quando viene visualizzato il pannello Visualizzazione lavoro, richiedere il pannello dello stato del controllo sincronizzazione.

Notare i valori sul pannello e il modo in cui vengono modificati tramite rollback.

33. Ritornare al programma.

34. Ritornare al pannello delle richieste di base e chiudere il programma premendo F3.

35. Digitare il comando DSPJRN CMTLIB/JRNTEST.

Notare le voci aggiuntive che vengono visualizzate nel giornale per l'utilizzo della voce rollback (voce C RB). Quando viene eseguito il rollback del record ITMP, tre voci vengono ubicate nel giornale. Ciò si verifica poiché qualsiasi modifica al file database sotto il controllo sincronizzazione produce una voce precedente (R BR) e una successiva (R UR).

36. Visualizzare le voci con il codice giornale R e questi tipi di voci: UB, UP, BR e UR. Utilizzare l'opzione 5 per visualizzare le voci complete. Poiché il campo *Quantità* è compreso decimale, utilizzare F11 per richiedere la visualizzazione esadecimale. Notare quanto segue:

- Il valore disponibile del record ITMP nel record UB
- Come viene ridotto il valore disponibile tramite il record UP
- In che modo il record BR è uguale al record UP
- In che modo il record UR restituisce il valore come visualizzato originariamente per il record UB

L'ultima voce è la voce RB per la fine del rollback.

37. Digitare il comando CALL ITMPCSC, premere Invio e premere F4. Notare l'ultima transazione immessa.

38. Digitare le seguenti transazioni:

| Quantità | Elemento |
|-----------------|---|
| 13 | AA |
| 101 | CC (E' la condizione per simulare una condizione di errore imprevisto, che causa la fine del programma. Tale simulazione si verifica dividendo un campo per 0. Il programma visualizzerà un messaggio di interrogazione o terminerà, in base all'impostazione dell'attributo lavoro INQMSGRPY. Se viene visualizzato un messaggio di interrogazione, immettere C per chiudere il programma. Poiché il programma CL è stato modificato per monitorare gli errori del programma RPG, non verrà visualizzata la seconda interrogazione.) |

39. Digitare il comando DSPJRN CMTLIB/JRNTEST.

Si è verificato lo stesso tipo di gestione di rollback, ma questa volta il rollback è stato causato dal parametro EXEC del comando MONMSG nel programma CL, invece del programma RPG. Visualizzare le due voci RB per individuare il programma che le ha causate.

40. Digitare il comando WRKJOB e scrivere il nome lavoro completo da utilizzare successivamente.

41. Digitare il comando CALL ITMPCSC ed immettere la seguente transazione:

| Quantità | Elemento |
|----------|---|
| 14 | AA |
| 102 | CC (Dovrebbe verificarsi un'operazione RPG DSPLY sulla coda messaggi esterna. Utilizzare il tasto Richiesta di sistema e selezionare l'opzione 1 sul menu richiesta di sistema per il trasferimento su un lavoro secondario.) |

42. Collegarsi al lavoro secondario e ristabilire l'ambiente.

43. Digitare il comando ENDJOB e specificare il nome lavoro completo precedentemente identificato e OPTION(*IMMED). In questo modo viene simulata una fine anomala del lavoro o del sistema.

44. Attendere circa 30 secondi, digitare il comando CALL ITMPCSC e premere F4.

Notare l'ultima transazione sincronizzata. Dovrebbe essere la voce AA precedentemente immessa.

45. Ritornare al pannello delle richieste di base e chiudere il programma premendo F3.

46. Digitare il comando DSPJRN CMTLIB/JRNTEST.

Si è verificato lo stesso tipo di gestione di rollback ma questa volta il rollback è stato causato dal sistema invece che da uno dei programmi. La voce RB è stata scritta dal programma QWTPITPP, che è il programma di fine anomala di gestione lavoro.

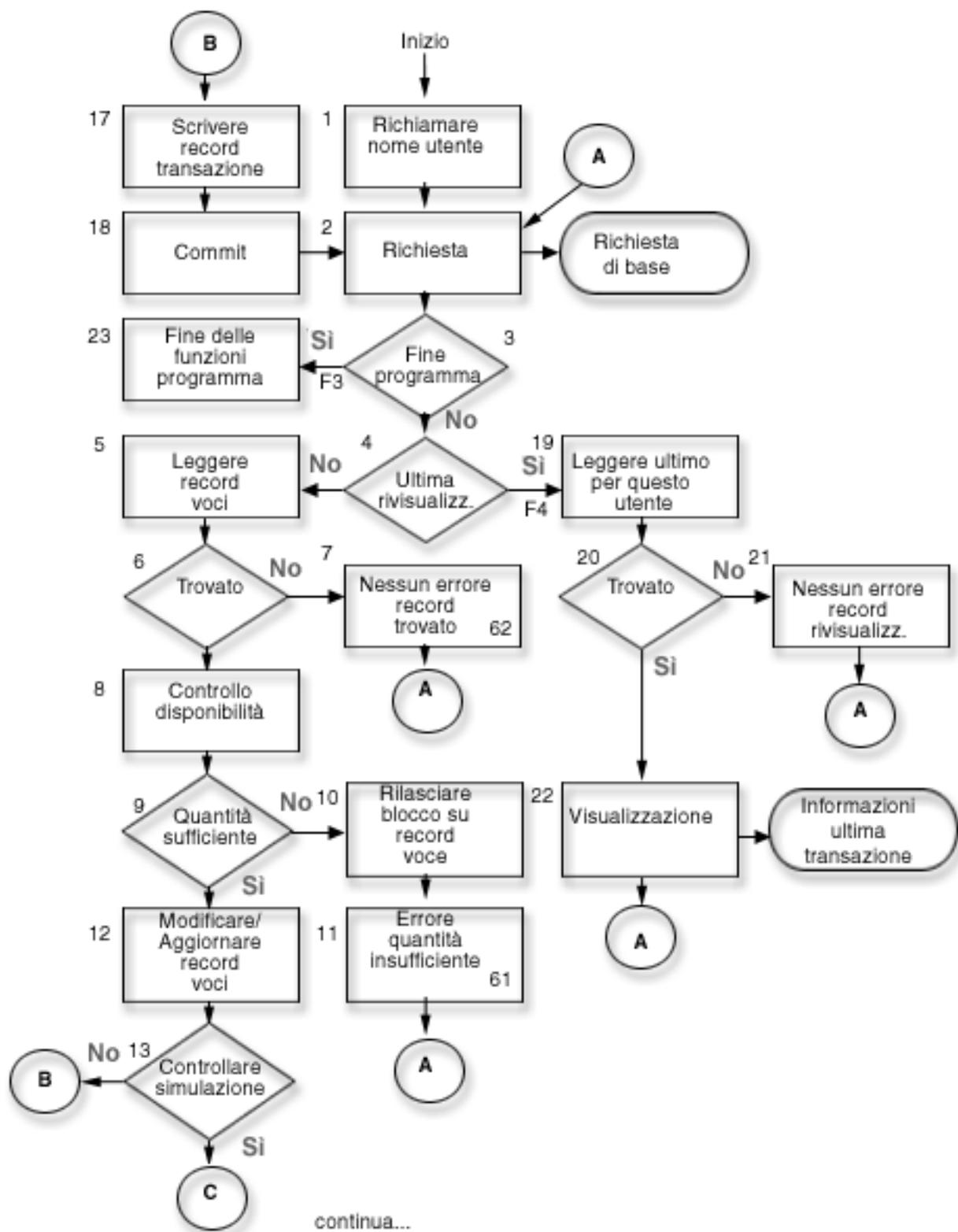
Sono state utilizzate le funzioni di base del controllo sincronizzazione. E' possibile procedere con il controllo sincronizzazione sulle applicazioni o tentare altre funzioni come ad esempio:

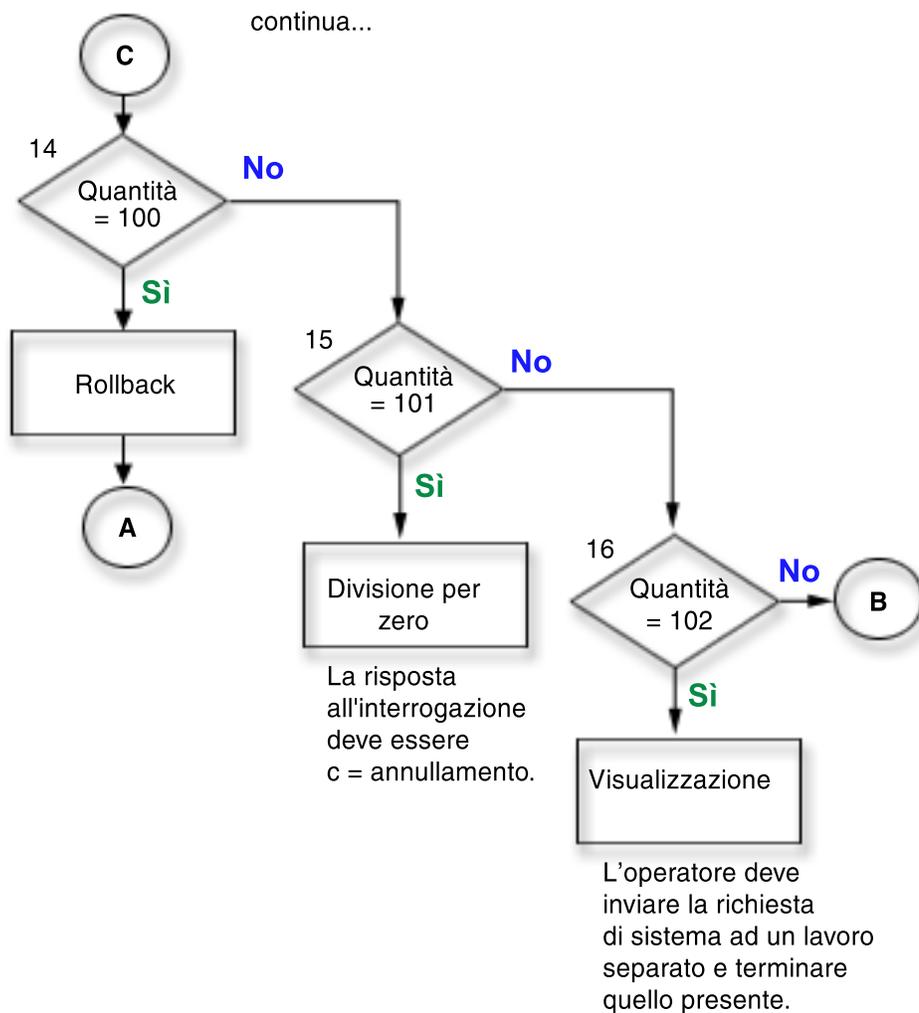
- Utilizzare un oggetto di notifica
- Bloccare i record che sono di sola lettura tramite LCKLVL(*ALL)
- Bloccare più record nello stesso file tramite LCKLVL(*ALL)

Flusso logico di un problema pratico

Nella seguente immagine viene illustrato il flusso del Problema pratico del controllo sincronizzazione. Fare clic sulle figure con un numero per visualizzare ogni fase associata al flusso logico mentre questo è in corso. Inoltre, è possibile andare a Fasi associate al flusso logico del programma pratico per visualizzare

tutte le fasi.





Fasi associate al flusso logico del problema pratico

Le seguenti fasi sono associate al flusso logico del problema pratico.

1. Richiamare il nome utente inoltrato come un parametro. Viene utilizzato per scrivere sul file TRNP e per richiamare l'ultima transazione immessa da ogni operatore. Questa applicazione utilizza nomi utente univoci per gli operatori.
2. Richiedere il pannello principale tramite il nome di formato PROMPT.
3. Se viene premuto F3, avviare una funzione di fine del programma.
4. Se viene premuto F4, avviare una routine per accedere all'ultima transazione immessa dall'operatore.
5. Leggere il record voci tramite il campo *ITEM*. Poiché il file è un file di aggiornamento, questa richiesta blocca il record.
6. Ricercare una condizione non trovata nel file ITMP.
7. Se non esiste nessun record ITMP, impostare sull'indicatore 62 per visualizzare il messaggio di errore e ritornare alla fase 2 (Vedere pagina 84).
8. Sottrarre QTY (Quantità richiesta) da ONHAND (Equilibrio disponibile) nell'area di lavoro.
9. Controllare se esiste quantità sufficiente per rispondere alla richiesta.
10. Se la quantità esistente è insufficiente, rilasciare il blocco sul record nel file ITMP. Questa fase è necessaria a causa della quantità insufficiente.

