



**SQL Replication Guide and Reference**





**Nota**

Prima di utilizzare queste informazioni e il prodotto a cui si riferiscono, leggere le informazioni riportate in "Informazioni particolari" a pagina 517.

# Indice

## Capitolo 1. Pianificazione per la replica

<b>SQL</b> . . . . .	<b>1</b>
Pianificazione di migrazione . . . . .	1
Pianificazione memoria . . . . .	1
Memoria utilizzata dal programma Capture . . . . .	1
Memoria utilizzata dal programma Apply . . . . .	3
Pianificazione memorizzazione . . . . .	3
Influenza della registrazione per il server di origine DB2. . . . .	3
Influenza della registrazione per i server di destinazione . . . . .	4
Requisiti di memorizzazione delle tabelle di destinazione e delle tabelle di controllo . . . . .	5
Requisiti di spazio per i file di trasferimento relativi al programma Capture . . . . .	6
Requisiti di spazio per i file di trasferimento relativi al programma Apply . . . . .	7
Requisiti di spazio per i file di log diagnostica (z/OS, Linux, UNIX, Windows) . . . . .	8
Pianificazione rilevazione di conflitti . . . . .	8
Pianificazione origini relazionali Non-DB2 . . . . .	9
Velocità di trasmissione della transazione per i trigger Capture . . . . .	9
Influenza della registrazione per i server di origine relazionali non-DB2 . . . . .	9
Coesistenza di trigger esistenti con trigger Capture . . . . .	9
Blocchi per i server di origine Oracle . . . . .	10
Pianificazione della traduzione di tabelle codici . . . . .	10
Replica tra database con tabelle codici compatibili . . . . .	10
Tabelle codici per la replica SQL . . . . .	11
Pianificazione della replica per DB2 per z/OS . . . . .	12
Ottimizzazione delle prestazioni . . . . .	12

## Capitolo 2. Requisiti di autorizzazione per la replica SQL . . . . . 13

Requisiti di autorizzazione per l'amministrazione. . . . .	13
Requisiti di autorizzazione per il programma Capture . . . . .	14
Requisiti di autorizzazione per il programma Apply . . . . .	15
Requisiti di autorizzazione per i trigger Capture sui database relazionali non DB2 . . . . .	17
Managing user IDs and passwords for remote servers (Linux, UNIX, Windows) . . . . .	17

## Capitolo 3. Configurazione dei server per la replica SQL . . . . . 21

Requisiti di connettività per la replica SQL . . . . .	21
Connecting to System i servers from Windows . . . . .	21
Connessione ai server relazionali non-DB2 . . . . .	22
Creazione delle tabelle di controllo per la replica SQL . . . . .	23
Creazione delle tabelle di controllo per la replica SQL . . . . .	23

Creating control tables (System i) . . . . .	24
Creazione delle tabelle di controllo per le origini relazionali non-DB2. . . . .	25
Creazione di più serie di tabelle di controllo Capture . . . . .	25
Creazione delle tabelle di controllo in un database a partizione multipla . . . . .	26
Impostazione dei programmi di replica . . . . .	26
Impostazione dei programmi di replica (Linux, UNIX, Windows) . . . . .	26
Creating SQL packages to use with remote systems (System i) . . . . .	29
Impostazione dei programmi di replica (z/OS) . . . . .	31
Acquisizione di più partizioni del database. . . . .	31
Replica di tabelle partizionate . . . . .	32
Running DB2 Query Patroller in a SQL replication environment . . . . .	32
Setting up journals (System i) . . . . .	33

## Capitolo 4. Registrazione di tabelle e viste come origini di repliche SQL . . . . . 39

Registrazione di tabelle DB2 come origini . . . . .	39
Registrazione di tabelle relazionali non DB2 come origini . . . . .	41
Opzioni di registrazione per tabelle di origine . . . . .	43
Registrazione di una serie secondaria di colonne (serie secondaria verticale) . . . . .	43
Replica di modifica e acquisizione e copia di aggiornamento completo . . . . .	44
Colonne post-immagine e pre-immagine. . . . .	45
Prefisso pre-immagine. . . . .	48
Arresto del programma Capture a seguito di errore . . . . .	48
Opzioni relative alla modalità di memorizzazione degli aggiornamenti del programma Capture . . . . .	49
Impedimento della ricattura delle modifiche (replica di aggiornamenti) . . . . .	50
Opzioni per la rilevazione di conflitti (replica di aggiornamenti) . . . . .	54
Registering tables that use remote journaling (System i) . . . . .	55
Using relative record numbers (RRN) instead of primary keys (System i) . . . . .	56
Comportamento delle viste come origini di repliche . . . . .	57
Viste su una sola tabella . . . . .	57
Viste sull'unione di due o più tabelle. . . . .	57
Registrazione di viste di tabelle come origini . . . . .	59
Gestione di tabelle CCD come origini (IMS) . . . . .	60