11. Impostare su indicatore 61 per indicare un messaggio di errore di visualizzazione quantità insufficiente e ritornare alla fase 2 (Vedere pagina 84).
12. Modificare il campo ONHAND per il nuovo equilibrio e aggiornare il record ITMR.
13. Ricercare la voce speciale nel campo ITEM che può essere utilizzata per simulare le condizioni in cui è necessario il ROLLBACK.
14. Ricercare QTY=100. Emettere un'operazione di ROLLBACK. Questa simula una condizione in cui il programma necessita del rollback.
15. Ricercare QTY=101. Determina un'eccezione nel programma che produrrà un messaggio di interrogazione. Utilizzare divisione per zero per questa funzione. L'operatore deve immettere C per annullare il programma a meno che l'opzione INQMSGRPH di descrizione lavoro non fornisca automaticamente una risposta. Questo simula una condizione in cui si verifica un errore imprevisto e l'operatore annulla il programma.
16. Ricercare QTY=102. Emettere un pannello con un'operazione di interrogazione. Questo arresta il programma su questa fase e consente l'utilizzo della chiave Richiesta di sistema per ottenere un lavoro differente. Annullare il lavoro aggiornato. Questo simula una condizione in cui si è verificata una fine anomala del lavoro o del sistema durante un limite di sincronizzazione.
17. Scrivere il record transazione su TRNP.
18. Sincronizzare i record per la transazione e ritornare alla fase 2 (Vedere pagina 84).
19. Leggere il primo record sul percorso di accesso per il file TRNL, utilizzando USER come chiave. Poiché questo file si trova nella sequenza LIFO, questo sarà l'ultimo record transazione immesso dall'utente.
20. Ricercare una condizione di record non trovato nel file TRNL che verrebbe creato se il file non contenesse voci per questo utente.
21. Se non esistono record per questo utente, impostare su indicatore 64 per visualizzare un messaggio di errore e ritornare alla fase 2 (Vedere pagina 84).
22. Visualizzare l'ultima transazione immessa per questo utente. Queste informazioni possono essere utilizzate se l'operatore dimentica ciò che è stato immesso precedentemente o quando è stata riavviata la transazione. Quando l'operatore risponde, ritornare alla fase 2 (Vedere pagina 84).
23. Eseguire qualsiasi funzione di fine del programma.

Esempio: utilizzare un file di registrazione transazione per avviare un'applicazione

Questo esempio fornisce il codice di esempio e le istruzioni relative al modo in cui utilizzare un file di registrazione transazione per avviare un'applicazione dopo una fine anomala.

Nota: leggere la clausola di esonero della responsabilità inerente all'esempio di codice per quanto riguarda le informazioni legali rilevanti.

Un **file di registrazione transazione** viene utilizzato per avviare nuovamente un'applicazione dopo un errore di sistema o lavoro quando non viene utilizzato un oggetto di notifica. Un file di registrazione transazione viene utilizzato spesso nelle applicazioni interattive per riepilogare gli effetti di una transazione.

Ad esempio, in un'applicazione di immissione ordine, generalmente un record viene scritto su un file di registrazione transazione per ogni elemento ordinato. Il record contiene l'elemento ordinato, la quantità e il prezzo. In un'applicazione conto debitore, un record viene scritto su un file di registrazione transazione per ogni numero di conto che deve ricevere un addebito. Questo record generalmente contiene informazioni come il numero di conto, la somma addebitata e il fornitore.

In molte applicazioni in cui esiste già un file di registrazione transazione, un utente stazione di lavoro può richiedere le informazioni relative all'ultima transazione immessa. Aggiungendo il controllo sincronizzazione alle applicazioni in cui è già esistente un file di registrazione transazione, è possibile:

- Accertarsi che i file database vengano aggiornati al limite di sincronizzazione.
- Semplificare il riavvio della transazione.

E' necessario poter identificare in modo univoco l'utente stazione di lavoro, se viene utilizzato un file di registrazione transazione per il riavvio delle applicazioni sotto il controllo sincronizzazione. Se sul sistema vengono utilizzati nomi profilo utenti univoci, tale nome profilo può essere ubicato in un campo nel record di registrazione transazione. Questo campo può essere utilizzato come chiave sul file.

Nei seguenti esempi si suppone che per eseguire le transazioni venga utilizzato un file inventario di ordine e che un file registrazione transazione sia già esistente. Il programma effettua le seguenti operazioni:

1. Richiede all'utente stazione di lavoro la quantità e il numero di elementi.
2. Aggiorna la quantità nel PRDMSTP (File di produzione principale).
3. Scrive un record sul file di registrazione transazione (ISSLOGL).

Se la quantità dell'inventario a disposizione non è sufficiente, il programma rigetta la transazione. L'utente stazione di lavoro può richiedere al programma dove è stata interrotta la voce dati, poiché il numero degli elementi, la descrizione, la quantità, il nome utente e la data sono scritti sul file di registrazione transazione.

DDS per file fisico PRDMSTP

```
SEQNBR *... .. 1 ... .. 2 ... .. 3 ... .. 4 ... .. 5 ... .. 6 ... .. 7

1.00      A          R PRDMSTR          TEXT('Master record')
2.00      A          PRODCP           3          COLHDG('Product' 'Number')
3.00      A          DESCRP           20         COLHDG('Description')
4.00      A          ONHAND            5 0        COLHDG('On Hand' 'Amount')
5.00      A                                     EDTCDE(Z)
6.00      A          K PRODCP
```

DDS per file fisico ISSLOGP utilizzato da ISSLOGP

```
SEQNBR *... .. 1 ... .. 2 ... .. 3 ... .. 4 ... .. 5 ... .. 6 ... .. 7

1.00      A          R ISSLOGR          TEXT('Product log record')
2.00      A          PRODCP           3          COLHDG('Product' 'Number')
3.00      A          DESCRP           20         COLHDG('Description')
4.00      A          QTY              3 0        COLHDG('Quantity')
5.00      A                                     EDTCDE(Z)
6.00      A          USER            10         COLHDG('User' 'Name')
7.00      A          DATE             6 0        EDTCDE(Y)
8.00      A          COLHDG('Date')
```

DDS per file logico ISSLOGL

```
SEQNBR *... .. 1 ... .. 2 ... .. 3 ... .. 4 ... .. 5 ... .. 6 ... .. 7

1.00      A                                     LIFO
2.00      A          R ISSLOGR          PFILE(ISSLOGP)
3.00      A          K USER
```

DDS per file di visualizzazione PRDISSD utilizzato nel programma

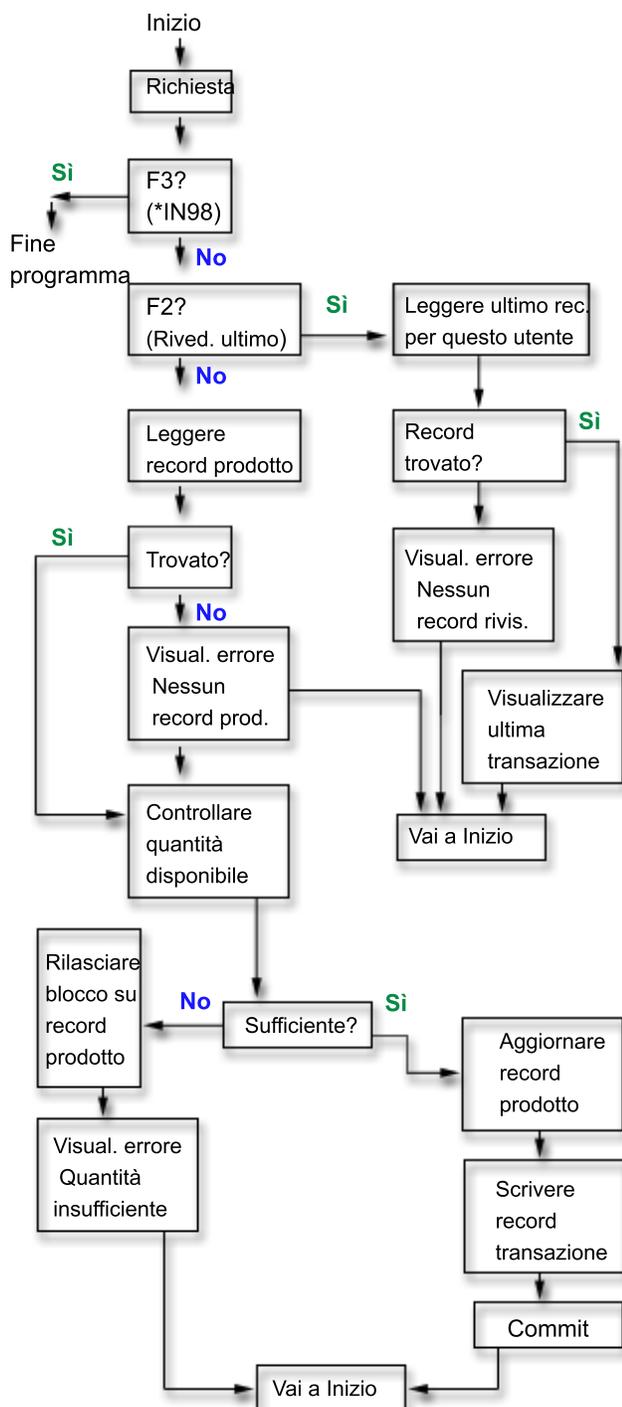
```
SEQNBR *... .. 1 ... .. 2 ... .. 3 ... .. 4 ... .. 5 ... .. 6 ... .. 7 ..

1.00      A                                     REF(ISSLOGP)
2.00      A          R PROMPT
3.00      A                                     CA03(98 'End of program')
4.00      A                                     CA02(97 'Where am I')
5.00      A          1 20'ISSUES PROCESSING'
6.00      A          3 2'Quantity'
7.00      A          QTY              R          I          +1
8.00      A 62                                     ERRMSG('Not enough +
```

| | | | | | |
|-------|---|-----------|---|----|------------------------------|
| 9.00 | A | | | | Qty' 62) |
| 10.00 | A | | | | +6'Product' |
| 11.00 | A | PRODCT | R | I | +1 |
| 12.00 | A | 61 | | | ERRMSG('No Product + |
| 13.00 | A | | | | record found' 62) |
| 14.00 | A | 55 | | 15 | 2'No Previous record exists' |
| 15.00 | A | | | 24 | 2'CF2 Last transaction' |
| 16.00 | A | R RESTART | | | |
| 17.00 | A | | | 1 | 20'LAST TRANSACTION + |
| 18.00 | A | | | | INFORMATION' |
| 19.00 | A | | | 5 | 2'Product' |
| 20.00 | A | PRODCT | R | | +1 |
| 21.00 | A | | | 7 | 2'Description' |
| 22.00 | A | DESCRP | R | | +1 |
| 23.00 | A | | | 9 | 2'Qty' |
| 24.00 | A | QTY | R | | +1REFFLD(QTY) |

Questo processo delinea il **Flusso di programma**.

Flusso di programma



*

Il codice dell'operazione COMMIT RPG viene specificato dopo che il file PRDMSTP viene aggiornato e il record viene scritto sul file di registrazione transazione. Poiché ogni richiesta all'operatore rappresenta un limite per una nuova transazione, la transazione viene considerata una singola transazione di invio.

Il nome utente viene inoltrato al programma quando questo viene richiamato. Il percorso di accesso per il file di registrazione transazione viene definito nella sequenza LIFO (last-in-first-out) in modo tale che il programma possa accedere facilmente all'ultimo record immesso.

L'utente stazione di lavoro può riavviare il programma in seguito ad un errore di sistema o lavoro, utilizzando la stessa funzione che ha individuato il punto in cui la voce dati è stata interrotta. Non è necessario aggiungere nessun codice ulteriore al programma. Se attualmente si sta utilizzando un file di registrazione transazione ma non lo si sta utilizzando per ricercare dove si trova dell'utente, aggiungere il nome utente al file di registrazione transazione (i nomi utente devono essere univoci) e utilizzare questo approccio nel programma.

Quanto segue indica il programma RPG utilizzato. Le istruzioni necessarie al controllo sincronizzazione sono contrassegnate da frecce (==>).

Programma RPG

```

SEQNBR *... .. 1 ... .. 2 ... .. 3 ... .. 4 ... .. 5 ... .. 6 ... .. 7 ..
=>1.00  FPRDMSTP UP E           K           DISK           KCOMIT
=>2.00  FISSLOGL IF E           K           DISK           KCOMIT
3.00   PRDISSD CP E           WORKSTN
4.00
5.00   *ENTRY      PLIST           USER      10
6.00   C*
7.00   C* Initialize fields used in Trans Log Rcd
8.00   C*
9.00   C           MOVE UPDATE   DATE
10.00  C*
11.00  C* Basic processing loop
12.00  C*
13.00  C           LOOP      TAG
14.00  C           EXFMTPROMPT
15.00  C 98           GOTO END           End of pgm
16.00  C 97           DO              Where am I
17.00  C           EXSR WHERE
18.00  C           GOTO LOOP
19.00  C           END
20.00  C           PRODC   CHAINPRDMSTR           61      Not found
21.00  C 61           GOTO LOOP
22.00  C           ONHAND  SUB QTY      TEST   50   62  Less than
23.00  C 62           DO              Not enough
24.00  C           EXCPTRLSMST           Release lock
25.00  C           GOTO LOOP
26.00  C           END
27.00  C*
28.00  C* Update master record and output the Transaction Log Record
29.00  C*
30.00  C           Z-ADDTEST   ONHAND
31.00  C           UPDATPRDMSTR
32.00  C           WRITEISSLOGR
=>33.00 C           COMMIT
34.00  C           GOTO LOOP
35.00  C*
36.00  C* End of program processing
37.00  C*
38.00  C           END      TAG
39.00  C           SETON           LR
40.00  C*
41.00  C* WHERE subroutine for "Where am I" requests
42.00  C*
43.00  C           WHERE     BEGSR
44.00  C           USER     CHAINISSLOGL           55      Not found
45.00  C N55           EXFMTRSTART
46.00  C           ENDSR
47.00  C OPRDMSTR E           RLSMST

```

Programma CL utilizzato per richiamare PRDISS programma RPG

```

SEQNBR *... .. 1 ... .. 2 ... .. 3 ... .. 4 ... .. 5 ... .. 6 ... .. 7 ..
1.00   PGM

```

| | | |
|------|-----------|-------------------------------|
| 2.00 | DCL | &USER *CHAR LEN(10) |
| 3.00 | STRCMTCTL | LCKLVL(*CHG) |
| 4.00 | RTVJOBA | USER(&USER) |
| 5.00 | CALL | PRDISS PARM(&USER) |
| 6.00 | MONMSG | MSGID(RPG9001) EXEC(ROLLBACK) |
| 7.00 | ENDCMTCTL | |
| 8.00 | ENDPGM | |

Per utilizzare il controllo sincronizzazione in questo programma, è necessario specificare il livello blocco di *CHG. Il record viene bloccato dalla modifica fino all'esecuzione di una operazione di sincronizzazione. Notare che se la grandezza dell'inventario è insufficiente, il record viene rilasciato esplicitamente. (Se il record non viene rilasciato esplicitamente nel programma, potrebbe essere rilasciato quando il record successivo verrà letto per l'aggiornamento dal file.)

In questo esempio, non esistono ulteriori vantaggi nell'utilizzo del livello blocco *ALL. Se è stato utilizzato *ALL, un'operazione di commit o rollback potrebbe essere utilizzata per rilasciare il record quando la grandezza esistente è insufficiente.

Il codice precedente è un programma CL che richiama il PRDISS programma RPG. Notare l'utilizzo dei comandi STRCMTCTL/ENDCMTCTL. Il nome utente univoco viene richiamato (comando RTVJOBA) e inoltrato al programma. L'utilizzo del comando MONMSG per causare un rollback viene descritto in Esempio: utilizzare un programma di elaborazione standard per avviare un'applicazione.

Esempio: utilizzare un oggetto di notifica per avviare un'applicazione

Quando un programma viene avviato dopo una fine anomala, può ricercare una voce nell'oggetto di notifica. Se ne esiste una, il programma può riavviare la transazione. Dopo aver riavviato la transazione, l'oggetto di notifica viene eliminato dal programma per impedire il riavvio della stessa transazione un'altra volta.

Di seguito vengono riportati i modi in cui è possibile utilizzare un oggetto di notifica:

- Se l'identificazione di sincronizzazione viene inserita in un file database, interrogare questo file per stabilire se riavviare ogni applicazione o il lavoro della stazione di lavoro.
- Se l'identificazione di sincronizzazione viene inserita in una coda messaggi per una stazione di lavoro particolare, un messaggio può essere inviato agli utenti della stazione di lavoro quando si collegano, per comunicargli l'ultima transazione sincronizzata.
- Se l'identificazione di sincronizzazione viene inserita in un file database che possiede una chiave o un nome utente, il programma può leggere questo file quando viene avviato. Se esiste un record nel file, riavviare il programma. Il programma può inviare un messaggio all'utente della stazione di lavoro identificando l'ultima transazione sincronizzata. Eventuali ripristini vengono eseguiti dal programma. Se un record esisteva nel file database, il programma cancella tale record alla fine del programma.
- Per un'applicazione batch, l'identificazione di sincronizzazione può essere inserita in un'area dati che contiene impostazioni generali e di commutazione e altre informazioni sullo stato necessarie per riavviare l'applicazione. Quando viene avviata l'applicazione, questa accede all'area dati e verifica i valori qui memorizzati. Se l'applicazione termina normalmente, l'area dati viene impostata per la successiva esecuzione.
- Per un'applicazione batch, l'identificazione di sincronizzazione può essere inviata ad una coda messaggi. Un programma che viene eseguito quando viene avviata l'applicazione può richiamare i messaggi dalla coda e riavviare i programmi.

Esistono varie tecniche per riavviare le applicazioni in base alle necessità dell'applicazione. Nello scegliere la tecnica, considerare quanto segue:

- Quando esistono più utenti di un programma nello stesso tempo, non è possibile utilizzare un'unica area dati come oggetto di notifica poiché dopo una fine anomala del sistema, l'identificazione di sincronizzazione per ogni utente li sovrapporrebbe nell'area dati.

- Il progetto di cancellare le informazioni nell'oggetto di notifica dovrebbe gestire la situazione quando un errore richiede immediatamente l'utilizzo delle informazioni:
 - Se le informazioni vengono cancellate immediatamente, non esisteranno nel caso in cui si verificherà un altro errore prima dell'elaborazione della transazione interrotta.
 - Le informazioni nell'oggetto di notifica non dovrebbero essere cancellate prima dell'elaborazione con esito positivo della transazione interrotta. In questo caso, esisteranno più di una voce nell'oggetto di notifica se questo è un file database o una coda messaggi.
 - Il programma deve accedere all'ultimo record se esiste più di una voce.
- Un oggetto di notifica non può essere utilizzato per fornire all'utente della stazione di lavoro l'ultima transazione sincronizzata poiché l'oggetto di notifica viene aggiornato solo se si verifica un errore di sistema o lavoro o se esistono delle modifiche non sincronizzate alla fine normale di un lavoro.
- E' importante che le informazioni vengano visualizzate all'utente della stazione di lavoro. Affinché ciò si verifichi, è necessario che il programma converta i codici conservati nell'oggetto di notifica nelle informazioni che aiuteranno l'utente nel riavvio.
- Le informazioni per il riavvio devono essere visualizzate se l'utente della stazione di lavoro lo richiede. E' necessaria una logica aggiuntiva nel programma per impedire che le informazioni vengano visualizzate di nuovo quando non è più utile.
- Un oggetto di notifica singolo e un programma di elaborazione standard possono fornire una funzione di riavvio se l'oggetto di notifica è un file database. Questo programma di elaborazione standard viene richiamato dai programmi che necessitano della funzione di riavvio per ridurre le modifiche su ogni programma individuale.

Per il codice di esempio per l'utilizzo di un oggetto di notifica, vedere quanto segue:

Nota: leggere la clausola di esonero della responsabilità inerente all'esempio di codice per quanto riguarda le informazioni legali rilevanti.

- Oggetto di notifica univoco per ogni programma
- Oggetto di notifica singolo per tutti i programmi

Esempio: oggetto di notifica univoco per ogni programma

Questo argomento fornisce il codice di esempio e le istruzioni per utilizzare un oggetto di notifica univoco per riavviare ogni programma.

L'utilizzo di un singolo oggetto di notifica univoco per ogni lavoro consente di utilizzare un'identificazione di sincronizzazione descritta esternamente anche se esistono più utenti dello stesso programma. Nei seguenti esempi, un file database viene utilizzato come oggetto di notifica ed è utilizzato solo da questo programma.

Il programma dispone di due file database (PRDMSTP e PRDLOCP) che devono essere aggiornati per le ricevute sull'inventario. Il file di visualizzazione utilizzato dal programma è denominato PRDRCTD. Un file database, PRDRCTP, viene utilizzato come oggetto di notifica. Questo oggetto di notifica è definito sul programma come un file ed è utilizzato come definizione della struttura dati di una funzione di notifica.

Vedere DDS per file fisico PRDMSTP (Vedere pagina 86) per visualizzare la DDS per il file fisico PRDMSTP.

Nota: leggere la clausola di esonero della responsabilità inerente all'esempio di codice per quanto riguarda le informazioni legali rilevanti.

DDS per file fisico PRDLOCP

```

SEQNBR *... .. 1 ... .. 2 ... .. 3 ... .. 4 ... .. 5 ... .. 6 ... .. 7

1.00      A          R PRDLOCR          TEXT('Location record')
2.00      A          PRODC          3          COLHDG('Product' 'Number')

```

| | | | | |
|------|---|----------|-----|-----------------------------|
| 3.00 | A | LOCATN | 6 | COLHDG('Location') |
| 4.00 | A | LOCAMT | 5 0 | COLHDG('Location' 'Amount') |
| 5.00 | A | | | EDTCDE(Z) |
| 6.00 | A | K PRODC | | |
| 7.00 | A | K LOCATN | | |

DDS per file di visualizzazione PRDRCTD

| SEQNBR | *... | ... | 1 | ... | 2 | ... | 3 | ... | 4 | ... | 5 | ... | 6 | ... | 7 | .. |
|--------|------|-----|--------|-----|---|-----|----|-----|----|-----|---|-----|---|-----|---|---------------------------|
| 1.00 | A | | | | | | | | | | | | | | | REF(PRDMSTP) |
| 2.00 | A | R | PROMPT | | | | | | | | | | | | | |
| 3.00 | A | | | | | | | | | | | | | | | CA03(98 'End of program') |
| 4.00 | A | | | | | | | | | | | | | | | SETOFF(71 'RESTART') |
| 5.00 | A | | | | | | | | 1 | 20 | | | | | | 'PRODUCT RECEIPTS' |
| 6.00 | A | | | | | | | | 3 | 2 | | | | | | 'Quantity' |
| 7.00 | A | | QTY | | | 3 | 0I | | | | | | | | | +1 |
| 8.00 | A | | | | | | | | | | | | | | | +6'Product' |
| 9.00 | A | | PRODC | | R | | | | | | | | | | | +1 |
| 10.00 | A | 61 | | | | | | | | | | | | | | ERRMSG('No record + |
| 11.00 | A | | | | | | | | | | | | | | | found in the + |
| 12.00 | A | | | | | | | | | | | | | | | master file' 62) |
| 13.00 | A | | | | | | | | | | | | | | | +6'Location' |
| 14.00 | A | | LOCATN | | R | | | | | | | | | | | +1REFFLD(LOCATN PRDLOCP) |
| 15.00 | A | 62 | | | | | | | | | | | | | | ERRMSG('No record + |
| 16.00 | A | | | | | | | | | | | | | | | found in the + |
| 17.00 | A | | | | | | | | | | | | | | | location file' 62) |
| 18.00 | A | | | | | | | | 9 | 2 | | | | | | 'Last Transaction' |
| 19.00 | A | 71 | | | | | | | | | | | | | | +6'This is restart + |
| 20.00 | A | | | | | | | | | | | | | | | information' |
| 21.00 | A | | | | | | | | | | | | | | | DSPATR(HI BL) |
| 22.00 | A | | | | | | | | 12 | 2 | | | | | | 'Quantity' |
| 23.00 | A | | | | | | | | 12 | 12 | | | | | | 'Product' |
| 24.00 | A | | | | | | | | 12 | 23 | | | | | | 'Location' |
| 25.00 | A | | | | | | | | 12 | 35 | | | | | | 'Description' |
| 26.00 | A | | LSTPRD | | R | | | | 14 | 15 | | | | | | REFFLD(PRODC) |
| 27.00 | A | | LSTLOC | | R | | | | 14 | 26 | | | | | | REFFLD(LOCATN *SRC) |
| 28.00 | A | | LSTQTY | | R | | | | 14 | 5 | | | | | | REFFLD(QTY *SRC) |
| 29.00 | A | | | | | | | | | | | | | | | EDTCDE(Z) |
| 30.00 | A | | LSTDSC | | R | | | | 14 | 35 | | | | | | REFFLD(DESCRP) |