## Capitolo 5. Sottoscrizione alle origini per la replica SQL . . . . . 63

Pianificazione del modo in cui raggruppare le origini e le destinazioni . . . . .	63
Pianificazione del numero di membri della serie di richieste . . . . .	64

Pianificazione del numero di serie di richieste per qualificatore Apply . . . . .	64
Creazione di serie di richieste . . . . .	65
Opzioni di elaborazione per le serie di richieste . . . . .	67
Come specificare se la serie di richieste è attiva	68
Specifica del numero di minuti di dati richiamati dal programma Apply . . . . .	68
Opzioni di caricamento per le tabelle di destinazione con integrità referenziale . . . . .	70
Specifica della modalità in cui il programma Apply replica le modifiche per i membri della serie di sottoscrizioni . . . . .	70
Definizione delle procedure memorizzate e delle istruzioni SQL per la serie di richieste . . . . .	71
Opzioni per pianificare la replica delle serie di richieste . . . . .	72
Pianificazione della serie di richieste . . . . .	74
Creazione dei membri delle serie di richieste . . . . .	74
Tipi di tabelle di destinazione . . . . .	77
Proprietà delle colonne per tutti i tipi di tabella di destinazione . . . . .	89

**Capitolo 6. Replica di tipi di dati speciali nella replica SQL. . . . . 95**

Restrizioni generali relative ai dati per la replica . . . . .	95
Tipi di dati LOB (large object) . . . . .	96
Replica di nuovi tipi di dati DB2 Versione 9.7 (Linux, UNIX, Windows) . . . . .	97
Replica delle tabelle con colonne identità . . . . .	99

**Capitolo 7. Creazione di serie secondarie di dati in un ambiente di replica SQL . . . . . 101**

Impostazione secondaria di dati durante la registrazione. . . . .	101
Impostazione secondaria dei dati di origine utilizzando le viste . . . . .	102
Definizione dei trigger sulle tabelle CD per impedire l'acquisizione di righe specifiche. . . . .	102
Creazione di serie secondarie di dati durante la sottoscrizione . . . . .	103

**Capitolo 8. Manipolazione dei dati in un ambiente di replica SQL . . . . . 105**

Ottimizzazione dei dati mediante le procedure memorizzate o le istruzioni SQL . . . . .	106
Associazione delle colonne di origine e di destinazione che presentano nomi diversi . . . . .	107
Creazione di colonne calcolate. . . . .	107

**Capitolo 9. Funzionamento del programma Capture per la replica SQL . . . . . 109**

Avvio del programma Capture (Linux, UNIX, Windows e z/OS) . . . . .	109
Avvio del programma Capture da un punto conosciuto nella registrazione DB2 . . . . .	111
Starting the Capture program (System i) . . . . .	112

Parametri di funzionamento predefiniti per il programma Capture . . . . .	112
Descrizioni dei parametri di funzionamento di Capture . . . . .	114
Metodi di modifica dei parametri di Capture. . . . .	123
Modifica del funzionamento di un programma Capture in esecuzione . . . . .	125
Modifica dei parametri di funzionamento salvati nella tabella IBMSNAP_CAPPARMS. . . . .	127
Arresto del programma Capture . . . . .	127
Reinizializzazione di Capture . . . . .	128
Sospensione del programma Capture (Linux, UNIX, Windows, z/OS) . . . . .	129
Ripresa di Capture (Linux, UNIX, Windows, z/OS) . . . . .	130