DDS per oggetto di notifica e struttura dati descritti esternamente (PRDRCTP)

| SEQNBR | *... | ... | 1 | ... | 2 | ... | 3 | ... | 4 | ... | 5 | ... | 6 | ... | 7 | .. |
|--------|------|-----|---------|-----|---|-----|-----|-----|---|-----|---|-----|---|-----|---|------------------------|
| 1.00 | A | | | | | | | | | | | | | | | LIFO |
| 2.00 | A | | | | | | | | | | | | | | | REF(PRDMSTP) |
| 3.00 | A | R | PRDRCTR | | | | | | | | | | | | | |
| 4.00 | A | | USER | | | | 10 | | | | | | | | | |
| 5.00 | A | | PRODC | | R | | | | | | | | | | | |
| 6.00 | A | | DESCRP | | R | | | | | | | | | | | |
| 7.00 | A | | QTY | | | | 3 0 | | | | | | | | | |
| 8.00 | A | | LOCATN | | R | | | | | | | | | | | REFFLD(LOCATN PRDLOCP) |
| 9.00 | A | K | USER | | | | | | | | | | | | | |

Il programma elabora l'oggetto di notifica nel seguente modo:

- All'inizio, il programma elabora in modo casuale l'oggetto di notifica e visualizza un record se esiste per la chiave specifica:
 - Se esistono più record, viene utilizzato l'ultimo per questa chiave poiché il file PRDRCTP si trova nella sequenza LIFO.
 - Se non esistono record, non è stata interrotta una transazione e quindi non è necessario il riavvio.
 - Se il programma non funziona precedentemente alla prima operazione di sincronizzazione riuscita, non è necessario il riavvio.
- La routine per eliminare l'oggetto di notifica si verifica alla fine del programma:

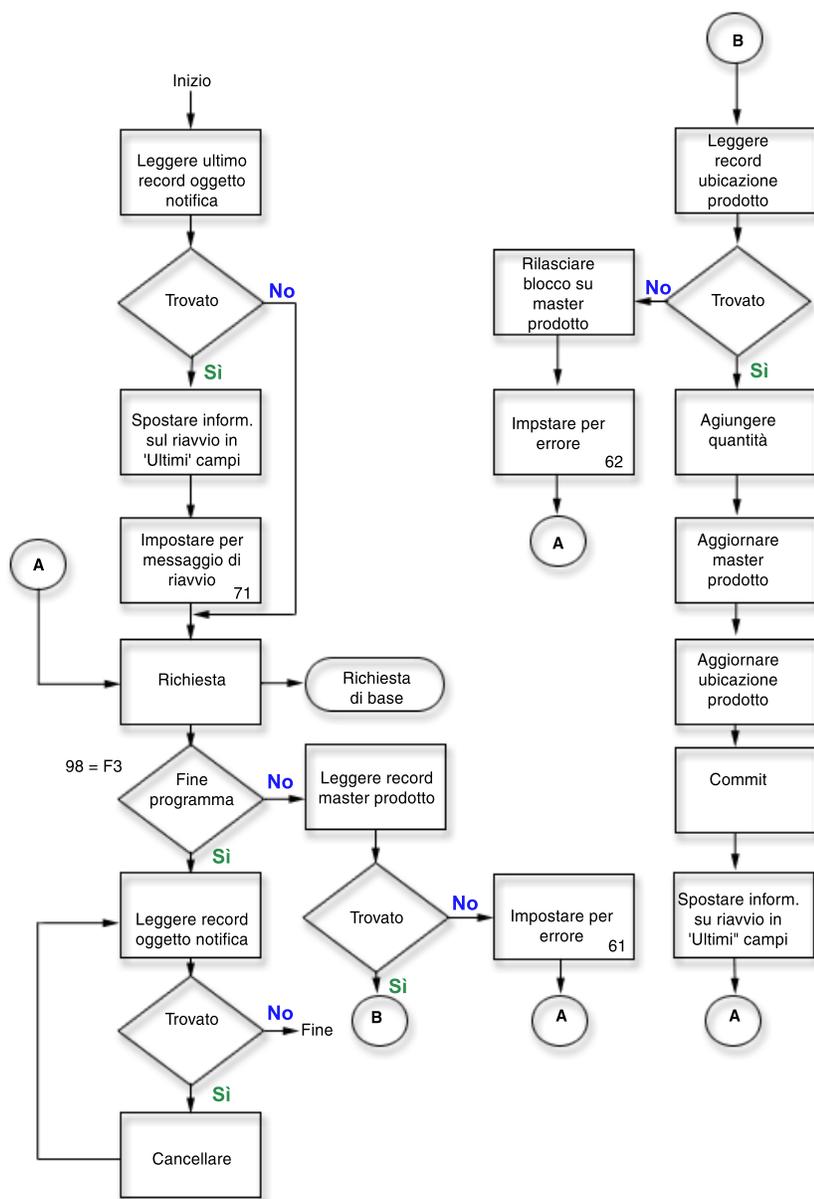
- Se si sono verificati più errori, la routine può gestire la cancellazione di più record nell'oggetto di notifica.
- Sebbene il sistema posizioni l'identificazione di sincronizzazione in un file database, è necessario specificare l'identificazione di sincronizzazione come una variabile nel programma RPG.
- Poiché RPG consente ad una struttura dati di essere descritta esternamente, la struttura dati rappresenta un modo conveniente per specificare l'identificazione di sincronizzazione. In questo esempio, la struttura dati utilizza la stessa descrizione esterna che il file database utilizza come oggetto di notifica.

L'elaborazione di questo programma richiede all'utente un numero del prodotto, un'ubicazione ed una quantità:

- E' necessario aggiornare due file:
 - PRDMSTP (File master del prodotto)
 - PRDLOCP (File di ubicazione del prodotto)
- E' necessario che esista un record in ogni file prima che questi siano aggiornati.
- Il programma sposta i campi di immissione sugli ultimi campi corrispondenti dopo che ogni transazione viene immessa con esito positivo. Questi ultimi campi vengono visualizzati all'operatore su ogni richiesta, come feedback di ciò che è stato immesso per ultimo.
- Se esistono delle informazioni per il riavvio, vengono spostate su questi ultimi campi e viene visualizzato un messaggio speciale sullo schermo.

Questo processo viene delineato nella seguente figura. Il nome utente viene inoltrato al programma per fornire un record univoco nell'oggetto di notifica.

Flusso di programma



*

Il seguente è il codice sorgente RPG per questo esempio. L'oggetto di notifica (file PRDRCTP) viene utilizzato come un file normale all'inizio e alla fine del programma e, inoltre, viene specificato come oggetto di notifica nel CL (comando STRCMTCTL) prima di richiamare il programma.

Sorgente RPG

SEQNBR *... .. 1 2 3 4 5 6 7 ..

```

1.00   FPRDMSTP UF  E           K       DISK       KCOMIT
2.00   FPRDLOCP UF  E           K       DISK       KCOMIT
3.00   FPRDRCTD CF  E                   WORKSTN
4.00   F*
5.00   F* The following file is the specific notify object for this pgm.
6.00   F*       It is accessed only in a restart situation and at the
7.00   F*       end of the program to delete any records. The records
  
```

```

8.00      F*          are written to the notify object by Commitment Control.
9.00      F*
10.00     FPRDRCTP UF E          K          DISK
11.00     ICMTID      E DSPRDRCTP

12.00     C          *ENTRY  PLIST
13.00     C          PARM          USER10 10
14.00     C          MOVE USER10  USER
15.00     C*
16.00     C* Check for restart information - get last rcd per user
17.00     C* PRDRCTP file access path is in LIFO sequence
18.00     C*
19.00     C          USER      CHAINPRDRCTR          20      Not found
20.00     C N20          DO          Restart
21.00     C          EXSR MOVLST          Move to last
22.00     C          SETON          71      Restart
23.00     C          END
24.00     C*
25.00     C* Basic processing loop
26.00     C*
27.00     C          LOOP      TAG
28.00     C          EXFMTPROMPT
29.00     C 98          GOTO END          End of pgm
30.00     C          PRODC T      CHAINPRDMSTR          61      Not found
31.00     C 61          GOTO LOOP
32.00     C          KEY      KLIST
33.00     C          KFLD          PRODC T
34.00     C          KFLD          LOCATN
35.00     C          KEY      CHAINPRDLOC R          62      Not found
36.00     C 62          DO
37.00     C          EXCPTRLMS T          Release lck
38.00     C          GOTO LOOP
39.00     C          END
40.00     C          ADD QTY      ONHAND          Add
41.00     C          ADD QTY      LOCAM T
42.00     C          UPDATPRDMSTR          Update
43.00     C          UPDATPRDLOC R          Update
44.00     C*

45.00     C* Commit and move to previous fields
46.00     C*
47.00     C          CMTID      COMIT
48.00     C          EXSR MOVLST          Move to last
49.00     C          GOTO LOOP
50.00     C*
51.00     C* End of program processing
52.00     C*
53.00     C          END      TAG
54.00     C          SETON          LR
55.00     C*56.00     C* Delete any records in the notify object
57.00     C*
58.00     C          DL TLP      TAG
59.00     C          USER      CHAINPRDRCTR          20      Not found
60.00     C N20          DO
61.00     C          DELETPRDRCTR          Delete
62.00     C          GOTO DL TLP
63.00     C          END
64.00     C*
65.00     C* Move to -Last Used- fields for operator feedback
66.00     C*
67.00     C          MOVLST      BEGSR
68.00     C          MOVE PRODC T      LSTPRD
69.00     C          MOVE LOCATN      LSTLOC
70.00     C          MOVE QTY      LSTQTY
71.00     C          MOVE DESCRP      LSTDSC
72.00     C          ENDSR
73.00     C OPRDMSTR E          RLSMST

```

Esempio: oggetto di notifica singolo per tutti i programmi

L'utilizzo di un oggetto di notifica singolo per tutti i programmi è vantaggioso poiché tutte le informazioni necessarie per il riavvio si trovano nello stesso oggetto ed è possibile utilizzare in tutti i programmi l'approccio standard all'oggetto di notifica. In questa situazione, utilizzare una combinazione univoca delle identificazioni utente e programma per accertarsi che il programma acceda alle corrette informazioni quando viene riavviato.

Poiché le informazioni necessarie per il riavvio possono variare da un programma all'altro, è necessario utilizzare una struttura dati descritta esternamente per l'identificazione di sincronizzazione. Se viene utilizzato un oggetto di notifica singolo, il programma precedente potrebbe descrivere la struttura dati all'interno del programma piuttosto che esternamente. Ad esempio:

| | | |
|----|------|--------|
| 1 | 10 | USER |
| 11 | 20 | PGMNAM |
| 21 | 23 | PRODC |
| 24 | 29 | LOCATN |
| 30 | 49 | DESC |
| 50 | 51 0 | QTY |
| 52 | 220 | DUMMY |

In ogni programma che utilizza questo oggetto di notifica, le informazioni specificate per l'identificazione di sincronizzazione dovrebbero essere univoche per il programma (i nomi utente e programma non sono univoci). L'oggetto di notifica deve essere abbastanza grande per contenere il maggior numero di informazioni che qualsiasi programma può inserire nell'identificazione di sincronizzazione.

In Esempio: oggetto di notifica univoco per ogni programma vengono forniti ulteriori esempi relativi all'utilizzo degli oggetti di notifica.

Esempio: utilizzare un programma di elaborazione standard per avviare un'applicazione

Un programma di elaborazione standard è un modo per riavviare l'applicazione tramite un file database come oggetto di notifica per tutte le applicazioni. Questo approccio presume che i nomi profilo utente siano univoci per ogni utente di tutte le applicazioni che utilizzano il programma standard.

Per questo approccio, il file fisico NFOBJP viene utilizzato come oggetto di notifica e definito come:

| | |
|-----------------------------|--|
| Nome profilo utente univoco | 10 caratteri |
| Identificazione programma | 10 caratteri |
| Informazioni sul riavvio | campo caratteri (Deve essere abbastanza grande per contenere il numero massimo di informazioni sul riavvio dei programmi che richiedono le informazioni per il riavvio. Questo campo è necessario ai programmi delle applicazioni. Nell'esempio, si presuppone che abbia una lunghezza di 200.) |

Il file è creato con SHARE(*YES). I primi due campi nel file rappresentano la chiave sul file. (Inoltre, questo file può essere definito come una struttura dati nei programmi RPG.)

Nota: leggere la clausola di esonero della responsabilità inerente all'esempio di codice per quanto riguarda le informazioni legali rilevanti.

Quanto segue fornisce il codice di esempio per un programma di elaborazione standard:

- Esempio: codice per un programma di elaborazione standard
- Esempio: programma di elaborazione sincronizzazione standard

- Esempio: utilizzare un programma di elaborazione standard per decidere di riavviare un'applicazione

Esempio: codice per un programma di elaborazione standard

Il seguente è un esempio di utilizzo di un programma di elaborazione standard. L'applicazione visualizzata nel seguente esempio di codice viene eseguito nel seguente modo:

1. Il programma dell'applicazione riceve il nome utente in un parametro e lo utilizza con il nome programma come un identificativo univoco nell'oggetto di notifica.
2. Il programma dell'applicazione inoltra un codice richiesta di R al programma di elaborazione sincronizzazione standard, che determina se un record esiste nell'oggetto di notifica.
3. Se il programma di elaborazione sincronizzazione standard restituisce un codice di 1, viene trovato un record e il programma dell'applicazione presenta all'utente le informazioni necessarie per il riavvio.
4. Il programma dell'applicazione procede con la normale elaborazione.
5. Quando una transazione viene completata, i valori vengono salvati come riferimento in modo tale che l'utente stazione di lavoro possa vedere le operazioni effettuate per la transazione precedente.

Le informazioni salvate non vengono fornite dall'oggetto di notifica poiché questo viene aggiornato solo se si verificano degli errori nel lavoro o nel sistema.

Vedere Flusso di elaborazione per il flusso di questo programma.

Nota: leggere la clausola di esonero della responsabilità inerente all'esempio di codice per quanto riguarda le informazioni legali rilevanti.

Esempio programma dell'applicazione

SEQNBR *... .. 1 2 3 4 5 6 7 ..

```

1.00      FPRDMSTP UF E          K          DISK          KCOMIT
2.00      FPRDLOCP UF E          K          DISK          KCOMIT
3.00      FPRDRCTD CF E          WORKSTN
4.00      F*
5.00      F* The following is a compile time array which contains the
6.00      F*   restart information used in the next example
7.00      F*
8.00      E              RTXT   50  50  1              Restart text
9.00      I*
10.00     I* Data structure used for info passed to notify object
11.00     I*
12.00     ICMTID      DS
13.00     I              1  10  USER
14.00     I              11  20  PGMNAM
15.00     I              21  23  PRODC
16.00     I              24  29  LOCATN
17.00     I              30  49  DESCRP
18.00     I              P  50  510QTY
19.00     I              52  170  DUMMY
20.00     I              171 220  RSTART
21.00     C              *ENTRY  PLIST
22.00     C              PARM      USER10 10
23.00     C*
24.00     C* Initialize fields used to communicate with std program
25.00     C*
26.00     C              MOVE USER10  USER
27.00     C              MOVE 'PRDRC2' PGMNAM
28.00     C              MOVE 'R'      RQSCOD          Read Rqs
29.00     C              CALL 'STDCMT'
30.00     C              PARM          RQSCOD  1
31.00     C              PARM          RTNCOD  1
32.00     C              PARM          CMTID 220          Data struct
33.00     C              RTNCOD  IFEQ '1'          Restart
34.00     C              EXSR MOVLST          Move to last
35.00     C SETON              71          Restart
36.00     C              END

```

```

37.00 C*
38.00 C* Initialize fields used in notify object
39.00 C*
40.00 C MOVEARTXT,1 RSTART Move text
41.00 C*
42.00 C* Basic processing loop
43.00 C*
44.00 C LOOP TAG
45.00 C EXFMTPROMPT
46.00 C 98 GOTO END
47.00 C PRODCT CHAINPRDMSTR 61 Not found
48.00 C 61 GOTO LOOP
49.00 C KEY KLIST
50.00 C KFLD PRODCT
51.00 C KFLD LOCATN
SEQNBR *... .. 1 ... .. 2 ... .. 3 ... .. 4 ... .. 5 ... .. 6 ... .. 7 ..

52.00 C KEY CHAINPRDLOCR 62 Not found
53.00 C 62 DO
54.00 C EXCPTRLSTMST Release lck
55.00 C GOTO LOOP
56.00 C END
57.00 C ADD QTY ONHAND Add
58.00 C ADD QTY LOCAMT
59.00 C UPDATPRDMSTR Update
60.00 C UPDATPRDLOCR Update
61.00 C*
62.00 C* Commit and move to previous fields
63.00 C*
64.00 C CMTID COMIT
65.00 C EXSR MOVLST Move to last
66.00 C GOTO LOOP
67.00 C* End of program processing
68.00 C*
69.00 C END TAG
70.00 C MOVE 'D' RQSCOD Dlt Rqs
71.00 C CALL 'STDCMT'
72.00 C PARM RQSCOD
73.00 C PARM RTNCOD
74.00 C PARM CMTID
75.00 C SETON LR
76.00 C*
77.00 C* Move to -Last Used- fields for operator feedback
78.00 C*
79.00 C MOVLST BEGSR
80.00 C MOVE PRODCT LSTPRD
81.00 C MOVE LOCATN LSTLOC
82.00 C MOVE DESCRP LSTDSC
83.00 C MOVE QTY LSTQTY
84.00 C ENDSR
85.00 OPRDMSTR E RLSMST
86.00 ** RTXT Restart Text
87.00 Inventory Menu - Receipts Option

```

Esempio: codice per un programma di elaborazione della sincronizzazione standard

Il programma di elaborazione STDCMT (sincronizzazione standard) esegue le funzioni necessarie per comunicare con un unico oggetto di notifica utilizzato da tutte le applicazioni. Mentre la funzione di controllo sincronizzazione scrive automaticamente una voce sull'oggetto di notifica, un programma standard scritto dall'utente deve elaborare l'oggetto di notifica. Il programma standard semplifica e normalizza l'approccio.

Il programma viene scritto per verificare i parametri inoltrati ed eseguire l'azione adeguata come segue:

O=Apertura

Il programma chiamante richiede che il file oggetto di notifica rimanga aperto al ritorno. Poiché l'oggetto di notifica viene aperto implicitamente dal programma RPG, questo non dovrebbe chiuderlo. L'indicatore 98 è impostato in modo tale che il programma ritorni con LR disattivo per conservare le aree di lavoro del programma e in modo tale che il programma lasci aperto l'oggetto di notifica per poter essere richiamato senza sovraccarico eccessivo.

C=Chiusura

Il programma chiamante che lo ha determinato non necessita più dell'oggetto di notifica e richiede una chiusura. L'indicatore 98 è disattivato per consentire una chiusura completa dell'oggetto di notifica.

R=Lettura

Il programma chiamante richiede che un record con campi chiave corrispondente vengano letti e restituiti. Il programma utilizza i campi chiave inoltrata per tentare di richiamare un record da NFYOBJP. Se esistono record duplicati per la stessa chiave, viene restituito l'ultimo record. Il codice di ritorno viene impostato di conseguenza e, se il record era esistente, viene restituito nella struttura dati CMTID.

W=Scrittura

Il programma chiamante richiede che un record venga scritto sull'oggetto di notifica per consentire il riavvio del programma chiamante alla successiva chiamata. Il programma scrive i contenuti dei dati inoltrati come un record in NFYOBJP.

D=Cancellazione

Il programma chiamante richiede che i record per questo campo chiave corrispondente vengano cancellati. Solitamente, la funzione viene eseguita quando il programma chiamante completa con esito positivo l'eliminazione delle informazioni relative al riavvio. Il programma tenta di cancellare qualsiasi record per i campi chiave inoltrata. Se non esiste nessun record, viene restituito un codice di ritorno differente.

S=Ricerca

Il programma chiamante richiede la ricerca di un record per un utente particolare a prescindere da quale programma lo ha scritto. Questa funzione viene utilizzata nel programma per il collegamento per indicare il necessario riavvio. Il programma utilizza solo il nome utente come chiave per controllare se esistono dei record. Il codice di ritorno viene impostato in modo adeguato e i contenuti dell'ultimo record per questa chiave (se esistente) vengono letti e restituiti.

Nota: leggere la clausola di esonero della responsabilità inerente all'esempio di codice per quanto riguarda le informazioni legali rilevanti.

Quanto segue visualizza il programma di elaborazione sincronizzazione standard, STDCMT.

Programma di elaborazione sincronizzazione standard

SEQNBR *... .. 1 2 3 4 5 6 7 ..

| | | | | | |
|-------|-------------|--------|--------------|---------------|------------|
| 1.00 | FNFYOBJP UF | E | K | DISK | A |
| 2.00 | ICMTID | DS | | | |
| 3.00 | I | | | 1 10 UNQUSR | |
| 4.00 | I | | | 11 20 UNQPGM | |
| 5.00 | I | | | 21 220 BIGFLD | |
| 6.00 | C | *ENTRY | PLIST | | |
| 7.00 | C | | PARM | RQSCOD 1 | |
| 8.00 | C | | PARM | RTNCOD 1 | |
| 9.00 | C | | PARM | CMTID 220 | |
| 10.00 | C | UNQUSR | CABEQ*BLANKS | BADEND | H1 Invalid |
| 11.00 | C | UNQPGM | CABEQ*BLANKS | BADEND | H2 Invalid |
| 12.00 | C* | | | | |

```

13.00 C* 'O' for Open
14.00 C*
15.00 C          RQSCOD  IFEQ 'O'          Open
16.00 C          SETON          98      End LR
17.00 C          GOTO END
18.00 C          END
19.00 C*
20.00 C* 'C' for Close
21.00 C*
22.00 C          RQSCOD  IFEQ 'C'          Close
23.00 C          SETOF          98
24.00 C          GOTO END
25.00 C          END
26.00 C*
27.00 C* 'R' for Read - Get last record for the key
28.00 C*
29.00 C          RQSCOD  IFEQ 'R'          Read
30.00 C          KEY      KLIST
31.00 C          KFLD          UNQUSR
32.00 C          KFLD          UNQPGM
33.00 C          KEY      CHAINNFYOBJR    51      Not found
34.00 C  51          MOVE '0'          RTNCOD
35.00 C  51          GOTO END
36.00 C          MOVE '1'          RTNCOD          Found
37.00 C          LOOP1    TAG
38.00 C          KEY      READENFYOBJR    20      EOF
39.00 C  20          GOTO END
40.00 C          GOTO LOOP1
41.00 C          END
42.00 C*
43.00 C* 'W' FOR Write
44.00 C*
45.00 C          RQSCOD  IFEQ 'W'          Write
46.00 C          WRITENFYOBJR
47.00 C          GOTO END
48.00 C          END
49.00 C*
50.00 C* 'D' for Delete - Delete all records for the key
51.00 C*
52.00 C          RQSCOD  IFEQ 'D'          Delete
53.00 C          KEY      CHAINNFYOBJR    51      Not found
54.00 C  51          MOVE '0'          RTNCOD
55.00 C  51          GOTO END
56.00 C          MOVE '1'          RTNCOD          Found
57.00 C          LOOP2    TAG
58.00 C          DELETNFYOBJR
59.00 C          KEY      READENFYOBJR    20      EOF
60.00 C  N20          GOTO LOOP2
61.00 C          GOTO END
62.00 C          END
63.00 C*
64.00 C* 'S' for Search for the last record for this user
65.00 C*          (Ignore the -Program- portion of the key)
66.00 C*
67.00 C          RQSCOD  IFEQ 'S'          Search
68.00 C          UNQUSR  SETLLNFYOBJR    20      If equal
69.00 C  N20          MOVE '0'          RTNCOD
70.00 C  N20          GOTO END
71.00 C          MOVE '1'          RTNCOD          Found
72.00 C          LOOP3    TAG
73.00 C          UNQUSR  READENFYOBJR    20      EOF
74.00 C  N20          GOTO LOOP3
75.00 C          GOTO END
76.00 C          END
77.00 C*
78.00 C* Invalid request code processing
79.00 C*

```

```

80.00      C              SETON              H2      Bad RQS code
81.00      C              GOTO BADEND
82.00      C*
83.00      C* End of program processing
84.00      C*
85.00      C              END      TAG
86.00      C N98          SETON              LR
87.00      C              RETRN
88.00      C* BADEND tag is used then fall thru to RPG cycle error return
89.00      C              BADEND  TAG

```

Esempio: utilizzare un programma di elaborazione standard per decidere se riavviare l'applicazione

Questo argomento fornisce il codice CL di esempio per l'utilizzo di un programma di elaborazione standard per decidere se riavviare o meno un'applicazione dopo un IPL anomalo.