**Capitolo 10. Funzionamento del programma Apply per la replica SQL . 131**

Avvio del programma Apply (Linux, UNIX, Windows, z/OS) . . . . .	131
Starting an Apply program (System i) . . . . .	133
Parametri di funzionamento predefiniti per il programma Apply. . . . .	134
Descrizioni dei parametri di funzionamento di Apply . . . . .	136
Metodi per modificare i parametri di funzionamento di Apply . . . . .	144
Modifica di parametri Apply salvati nella tabella IBMSNAP_APPPARMS (z/OS, Linux, UNIX, Windows) . . . . .	145
Arresto del programma Apply. . . . .	145
Modifica della routine di chiusura ASNDONE (z/OS, Linux, UNIX, Windows) . . . . .	146
Modifying the ASNDONE exit routine (System i) . . . . .	147
Aggiornamento delle tabelle di destinazione utilizzando la routine di chiusura ASNLOAD . . . . .	148
Aggiornamento di tabelle di destinazione con la routine di chiusura ASNLOAD (Linux, UNIX, Windows) . . . . .	149
Aggiornamento delle tabelle di destinazione tramite la routine di chiusura ASNLOAD (z/OS). . . . .	151
Personalizzazione del funzionamento della routine di chiusura ASNLOAD (z/OS, Linux, UNIX, Windows) . . . . .	152
Refreshing target tables with the ASNLOAD exit routine (System i) . . . . .	154

**Capitolo 11. Funzionamento dei programmi di replica (z/OS) . . . . . 157**

Utilizzo di attività di sistema per il funzionamento dei programmi di replica . . . . .	157
Utilizzo di JCL per il funzionamento dei programmi di replica . . . . .	157
Avvio del programma Apply su z/OS con JCL . . . . .	158
Lavorare con i programmi di replica SQL in esecuzione mediante il comando MVS MODIFY. . . . .	159
Avvio del programma Capture con JCL . . . . .	161
Utilizzare ARM (Automatic Restart Manager) per il riavvio automatico della replica e della pubblicazione (z/OS). . . . .	162

Migrazione dell'ambiente di replica in modalità di  
condivisione dei dati (z/OS) . . . . . 163

## Capitolo 12. Modifica di un ambiente di replica SQL . . . . . 165

Registrazione di nuovi oggetti . . . . . 165  
Modifica degli attributi di registrazione per gli  
oggetti registrati . . . . . 166  
Aggiunta di colonne a tabelle di origine . . . . . 166  
Arresto della cattura delle modifiche per gli oggetti  
registrati . . . . . 168  
Esecuzione di registrazioni idonee per la  
riattivazione . . . . . 169  
Rimozione delle registrazioni . . . . . 170  
Modifica degli schemi Capture . . . . . 171  
Creazione di nuove serie di richieste . . . . . 173  
Aggiunta di nuovi membri di serie di richieste alle  
serie di richieste esistenti . . . . . 173  
Disabilitazione dei membri della serie di richieste  
dalle serie di richieste esistenti . . . . . 174  
Abilitazione dei membri della serie di richieste  
nelle serie di richieste esistenti . . . . . 175  
Modifica delle proprietà delle serie di richieste . . . . . 175  
Modifica dei nomi delle serie di richieste . . . . . 176  
Suddivisione di una serie di richieste . . . . . 178  
Unione delle serie di richieste . . . . . 181  
Modifica dei qualificatori Apply delle serie di  
richieste . . . . . 183  
Disattivazione delle serie di richieste . . . . . 185  
Rimozione delle serie di richieste . . . . . 187  
Coordinamento degli eventi della replica con gli  
eventi delle applicazioni del database . . . . . 187  
  Impostazione di un evento END\_SYNCHPOINT  
  utilizzando il segnale tipo USER . . . . . 188  
  Uso del segnale CMD STOP di Capture . . . . . 189  
  Esecuzione di un segnale handshake  
  CAPSTART esternamente al programma Apply . . . . . 192  
  Esecuzione di un segnale CAPSTOP . . . . . 193  
Adjusting for Daylight Savings Time (System i) . . . . . 194  
Opzioni per la promozione della configurazione  
della replica su un altro sistema . . . . . 195

## Capitolo 13. Gestione di un ambiente di replica SQL . . . . . 197

Gestione di sistemi origine . . . . . 197  
  Accesso alle viste e alle tabelle di origine . . . . . 197  
  Registrazioni origine e ricevitori di giornale . . . . . 197  
Gestione delle tabelle di controllo . . . . . 201  
  Programma di utilità RUNSTATS per la replica  
  SQL (Linux, UNIX, Windows, z/OS) . . . . . 201  
  Rebind di pacchetti e piani (z/OS, Linux, UNIX,  
  Windows) . . . . . 202  
  Riorganizzazione delle tabelle di controllo . . . . . 202  
  Eliminazione di tabelle di controllo dinamiche  
  gestite dai programmi Capture (Linux, UNIX,  
  Windows, z/OS) . . . . . 203  
  Eliminazione tabella UOW e CD . . . . . 204  
  Suggerimenti per l'eliminazione di altre tabelle  
  di controllo dinamiche . . . . . 205

  Impedimento malfunzionamenti di replica e  
  recupero da errori . . . . . 206  
Manutenzione delle tabelle di destinazione . . . . . 208

## Capitolo 14. Table differencing and repair . . . . . 209

Table difference utility (asntdiff) . . . . . 209  
Table repair utility (asntrep) . . . . . 214

## Capitolo 15. Replication Alert Monitor 215

Monitoring replication with the Replication Alert  
Monitor . . . . . 215  
Alert conditions and notifications for the  
Replication Alert Monitor . . . . . 217  
  Alert conditions for the Replication Alert  
  Monitor . . . . . 217  
  E-mail notifications for replication alert  
  conditions . . . . . 220  
  Specifying where to send monitor alerts . . . . . 222  
  The ASNMAIL exit routine for sending alerts in  
  replication (Linux, UNIX, Windows). . . . . 222  
Setting up the Replication Alert Monitor . . . . . 223  
  Memory used by the Replication Alert Monitor . . . . . 223  
  Authorization requirements for the Replication  
  Alert Monitor . . . . . 224  
  Optional: Binding the Replication Alert Monitor  
  program packages (Linux, UNIX, Windows) . . . . . 224  
  Creating control tables for the Replication Alert  
  Monitor . . . . . 225  
  Defining contact information for the Replication  
  Alert Monitor . . . . . 226  
  Creating monitors for replication or publishing . . . . . 227  
  Selecting alert conditions for the Replication  
  Alert Monitor . . . . . 227  
  Changing alert conditions for the Replication  
  Alert Monitor . . . . . 228  
  Defining suspension periods for the Alert  
  Monitor . . . . . 229  
Operating the Replication Alert Monitor . . . . . 230  
  Starting monitors . . . . . 230  
  Reinitializing monitors . . . . . 231  
  Suspending and resuming a monitor . . . . . 231  
  Ending a monitor suspension . . . . . 232  
  Stopping monitors . . . . . 232  
  Reviewing Monitor program messages . . . . . 233  
Parameters of the Replication Alert Monitor . . . . . 233  
  Default values of Replication Alert Monitor  
  parameters . . . . . 233  
  Descriptions of the Replication Alert Monitor  
  parameters . . . . . 233  
  Changing runtime parameters for the  
  Replication Alert Monitor . . . . . 236  
  Specifying how often the Replication Alert  
  Monitor runs . . . . . 237  
  Specifying notification criteria for selected alert  
  conditions . . . . . 237  
  Specifying notification criteria for operational  
  errors . . . . . 237  
  Specifying prune intervals for data from the  
  Replication Alert Monitor . . . . . 238

## Capitolo 16. Replication services (Windows) . . . . . 239

Description of Windows services for replication	239
Creating a replication service	240
Starting a replication service	241
Stopping a replication service	241
Viewing a list of replication services	241
Dropping a replication service	241

## Capitolo 17. Pianificazione di programmi di replica SQL su vari sistemi operativi . . . . . 243

Pianificazione di programmi sui sistemi operativi Linux e UNIX	243
Pianificazione di programmi sui sistemi operativi Windows	243
Pianificazione dei programmi sui sistemi operativi z/OS	244
Scheduling programs on the System i operating system	244

## Capitolo 18. Modalità di comunicazione dei componenti della replica SQL . . . . . 245

Centro di replica, ASNCLP, trigger o programma Capture e programma Apply	245
Il programma Capture e il programma Apply	246
Trigger Capture e programma Apply	247
Strumenti di gestione e Replication Alert Monitor	249
Replication Alert Monitor, il programma Capture e il programma Apply	249

## Capitolo 19. Visualizzazione dei prospetti relativi ai programmi di replica SQL . . . . . 251

Verifica dello stato dei programmi di replica (z/OS, Linux, UNIX, Windows)	251
Analisi dei dati cronologici per le tendenze	252
Analisi dei messaggi del programma Capture	254
Esame della velocità di trasmissione del programma Capture	254
Visualizzazione latenza dei dati elaborati dal programma Capture	254
Analisi dei messaggi del programma Apply	255
Esame della velocità di trasmissione del programma Apply	256
Visualizzazione della durata media della replica di transazioni	256
Checking the status of the Capture and Apply journal jobs (System i)	257
Monitoring the progress of the Capture program (System i)	257