Il programma iniziale può richiamare il programma di elaborazione della sincronizzazione standard per stabilire se sia necessario il riavvio. L'utente stazione di lavoro può decidere se effettuare o meno il riavvio.

Il programma iniziale invia un codice richiesta di S (ricerca) al programma standard, che ricercherà eventuali record dell'utente. Se esiste un record, le informazioni relative al riavvio vengono inviate al programma iniziale e visualizzate per l'utente stazione di lavoro.

L'identificazione sincronizzazione nell'oggetto di notifica dovrebbe contenere le informazioni che il programma iniziale può visualizzare identificando ciò di cui il programma necessita per essere riavviato. Ad esempio, gli ultimi 50 caratteri dell'identificazione sincronizzazione possono essere riservati per contenere tali informazioni. Nel programma dell'applicazione, queste informazioni possono trovarsi in una schiera tempo di compilazione ed essere spostati in una struttura dati in una fase iniziale. In Esempio: codice per un programma di elaborazione sincronizzazione standard viene descritto come includerle nel programma dell'applicazione.

Il seguente è un esempio di un programma iniziale che stabilisce se un record esiste nell'oggetto di notifica.

Nota: leggere la clausola di esonero della responsabilità inerente all'esempio di codice per quanto riguarda le informazioni legali rilevanti.

Esempio programma iniziale

```

SEQNBR *... .. 1 ... .. 2 ... .. 3 ... .. 4 ... .. 5 ... .. 6 ... .. 7

1.00      PGM
2.00      DCLF          CMTINLD
3.00      DCL          &RQSCOD *CHAR LEN(1) VALUE(S) /* Search */
4.00      DCL          &RTNCOD *CHAR LEN(1)
5.00      DCL          &CMTID *CHAR LEN(220)
6.00      DCL          &USER *CHAR LEN(10)
7.00      DCL          &INFO *CHAR LEN(50)
8.00      RTVJOBA     USER(&USER)
9.00      CHGVAR      &CMTID (&USER *CAT XX)
10.00     /* The XX is required to prevent a blank Pgm nam */
11.00     CALL        STDCMT PARM(&RQSCOD &RTNCOD &CMTID)
12.00     IF          (&RTNCOD *EQ '1') DO /* RESTART REQD */
13.00     CHGVAR      &INFO %SST(&CMTID 171 50)
14.00     SNDRCVF     RCDfmt(RESTART)
15.00     ENDDO
16.00     /*
17.00     /* Enter normal initial program statements */
18.00     /* or -TFRCTL- to first menu program */
19.00     /*
20.00     ENDPGM

```

Risolvere i problemi relativi alle transazioni e al controllo sincronizzazione

Le seguenti pagine forniscono le attività necessarie per risolvere i problemi relativi al controllo sincronizzazione:

Errori controllo sincronizzazione

Queste informazioni descrivono le condizioni che creano gli errori, elencano gli errori del controllo sincronizzazione e forniscono i metodi per esaminare gli errori.

Rilevare stalli

Lo scopo di tale attività è di trovare le condizioni di stallo.

Ripristinare le transazioni dopo un errore nelle comunicazioni

Lo scopo di tale attività è di gestire le transazioni che eseguono il lavoro su un sistema remoto quando la comunicazione con tale sistema non ha esito positivo.

Quando forzare i commit e i rollback e quando annullare la risincronizzazione

Queste informazioni descrivono quando e come forzare un rollback o un commit e quando annullare la risincronizzazione.

Errori controllo sincronizzazione

Quando viene utilizzato il controllo sincronizzazione, è importante individuare quali condizioni causano errori e quali no. In generale, gli errori si verificano quando le funzioni di controllo sincronizzazione vengono utilizzate in modo incoerente, ad esempio viene effettuata l'esecuzione di un comando ENDCMTCTL (Fine controllo sincronizzazione) quando i file che utilizzano la definizione sincronizzazione sono aperti.

Errori durante l'elaborazione della sincronizzazione

Se si verifica un errore di sistema o nelle comunicazioni durante l'operazione di sincronizzazione, è necessario effettuare la risincronizzazione per assicurarsi che i gestori transazioni mantengano i dati coerenti su tutti i sistemi coinvolti nella transazione. Il comportamento della risincronizzazione e il modo in cui influisce sull'operazione sincronizzazione, dipende da questi fattori :

- L'opzione di sincronizzazione *Attesa risultato*. Per ulteriori dettagli, fare riferimento a *Definizione sincronizzazione per sincronizzazione a due fasi: Non attendere risultato*.
- Lo stato della transazione. Per i dettagli, fare riferimento a *Stati della transazione per il controllo sincronizzazione a due fasi*.

Se l'errore è talmente grave che non è possibile ripararlo, oppure non è possibile ripararlo in modo tempestivo, gli operatori di sistema degli altri sistemi coinvolti nella transazione, devono prendere una decisione euristica. La decisione euristica esegue il commit o il rollback delle modifiche apportate al sistema durante la transazione. Se in seguito a tale decisione l'errore viene riparato, e la risincronizzazione rileva che la decisione ha causato dei problemi nell'integrità dei dati, viene inviato il messaggio CPD83D9 o CPD83E9 alla coda messaggi QSYSOPR.

Le seguenti informazioni forniscono ulteriori dettagli relativi alla gestione degli errori di controllo sincronizzazione:

- Condizioni di errore
- Condizioni di non errore
- Messaggi di errore da monitorare durante il controllo sincronizzazione
- Messaggi di errore da monitorare in seguito ad un comando CALL
- Errore nell'elaborazione normale del commit o del rollback

Condizioni di errore

Se si verifica un errore, viene inviato un messaggio di uscita che è possibile monitorare in un programma. I seguenti sono alcuni errori tipici correlati al controllo sincronizzazione:

- I comandi STRCMTCTL consecutivi vengono eseguiti senza un comando ENDCMTCTL.
- I file sono aperti sotto il controllo sincronizzazione ma non è stato eseguito nessun comando STRCMTCTL.

Questo non è una condizione di errore per i programmi in esecuzione in un gruppo attivazione, che devono utilizzare la definizione sincronizzazione a livello del lavoro. Questa può essere avviata solo da un programma unico, ma una volta avviata, la definizione sincronizzazione a livello del lavoro viene utilizzata da un programma in esecuzione in qualsiasi gruppo attivazione che non sta utilizzando una definizione sincronizzazione a livello del gruppo attivazione. I programmi in esecuzione in un gruppo attivazione che devono utilizzare la definizione sincronizzazione a livello del gruppo attivazione, devono avviare tale definizione tramite il comando STRCMTCTL.

- I file aperti per l'emissione sotto il controllo sincronizzazione non vengono registrati su giornale.
- La prima operazione di apertura di un file condiviso posiziona i file sotto il controllo sincronizzazione, mentre ciò non avviene durante le successive operazioni di apertura.
- La prima operazione di apertura di un file condiviso non posiziona il file sotto il controllo sincronizzazione, mentre ciò avviene durante le successive operazioni di apertura.
- Il limite blocco record per il lavoro viene raggiunto in una transazione singola.
- Il programma emette un'operazione di lettura, un'operazione di sincronizzazione e una modifica sullo stesso record. L'operazione di lettura deve essere emessa nuovamente dopo l'operazione di sincronizzazione poiché questa ha liberato il blocco sul record.
- Per un'ubicazione ad una fase, le risorse posizionate sotto il controllo sincronizzazione non risiedono nella stessa ubicazione delle risorse già sotto il controllo sincronizzazione per la relativa definizione.
- Le modifiche non sincronizzate esistono quando viene emesso un comando ENDCMTCTL.

Questa non è una condizione di errore per il comando ENDCMTCTL se tutti i file sono chiusi, qualsiasi database remoto viene scollegato e nessuna risorsa sincronizzazione API viene ancora associata alla definizione sincronizzazione da terminare.

- Viene eseguito un commit, un rollback o un comando ENDCMTCTL e non è stato eseguito il comando STRCMTCTL.

Questa non è una condizione di errore per i programmi in esecuzione in un gruppo attivazione e la definizione sincronizzazione a livello del lavoro è attiva. Questa può essere avviata solo da un programma unico, ma una volta avviata, la definizione sincronizzazione a livello del lavoro viene utilizzata da un programma in esecuzione in qualsiasi gruppo attivazione che non sta utilizzando una definizione sincronizzazione a livello del gruppo attivazione. I programmi in esecuzione in un gruppo attivazione e che devono utilizzare la definizione sincronizzazione, devono prima avviare la definizione sincronizzazione a livello del gruppo attivazione con il comando STRCMTCTL.

- Un comando ENDCMTCTL viene eseguito con i file aperti sotto il controllo sincronizzazione per la relativa definizione.
- Un lavoro che esegue un'operazione di salvataggio possiede una o più definizioni sincronizzazione che non si trovano al limite di sincronizzazione.
- E' stata interrotta un'operazione salva-mentre-attivo, poiché altri lavori con risorse sincronizzabili non hanno raggiunto il limite di sincronizzazione nel tempo specificato per il parametro SAVACTWAIT.
- Un processo salva-mentre-attivo non è stato in grado di proseguire a causa delle risorse sincronizzabili API aggiunte a più di una definizione sincronizzazione per un lavoro singolo.
- Esistono più di 1023 definizioni sincronizzazione per un lavoro singolo.
- La conversazione su un'ubicazione remota è stata persa a causa di un errore di risorsa. Questo potrebbe causare il rollback della transazione.
- Una risorsa ad una fase, aperta per l'aggiornamento, è presente su un nodo che non ha avviato l'operazione di sincronizzazione. E' necessario eliminare o la risorsa o il nodo che ha avviato la richiesta di sincronizzazione.

- Mentre la transazione si trova nello stato RBR (rollback necessario), viene richiesta un'operazione di sincronizzazione. E' necessario effettuare un'operazione di rollback.
- Un programma di uscita API emette una richiesta di commit o una richiesta di rollback.
- Un programma trigger emette una richiesta di commit o di rollback per la definizione sincronizzazione sotto cui è stato richiamato il programma trigger.

Il programma trigger può avviare una definizione sincronizzazione separata ed emettere una richiesta di commit o rollback per tale definizione.