## Capitolo 20. Personalizzazione ed esecuzione di script SQL per la replica . . . . . 259

## Capitolo 21. Regole di denominazione per gli oggetti della replica SQL . . . . . 261

## Capitolo 22. Comandi di sistema per la replica SQL (Linux, UNIX, Windows, z/OS) . . . . . 263

asncap: Avvio Capture	263
asnccmd: Funzionamento Capture	271
asnapply: avvio di Apply	275
asnacmd: utilizzo di Apply	281
asnanalyze: Funzionamento di Analyzer	282
asnmon: Starting a Replication Alert Monitor	285
asnmcmd: Working with a running Replication Alert Monitor	290
asnpwd: Creating and maintaining password files	293
anscrt: Creating a replication service	297
ansndrop: Dropping a replication service	300
ansnlist: Listing replication services	301
asntdiff: Comparing data in source and target tables	302
asntdiff -f (input file) command option	307
asntrep: Operating the replication trace facility	310
asntrep: Repairing differences between source and target tables	317

## Capitolo 23. System commands for SQL replication (System i) . . . . . 321

ADDDPRREG: Adding a DPR registration (System i)	321
ADDDPRSUB: Adding a DPR subscription set (System i)	329
ADDDPRSUBM: Adding a DPR subscription-set member (System i)	344
ANZDPR: Operating the Analyzer (System i)	354
CHGDPRCAPA: Changing DPR Capture attributes (System i)	357
CRTDPRBL: Creating the replication control tables (System i)	362
ENDDPRAPY: Stopping Apply (System i)	363
ENDDPRCAP: Stopping Capture (System i)	366
GRTDPRAUT: Authorizing users (System i)	368
INZDPRCAP: Reinitializing DPR Capture (System i)	376
OVRDPRCAPA: Overriding DPR Capture attributes (System i)	377
RMVDPRREG: Removing a DPR registration (System i)	382
RMVDPRSUB: Removing a DPR subscription set (System i)	383
RMVDPRSUBM: Removing a DPR subscription-set member (System i)	385
RVKDPRAUT: Revoking authority (System i)	386
STRDPRAPY: Starting Apply (System i)	388
STRDPRCAP: Starting Capture (System i)	394
WRKDPTRC: Using the DPR trace facility (System i)	402

## Capitolo 24. Strutture della tabella di replica SQL . . . . . 407



Tables at a glance . . . . .	407
Tabella nel server di controllo Capture . . . . .	414
IBMSNAP_AUTHTKN table (System i). . . . .	416
Tabella IBMSNAP_CAPENQ (z/OS, Linux, UNIX, Windows) . . . . .	417
Tabella IBMSNAP_CAPMON . . . . .	418
Tabella IBMSNAP_CAPPARMS . . . . .	420
Tabella IBMSNAP_CAPSCHEMAS . . . . .	423
Tabella IBMQREP_COLVERSION (z/OS) . . . . .	424
Tabella IBMSNAP_CAPTRACE . . . . .	425
Tabella CCD (non DB2) . . . . .	426
Tabella CD . . . . .	427
Tabella IBMQREP_IGNTRAN . . . . .	428
Tabella IBMQREP_IGNTRANTRC . . . . .	429
Tabella IBMSNAP_PARTITIONINFO . . . . .	430
tabella IBMSNAP_PRUNCNTL . . . . .	431
Tabella IBMSNAP_PRUNE_LOCK . . . . .	433
Tabella IBMSNAP_PRUNE_SET . . . . .	434
IBMSNAP_REG_EXT (System i) . . . . .	434
Tabella IBMSNAP_REGISTER . . . . .	436
Tabella IBMSNAP_REG_SYNCH (relazionale non DB2). . . . .	443
Tabella IBMSNAP_RESTART . . . . .	443
Tabella IBMSNAP_SEQTABLE (Informix) . . . . .	446
Tabella IBMSNAP_SIGNAL . . . . .	446
Tabella IBMQREP_TABVERSION (z/OS) . . . . .	449
Tabella IBMSNAP_UOW . . . . .	450
Tabella nel server di controllo Apply . . . . .	452
Tabella ASN.IBMSNAP_APPENQ . . . . .	453
ASN.IBMSNAP_APPLY_JOB (System i). . . . .	454
Tabella ASN.IBMSNAP_APPPARMS. . . . .	455
Tabella ASN.IBMSNAP_APPLYTRACE . . . . .	458
Tabella ASN.IBMSNAP_APPLYTRAIL . . . . .	459
Tabella ASN.IBMSNAP_SUBS_COLS . . . . .	465
Tabella ASN.IBMSNAP_SUBS_EVENT . . . . .	467
tabella ASN.IBMSNAP_SUBS_MEMBR . . . . .	468
tabella ASN.IBMSNAP_SUBS_SET . . . . .	472
Tabella ASN.IBMSNAP_SUBS_STMTS . . . . .	479
Control tables at the Monitor control server . . . . .	480
IBMSNAP_ALERTS table . . . . .	481
IBMSNAP_CONDITIONS table . . . . .	483
IBMSNAP_CONTACTGRP table . . . . .	488
IBMSNAP_CONTACTS table . . . . .	489
IBMSNAP_GROUPS table . . . . .	490
IBMSNAP_MONENQ table. . . . .	490