Condizioni di non errore

E' possibile che queste situazioni causino un messaggio di errore anche se il controllo sincronizzazione le consente. Di seguito vengono riportate alcune situazioni per il controllo sincronizzazione in cui non si verificano errori:

- Viene eseguita un'operazione di commit o rollback e nessuna risorsa si trova sotto il controllo sincronizzazione. Ciò consente di includere le operazioni di commit o rollback nel programma, senza considerare se esistono delle risorse sotto il controllo sincronizzazione. Inoltre, consente di specificare un'identificazione di sincronizzazione prima di effettuare delle modifiche sincronizzabili.
- Viene eseguita un'operazione di commit o rollback e non esistono modifiche risorse non sincronizzate. Ciò consente di includere le operazioni di commit e rollback all'interno del programma senza considerare se esistono modifiche risorse non sincronizzate.
- Viene chiuso un file sotto il controllo sincronizzazione ed esistono record non sincronizzati. Questa situazione consente di richiamare un altro programma per eseguire l'operazione di commit o rollback. Ciò si verifica sia se il file viene condiviso sia se non viene condiviso. Questa funzione consente ad un sotto programma di effettuare le modifiche del database che fanno parte di una transazione che coinvolge più programmi.
- Un lavoro termina, sia in modo normale che anomalo, con modifiche non sincronizzate per una o più definizioni sincronizzazione. Le modifiche per tutte le definizioni sincronizzazione sono sottoposte a rollback.
- Un gruppo attivazione termina con modifiche in sospeso per la definizione sincronizzazione a livello del gruppo attivazione. Se il gruppo attivazione termina normalmente e non sono stati riscontrati errori durante la chiusura dei file aperti sotto il controllo sincronizzazione nell'ambito dello stesso gruppo attivazione che sta terminando, viene eseguita una sincronizzazione implicita dal sistema. In caso contrario, viene eseguito un rollback implicito.
- Un programma accede di nuovo ad un record modificato che non è stato sincronizzato. Ciò consente ad un programma di:
 - Aggiungere un record e aggiornarlo prima di specificare l'operazione di sincronizzazione.
 - Aggiornare due volte lo stesso record prima di specificare l'operazione di sincronizzazione.
 - Aggiungere un record e cancellarlo prima di specificare un'operazione di sincronizzazione.
 - Accedere nuovamente ad un record non sincronizzato da un file logico differente (sotto il controllo sincronizzazione).
- E' possibile specificare LCKLVL(*CHG o *CS) sul comando STRCMTCTL e aprire un file con un'operazione di sincronizzazione per la sola lettura. In questo caso non si verifica alcun blocco sulla richiesta. Viene trattato come se il controllo sincronizzazione non fosse in funzione, ma il file non viene visualizzato sull'opzione menu WRKJOB dei file sotto il controllo sincronizzazione.
- Viene emesso il comando STRCMTCTL e non viene aperto alcun file sotto il controllo sincronizzazione. In questa situazione, qualsiasi modifica a livello record apportata ai file non viene effettuata sotto il controllo sincronizzazione.

Messaggi di errore da monitorare durante il controllo sincronizzazione

E' possibile che alcuni messaggi di errore differenti vengano restituiti dalle operazioni di commit o rollback oppure vengano inviati alla registrazione lavori, in base al tipo di messaggio e a quando si è verificato l'errore.

Tali messaggi possono verificarsi nei seguenti casi:

- Elaborazione normale commit o rollback
- Elaborazione commit o rollback durante la fine del processo di lavoro
- Elaborazione commit o rollback durante la fine del gruppo attivazione

E' possibile monitorare i seguenti messaggi durante la fine del gruppo attivazione o la fine del processo di lavoro. Inoltre, è possibile monitorare solo i messaggi CPFxxxx. I messaggi CPDxxxx vengono inviati sempre come messaggi di diagnostica, che non è possibile monitorare. Qualsiasi errore riscontrato al termine di una definizione sincronizzazione a livello del gruppo attivazione durante la fine del gruppo attivazione o qualsiasi definizione sincronizzazione durante la fine del lavoro, vengono lasciati nella registrazione lavori come messaggi di diagnostica.

I messaggi di errore correlati al controllo sincronizzazione da ricercare sono i seguenti:

CPD8351

Le modifiche potrebbero non essere state sincronizzate.

CPD8352

Modifiche non sincronizzate sull'ubicazione remota &3.

CPD8353

Le modifiche sull'RDB (relational database) &1 potrebbero non essere state sincronizzate.

CPD8354

Le modifiche sul file DDM &1 potrebbero non essere state sincronizzate.

CPD8355

Le modifiche sull'oggetto DDL &1 potrebbero non essere state sincronizzate.

CPD8356

Le modifiche di rollback potrebbero essere state sincronizzate.

CPD8358

Le modifiche sull'RDB (relational database) &1 potrebbero non essere state sottoposte a rollback.

CPD8359

Le modifiche al file DDM &1 potrebbero non essere state sottoposte a rollback.

CPD835A

Le modifiche all'oggetto DDL &3 potrebbero non essere state sottoposte a rollback.

CPD835C

Oggetto di notifica &1 in &2 non aggiornato.

CPD835D

La risorsa DRDA non consente di mantenere il cursore SQL.

CPF835F

Operazione di commit o rollback non riuscita.

CPD8360

L'assegnazione dei membri, o dei file, o di entrambi è stata già annullata.

CPD8361

Il programma di uscita API &1 ha avuto esito negativo durante la sincronizzazione.

CPD8362

Il programma di uscita API &1 ha avuto esito negativo durante il roll back.

CPD8363

Il programma di uscita API &1 si è interrotto dopo &4 minuti durante la sincronizzazione.

CPD8364

Il programma di uscita API &1 si è interrotto dopo &4 minuti durante il rollback.

CPD836F

Si è verificato un errore di protocollo durante l'operazione di controllo sincronizzazione.

CPD83D1

La risorsa API &4 non può essere l'ultimo agent.

CPD83D2

Risorsa non compatibile con il controllo sincronizzazione.

CPD83D7

Operazione di commit modificata in operazione di rollback.

CPD83D9

Si è verificata una condizione euristica mista.

CPF83DB

L'operazione di commit è risultata in un rollback.

CPD83DC

Azione in caso di problemi utilizzata per determinare l'operazione di commit o rollback; errore &2.

CPD83DD

Conversazione interrotta; errore &1.

CPD83DE

Informazioni di ritorno non valide.

CPD83EC

Il programma di uscita API &1 ha indicato il rollback.

CPD83EF

Operazione di rollback avviata per successiva unità di lavoro logica.

CPF8350

Definizione sincronizzazione non trovata.

CPF8355

ENDCMTCTL non consentito. Modifiche attive in sospenso.

CPF8356

Controllo sincronizzazione terminato con &1 modifiche locali non sincronizzata.

CPF8358

Oggetto di notifica &1 in &2 non aggiornato.

CPF8359

Operazione di rollback non riuscita.

CPF835A

Fine della definizione sincronizzazione &1 cancellata.

CPF835B

Si sono verificati degli errori durante la fine del controllo sincronizzazione.

CPF835C

Controllo sincronizzazione terminato con modifiche remote non sincronizzate.

CPF8363

Operazione di sincronizzazione non riuscita.

CPF8364

Il valore del parametro controllo sincronizzazione non è valido. Codice errore &3.

CPF8367

Impossibile eseguire operazione di controllo sincronizzazione.

CPF8369

Impossibile ubicare la risorsa di sincronizzazione API sotto il controllo sincronizzazione; codice errore &1.

CPF83D0

Operazione di sincronizzazione non consentita.

CPF83D2

Sincronizzazione completa == Risincronizzazione in corso restituita.

CPF83D3

Sincronizzazione completa == Euristiche miste restituite.

CPF83D4

Voce giornale unità logica di lavoro non inviata.

CPF83E1

Operazione di sincronizzazione non riuscita a causa di una violazione di vincoli.

CPF83E2

Operazione di rollback necessaria.

CPF83E3

Il livello di nidificazione richiesto non è attivo.

CPF83E4

Controllo sincronizzazione terminato con risorse non sincronizzate.

CPF83E6

Operazione di controllo sincronizzazione completata con risincronizzazione in corso.

CPF83E7

Commit o rollback della transazione globale X/Open non consentita.

Monitorare gli errori in seguito ad un comando CALL

Quando viene richiamato un programma che utilizza il controllo sincronizzazione, è necessario monitorare gli errori imprevisti ed eseguire un'operazione di rollback nel caso in cui si verifichi un errore. Ad esempio, è possibile che esistano dei record non sincronizzati quando un programma riscontra un errore imprevisto,

come ad esempio un errore divisione per zero RPG. In base allo stato del parametro INQMSGRPY (risposta messaggio di interrogazione) di un lavoro, il programma invia un messaggio di interrogazione o esegue un'azione predefinita. Se la risposta dell'operatore o l'azione predefinita termina il programma, i record non sincronizzati rimarranno in attesa di un'operazione di commit o rollback.

Se viene richiamato un altro programma e questo causa un'operazione di sincronizzazione, viene sincronizzata la transazione parzialmente completata dal programma precedente.

Per impedire la sincronizzazione delle transazioni parzialmente completate, monitorare i messaggi di uscita in seguito al comando CALL. Ad esempio, se si tratta di un programma RPG, utilizzare la seguente codifica:

```
CALL RPGA
MONMSG MSGID(RPG9001)
EXEC(ROLLBACK) /*Rollback if pgm is canceled*/
```

Se si tratta di un programma COBOL:

```
CALL COBOLA
MONMSG MSGID(CBE9001)
EXEC(ROLLBACK) /*Rollback if pgm is canceled*/
```

Errore nell'elaborazione normale del commit o del rollback

Durante l'elaborazione del commit o del rollback potrebbero verificarsi degli errori in qualsiasi momento. Nella seguente tabella tale elaborazione viene divisa in quattro situazioni. La colonna centrale descrive le azioni eseguite dal sistema quando si riscontrano degli errori durante ogni situazione. La terza colonna suggerisce cosa sarebbe necessario fare o cosa dovrebbe fare l'applicazione in risposta ai messaggi. Tali suggerimenti sono coerenti con il modo in cui l'elaborazione del controllo sincronizzazione viene gestita dal sistema.

| Situazione | Elaborazione del commit o del rollback | Azione suggerita |
|--|--|---|
| Sincronizzazione I/E a livello record non riuscita | <ul style="list-style-type: none"> Se l'errore si verifica durante l'onda di preparazione, viene eseguito il rollback della transazione e viene inviato il messaggio CPF83DB. Se l'errore si verifica durante l'onda sincronizzata, l'elaborazione di sincronizzazione continua a sincronizzare il maggior numero di risorse rimanenti possibile. Il messaggio CPF8363 viene inviato alla fine dell'elaborazione di sincronizzazione. | Monitor dei messaggi; gestire come desiderato |
| Livello oggetto o programma di uscita commit e rollback per risorsa di sincronizzazione API non riuscito durante la sincronizzazione | <ul style="list-style-type: none"> Se l'errore si verifica durante l'onda di preparazione, viene eseguito il rollback della transazione e viene inviato il messaggio CPF83DB. Se l'errore si verifica durante l'onda sincronizzata, l'elaborazione continua a sincronizzare o eseguire il rollback del maggior numero di risorse rimanenti possibile. Viene restituito uno dei seguenti messaggi, in base al tipo di risorsa di sincronizzazione: <ul style="list-style-type: none"> – CPD8353 – CPD8354 – CPD8355 – CPD8361 <p>Il messaggio CPF8363 viene inviato alla fine dell'elaborazione di sincronizzazione.</p> | Monitor dei messaggi; gestire come desiderato |

| Situazione | Elaborazione del commit o del rollback | Azione suggerita |
|--|--|---|
| Rollback I/E a livello record non riuscito | <ol style="list-style-type: none"> 1. Restituito CPD8356 2. Tentativo di continuare l'elaborazione per eseguire il rollback delle risorse di sincronizzazione livello oggetto o API 3. Restituito CPF8359 alla fine dell'elaborazione | Monitor dei messaggi; gestire come desiderato |
| Livello oggetto o programma di uscita commit e rollback per risorse di sincronizzazione API non riuscito durante il rollback | <ol style="list-style-type: none"> 1. Viene restituito uno dei seguenti messaggi in base al tipo di risorsa di sincronizzazione: <ul style="list-style-type: none"> • CPD8358 • CPD8359 • CPD835A • CPD8362 2. L'elaborazione continua 3. Restituito CPF8359 alla fine dell'elaborazione | Monitor dei messaggi; gestire come desiderato |

Elaborazione del commit o del rollback durante la fine del lavoro

Tutte le situazioni descritte nella precedente tabella si verificano anche quando un lavoro sta terminando, ad eccezione di quando viene inviato uno dei seguenti messaggi:

- CPF8356 se vengono registrate solo le risorse locali
- CPF835C se vengono registrate solo le risorse remote
- CPF83E4 se vengono registrate sia le risorse locali sia quelle remote

Inoltre uno dei due messaggi può riferirsi specificatamente al completamento del lavoro se viene richiamato un programma di uscita commit e rollback per una risorsa sincronizzabile API. Se il programma di uscita commit e rollback non viene completato entro 5 minuti, il programma viene annullato, viene inviato un messaggio di diagnostica CPD8363 (per il commit) o CPD8364 (per il rollback) e il resto dell'elaborazione del commit o del rollback continua.