IBMSNAP_MONPARMS table. . . . .	490
IBMSNAP_MONSERVERS table . . . . .	492
IBMSNAP_MONTRACE table . . . . .	494
IBMSNAP_MONTRAIL table . . . . .	494
IBMSNAP_SUSPENDS table . . . . .	496
IBMSNAP_TEMPLATES table . . . . .	497
Tabella sul server di destinazione. . . . .	497
Tabella aggregati di base . . . . .	498
Tabella aggregati di modifica . . . . .	498
Destinazioni CCD . . . . .	499
Tabella oraria . . . . .	501
Tabella di replica . . . . .	502
Tabella copia utente . . . . .	502

<b>Appendice A. Gli schemi di codifica UNICODE e ASCII per la replica SQL (z/OS).</b> . . . . .	<b>505</b>
Regole per la scelta di uno schema di codifica . . . . .	505
Impostazione degli schemi di codifica . . . . .	506

<b>Appendice B. Avvio dei programmi di replica SQL dall'interno di un'applicazione (Linux, UNIX, Windows)</b> . . . . .	<b>507</b>
---	------------

<b>Appendice C. How the Capture program processes journal entry types for SQL replication (System i)</b> . . . . .	<b>509</b>
--	------------

<b>Documentazione del prodotto</b> . . . . .	<b>511</b>
Come contattare IBM . . . . .	511

<b>Come leggere i diagrammi di sintassi</b> . . . . .	<b>513</b>
---	------------

<b>Accessibilità dei prodotti</b> . . . . .	<b>515</b>
---	------------

<b>Informazioni particolari</b> . . . . .	<b>517</b>
Marchi . . . . .	519

<b>Indice analitico.</b> . . . . .	<b>521</b>
------------------------------------	------------



---

## Capitolo 1. Pianificazione per la replica SQL

Quando si pianifica la replica SQL, potrebbe essere necessario pianificare la migrazione, la memoria, la memorizzazione, i conflitti, i sistemi di origine, la traduzione di tabelle codici e le prestazioni.

---

### Pianificazione di migrazione

La pianificazione di migrazione consiste in una pianificazione dei problemi che potrebbero verificarsi effettuando la migrazione da una versione di replica ad un'altra.

Se si sta effettuando la migrazione da un ambiente di replica esistente, è necessario considerare alcuni problemi di migrazione. *WebSphere Information Integration Migrating to Replication Version 9* descrive come effettuerà la migrazione alla Versione 9 di replica. Per effettuare la migrazione alla Versione 9, per prima cosa è necessario che i server siano aggiornati alla Versione 8. *WebSphere Information Integration Migrating to SQL Replication Version 8* descrive come effettuare la migrazione alla Versione 8 di replica. Inoltre descrive come effettuare la migrazione di ambienti di replica che stanno utilizzando DB2 DataJoiner per replicare i dati replicate data da o a server relazionali non-DB2 relational servers. Questi documenti sono disponibili in linea al sito di supporto del prodotto WebSphere Information Integration.

---

### Pianificazione memoria

La pianificazione della memoria consiste nella pianificazione per la quantità di memoria richiesta dalla replica. La replica utilizza la memoria solo se è richiesta. La quantità di memoria richiesta è direttamente proporzionale alla quantità di dati replicata dall'origine e dalla simultaneità delle transazioni. Fondamentalmente, più dati sono replicati più si verificheranno transazioni simultanee e maggiore sarà la memoria richiesta dalla replica.

L'esecuzione dei programmi Capture e Apply potrebbe richiedere una quantità di memoria significativa.

### Memoria utilizzata dal programma Capture

Il programma Capture utilizza la memoria quando legge la registrazione di recupero di DB2. Esso memorizza i record della transazione singola in memoria finché legge il record di commit o interruzione associato. I dati associati ad una transazione interrotta vengono cancellati dalla memoria e i dati associati ad un record di commit vengono scritti nella tabella CD e nella tabella UOW. Le transazioni per le quali è stato eseguito il commit restano in memoria finché il programma Capture esegue il commit del lavoro, ovvero quando raggiunge l'intervallo di commit.

Per controllare quanta memoria sta utilizzando il programma Capture, consultare la colonna CURRENT\_MEMORY della tabella IBMSNAP\_CAPMON.

È possibile impostare il parametro **memory\_limit** quando si avvia il programma Capture per assicurarsi che Capture utilizzi una quantità specifica di memoria per la memorizzazione associata con la transazione. Altri utilizzi della memorizzazione

non vengono limitati da questo parametro. È anche possibile modificare il parametro **memory\_limit** mentre il programma Capture è in esecuzione. Se Capture raggiunge il limite di memoria, scrive alcune transazioni su un file di trasferimento. È necessario considerare le risorse di memoria utilizzate dal programma Capture in relazione ai propri requisiti di spazio di memorizzazione.

Quando si pianificano i requisiti di memoria del programma Capture, è necessario anche considerare la dimensione delle transazione dell'utente e gli intervalli di commit. Le esecuzioni di lunga durata senza richiedono molta memoria quando il programma Capture è in esecuzione. Generalmente, maggiore sarà l'intervallo di commit, minore sarà la memoria richiesta dal programma Capture.

L'informazione riguardo le registrazioni attive è letta e memorizzata nella memoria quando viene avviata un'istanza del programma Capture e quando vengono aggiunte registrazioni in modo dinamico mentre il programma Capture è in esecuzione.

#### z/OS

#### Linux UNIX Windows

Quando la replica legge i record di registrazione utilizza un buffer di memoria. La dimensione predefinita sui sistemi operativi z/OS è di 66 pagine da 1 KB, ed è di tipo ECSA (extended common service area). La replica utilizza ECSA solo in questa situazione. La dimensione predefinita del buffer sui sistemi operativi Linux®, UNIX® e Windows® è di 50 pagine da 4 KB.

#### System i

CURRENT\_MEMORY è il resoconto aggiornato di memoria supplementare assegnata per mantenere i record di transazione al di sopra della memoria standard utilizzata dai buffer I/O per le tabelle CD attive. Esso rappresenta un'indicazione di quanta memoria supplementare è stata utilizzata per mantenere un grande numero di transazioni, non è una somma accurata di tutta la memoria utilizzata da un particolare processo giornale.

Le informazioni memorizzate nella tabella IBMSNAP\_CAPMON forniscono statistiche operative che consentono di ottimizzare l'utilizzo della memoria. Notare che i valori in questa tabella non sono cumulativi tra gli intervalli di controllo ma sono relativi ad un particolare intervallo di controllo Capture. I dati nella colonna CURRENT\_MEMORY non contengono un conteggio aggiuntivo. Esso fa riferimento alla memoria in utilizzo alla fine dell'intervallo di controllo quando è creato un record. L'intervallo di controllo Capture determina la frequenza con cui il programma Capture inserisce i dati in questa tabella. Utilizzare uno dei metodi riportati di seguito per ottimizzare la quantità di memoria utilizzata dal programma Capture:

#### **Ottimizzare il limite di memoria per permettere i trasferimenti:**

1. Quando si avvia il programma, utilizzare il limite di memoria predefinito.
2. Controllare se i dati trasferiti dalla memoria ad un file temporaneo visualizzando la colonna TRANS\_SPILLED nella tabella IBMSNAP\_CAPMON. Questa colonna mostra il numero delle transazioni del sistema di origine che vengono trasferiti sul disco a cause delle restrizioni di memoria durante un intervallo di controllo Capture particolare.
3. Se i dati trasferiti dalla memoria utilizzano un limite di memoria maggiore o un minore intervallo di commit.