Elaborazione del commit e del rollback durante l'IPL

Tutte le situazioni descritte nella tabella precedente si verificano anche durante il ripristino IPL delle definizioni sincronizzazione, ad eccezione di quando viene inviato il messaggio CPF835F invece del messaggio CPF8359 o CPF8363. I messaggi inviati per una particolare definizione sincronizzazione possono essere visualizzati nella registrazione lavori di uno dei lavori QDBSRVxx o della registrazione QHST. Nella registrazione QHST, il messaggio CPI8356 indica l'inizio del ripristino IPL di una particolare definizione sincronizzazione. Il messaggio CPC8351 indica la fine del ripristino IPL di una particolare definizione sincronizzazione e tra quei due messaggi si trovano altri messaggi relativi al ripristino di tale definizione sincronizzazione.

Uno dei due messaggi può riferirsi specificatamente ad una definizione sincronizzazione se viene richiamato un programma di uscita commit e rollback per una risorsa sincronizzabile API. Se il programma di uscita commit e rollback non viene completato entro 5 minuti, il programma viene annullato, viene inviato un messaggio di diagnostica CPD8363 (per il commit) o CPD8364 (per il rollback) e il resto dell'elaborazione del commit o del rollback continua.

Rilevare stalli

Una condizione di stallo può verificarsi quando un lavoro mantiene un blocco su un oggetto, oggetto A, ed è in attesa di ottenere un blocco su un altro oggetto, oggetto B. Allo stesso tempo, un altro lavoro o transazione mantiene un blocco su un oggetto B ed è in attesa di ottenere un blocco su un oggetto A.

Effettuare le seguenti operazioni per rilevare se si è verificata una situazione di stallo ed eventualmente correggerla:

1. Ubicare il lavoro appeso nella lista dei lavori attivi. Determinare lo stato di un lavoro per rilevare se un lavoro è appeso.
2. Controllare gli oggetti che il lavoro è in attesa di bloccare. Per le transazioni con blocchi nell'ambito della transazione Visualizzare oggetti bloccati per una transazione. Per le transazioni con blocchi nell'ambito del lavoro, vedere Dettagli: proprietà lavoro attivo.
3. Per tutti gli oggetti che il lavoro è in attesa di bloccare, controllare la lista dei detentori di blocco (transazioni o lavori) e tentare di trovare un blocco in conflitto corrispondente al livello richiesto dal lavoro appeso.
4. Se una transazione possiede un blocco in conflitto, visualizzare i lavori associati a questa transazione e vedere se uno di essi è in attesa di blocco.
5. Determinare se questo lavoro in attesa sta tentando di bloccare uno degli oggetti bloccati dal lavoro iniziale appeso. Quando viene trovato il lavoro che sta tentando di bloccare uno degli oggetti bloccati dal lavoro iniziale appeso, è possibile identificare tali oggetti come punti critici.
6. Controllare la transazione per stabilire il corso appropriato dell'azione.
 - a. Controllare le proprietà transazione per rilevare quale applicazione l'ha avviata e verificare il codice applicazione.
 - b. Oppure tracciare fino a questo punto le azioni della transazione, cercando l'ID ciclo sincronizzazione nelle proprietà transazione e cercando in un giornale le voci contenenti questo ID. Per effettuare questo, è possibile utilizzare il comando RTVJRNE (Richiamo voce giornale) e specificare il parametro CMTCYCID.
 - c. Dopo aver ottenuto le informazioni rilevanti l'utente può scegliere di forzare un'operazione di rollback o commit.

Ripristinare le transazioni dopo un errore nelle comunicazioni

In caso di un errore nelle comunicazioni, il sistema solitamente completa automaticamente la risincronizzazione con qualsiasi sistema remoto. Tuttavia, se l'errore è irreversibile e le comunicazioni non verranno più ristabilite sul sistema remoto (se, ad esempio, la linea di comunicazione è interrotta), è necessario annullare la risincronizzazione e ripristinare le transazioni stesse. Le transazioni, inoltre, potrebbero possedere blocchi che devono essere rilasciati.

1. In iSeries Navigator visualizzare le informazioni sul controllo sincronizzazione per la transazione con cui si sta lavorando.
2. Trovare la transazione che si sta tentando di risincronizzare con il sistema remoto. Il campo **Risincronizzazione in corso** per la transazione è impostato su **sì**.
3. Cercare le transazioni che sono state collegate al sistema remoto, controllando lo Stato risorsa per le transazioni individuali.
4. Dopo aver individuato le transazioni, in base allo stato della transazione, potrebbe essere necessario forzare il commit o il rollback.
5. E' possibile decidere se eseguire il commit o il rollback, dopo aver analizzato le proprietà transazioni.
 - E' possibile utilizzare l'**ID unità di lavoro** per trovare altre parti della transazione su altri sistemi.
 - E' possibile, inoltre, stabilire di eseguire il commit o il rollback dallo stato della transazione. Ad esempio, se una transazione database ha eseguito una sincronizzazione a due fasi durante un errore di comunicazione e lo stato in seguito a tale errore è "preparato" o "ultimo agent in sospeso", è possibile scegliere di forzare la sincronizzazione sulla transazione.
6. Dopo aver forzato il commit o il rollback sulle transazioni incerte, arrestare la risincronizzazione sul collegamento non riuscito per le transazioni individuate.

Per ulteriori informazioni sul ripristino vedere Quando forzare i commit e i rollback e quando annullare la risincronizzazione.

Quando forzare i commit e i rollback e quando annullare la risincronizzazione

La decisione di forzare un commit, un rollback o di annullare la risincronizzazione viene definita **decisione euristica**. Una decisione euristica è l'azione che viene effettuata quando si forza il sistema ad eseguire il commit o il rollback di una transazione. Quando si effettua una decisione euristica, lo stato della transazione diventa euristico misto se la decisione non è coerente con i risultati delle altre ubicazioni nella transazione. E' necessario prendersi la responsabilità della determinazione dell'azione effettuata da tutte le altre ubicazioni che hanno partecipato alla transazione e della risincronizzazione dei record del database.

Prima di effettuare una decisione euristica, raccogliere il maggior numero possibile di informazioni relative alla transazione. Visualizzare i lavori associati alla definizione sincronizzazione e creare un record dei giornali e dei file coinvolti. E' possibile utilizzare queste informazioni successivamente se è necessario visualizzare le voci di giornale e applicare o eliminare manualmente le modifiche registrate su giornale.

L'ubicazione migliore dove ricercare le informazioni relative ad una transazione è l'ubicazione che ha avviato la transazione. Tuttavia, la decisione di eseguire il commit o il rollback potrebbe appartenere ad una risorsa API o all'ultimo agent.

Se una risorsa API è stata registrata come risorsa ultimo agent, la decisione finale relativa all'esecuzione del commit o del rollback appartiene alla risorsa API. E' necessario consultare le informazioni relative all'applicazione e al modo in cui essa utilizza la risorsa API per determinare se eseguire il commit o il rollback.

Se la transazione possiede un ultimo agent selezionato, la decisione di eseguire il commit o il rollback appartiene all'ultimo agent. E' necessario controllare lo stato dell'ultimo agent per informazioni relative alla transazione.

Quando è necessario effettuare le decisioni euristiche o annullare la risincronizzazione a causa di un errore di sistema o nelle comunicazioni che non può essere riparato, è possibile trovare tutte le transazioni incerte tramite le seguenti operazioni:

1. In iSeries Navigator, espandere il sistema che si desidera gestire.
2. Espandere i **Database** e il database locale del sistema.
3. Espandere **Transazioni**.
4. Espandere **Transazioni database** o **Transazioni globali**

In questo pannello è possibile visualizzare la definizione sincronizzazione, lo stato di risincronizzazione, l'ID unità di lavoro corrente e lo stato dell'unità di lavoro corrente per ogni transazione. Ricercare le transazioni con:

- Transazioni con **Stato dell'unità di lavoro logica** Preparato o Ultimo agent in sospenso.
- Transazioni che visualizzano lo stato positivo della **Risincronizzazione in corso**.

Per gestire il lavoro che partecipa alla transazione su questo sistema, fare clic con il tastino destro del mouse sulla transazione e selezionare **lavoro**.

Facendo clic con il tastino destro del mouse sulla transazione, è possibile, inoltre, selezionare **Forza sincronizzazione**, **Forza rollback** o **Annulla risincronizzazione**.

Prima di effettuare una decisione euristica o annullare la risincronizzazione, è possibile controllare lo stato dei lavori su altri sistemi associati alla transazione. Il controllo dei lavori sui sistemi remoti può aiutare ad evitare decisioni che causano incoerenze tra i sistemi.

1. Fare clic con il tastino destro del mouse sulla transazione che si desidera gestire.
2. Selezionare **Stato risorsa**

3. Nella finestra di dialogo Stato risorsa, selezionare il separatore **Conversazione** per i collegamenti SNA; selezionare **Collegamento** per i collegamenti TCP/IP.

Ogni risorsa di conversazione rappresenta un sistema remoto che partecipa alla transazione. Sui sistemi remoti, è possibile utilizzare iSeries Navigator per visualizzare le transazioni associate alla transazione.

La parte principale dell'ID unità di lavoro è la stessa su tutti i sistemi. Quando si effettua la visualizzazione delle informazioni sul controllo sincronizzazione sul sistema remoto, ricercare la parte principale dell'ID unità di lavoro uguale sui sistemi locali.

Ad esempio, se l'ID unità di lavoro sul sistema locale inizia con: APPN.RCHASL97.X*112233445566, ricercare l'ID unità di lavoro sul sistema remoto che inizia sempre con APPN.RCHASL97.X*112233445566.

Informazioni correlate al controllo sincronizzazione

Di seguito vengono elencati i manuali iSeries e i Redbook IBM^(TM) (in formato PDF), i siti Web e gli argomenti dell'Information Center correlati all'argomento del Controllo sincronizzazione. E' possibile visualizzare o stampare tutti i PDF.

Manuali

- COBOL/400 User's Guide



(circa 425 pagine)

- RPG/400 User's Guide



(circa 580 pagine)

Redbook

- Connecting WebSphere to DB2 UDB Server



(circa 458 pagine)

- Advanced Functions and Administration for DB2 Universal Database for iSeries



(circa 372 pagine)

- Stored Procedures and Triggers on DB2 Universal Database for iSeries



(circa 424 pagine)

- Striving for Optimal Journal Performance on DB2 Universal Database for iSeries



(circa 184 pagine)

Sito Web

Open Group (www.opengroup.org)



Informazioni ulteriori

- DB2 Universal Database for iSeries Database Programming
- DB2 Universal Database for iSeries SQL Programming Concepts
- XA APIs
- Journal management

Per salvare un PDF sulla propria stazione di lavoro per la visualizzazione o per la stampa:

1. Fare clic con il tastino destro del mouse sul PDF nel proprio browser (fare clic con il tastino destro sul collegamento precedente).
2. Fare clic su **Salva destinazione con nome...**
3. Spostarsi nell'indirizzario in cui si desidera salvare il PDF.
4. Fare clic su **Salva**.

Se è necessario Adobe Acrobat Reader per visualizzare o stampare questi PDF, è possibile scaricare una copia dal Sito Web Adobe (www.adobe.com/products/acrobat/readstep.html)





Printed in Denmark by IBM Danmark A/S