#### **Ottimizzazione del limite di memoria per impedire i trasferimenti:**

1. Quando viene avviato il programma Capture, utilizzare il limite di memoria predefinito. (il numero dipende dalle risorse del proprio sistema.)
2. Controllare la quantità di memoria utilizzata visualizzando la colonna CURRENT\_MEMORY nella tabella IBMSNAP\_CAPMON. Questa colonna mostra la quantità di memoria (in bytes) che il programma Capture ha utilizzato durante un particolare intervallo di controllo Capture.
3. Se viene utilizzata una quantità di memoria più piccola di quella specificata per il limite di memoria, impostare un valore più piccolo per il limite di memoria.

## Memoria utilizzata dal programma Apply

Il programma Apply utilizza memoria quando inserisce i dati. La quantità di memoria utilizzata è proporzionale alla dimensione delle colonne della tabella e il numero di righe inserite in una volta. Ad esempio, se il programma Apply sta inserendo una colonna LOB, potrebbe utilizzare 2 GB di memoria.

L'informazione riguardo le serie di richieste attive è letta e memorizzata quando il programma Apply è in esecuzione. La quantità di memoria utilizzato in una volta dal programma Apply è generalmente proporzionale alla quantità di memoria richiesta per elaborare la serie di richieste che presenta il numero di membri più alto.

---

## Pianificazione memorizzazione

La pianificazione della memorizzazione è importante per l'influenza della registrazione dei server di origine DB2, per l'influenza della registrazione per server di registrazione, per i requisiti di memorizzazione delle tabelle destinazione e delle tabelle di controllo, per i requisiti di spazio per i file diagnostici (Linux, UNIX, Windows, z/OS), per i requisiti di spazio per il caricamento dei file per il programma Capture e per i requisiti di spazio per il caricamento dei file per il programma Apply.

Oltre alla memorizzazione richiesta per DB2, è necessario garantire che la memorizzazione sia disponibile per la replica per i seguenti argomenti. Tutte le dimensioni indicate in questi argomenti sono solamente stimate. Per preparare e progettare un sistema production-ready, è inoltre necessario considerare alcuni elementi come ad esempio la protezione dal fallimento. Ad esempio, è necessario aumentare il periodo di mantenimento dei dati per considerare possibili interruzioni della rete.

**Suggerimento:** Se la stima di memorizzazione appare eccessivamente alta, riesaminare la frequenza con cui il programma Apply avvia serie di richieste e la frequenza con cui le tabelle di replica sono eliminate. È necessario considerare i trade-off fra l'utilizzo della memorizzazione, la capacità di tolleranza del fallimento e il sovraccarico della CPU.

## Influenza della registrazione per il server di origine DB2

In generale è necessario triplicare il volume della registrazione per tutte le tabelle interessate nella replica. In sostanza, è necessario uno spazio di registrazione per la tabella origine come anche la tabella CD e la replica delle tabelle di controllo. Questa sezione fornisce altri fattori che possono aiutare a compiere una stima più accurata dell'influenza della registrazione che si prevede nel proprio ambiente replica.

Considerare gli aggiornamenti compiuti all'origine del database dalle proprie applicazioni e i requisiti di replica. Ad esempio, se un'applicazione di aggiornamento aggiorna generalmente il 60% delle colonne in una tabella, i requisiti di replica possono provocare la crescita dei record di registrazione di più della metà rispetto ad una tabella simile che non è replicata.

#### z/OS

#### Linux UNIX Windows

- DB2 esegue la registrazione delle immagini di riga completa per ogni istruzione UPDATE. Questo accade perché, prima che si possa replicare una tabella, è necessario crearla (o modificarla) con la parola chiave DATA CAPTURE CHANGES.
- Uno dei requisiti di replica che aggiungono con maggior frequenza al file di log è l'acquisizione di pre- e post-immagini (come per la replica delle tabelle di destinazione in scenari di replica update-anywhere). Un modo per ridurre il volume della registrazione è quello di ridurre il numero di colonne definite per l'origine della replica. Ad esempio, non acquisire le pre-immagini se non è richiesto.

#### System i

- DB2 esegue la registrazione delle immagini di riga completa per ogni istruzione UPDATE. Un modo per ridurre il volume della registrazione è quello di ridurre il numero di colonne definite per l'origine di replica, ad esempio, non acquisire pre-immagini se non è richiesto.
- Per ridurre al minimo la quantità di memoria utilizzata per le tabelle CD e per le tabelle UOW, riorganizzare spesso queste tabelle perché l'eliminazione non ripristina DASD automaticamente. È possibile utilizzare la parola chiave RGZCTLTBL (Riorganizza tabelle di controllo) sul comando ENDDPRCAP per riorganizzare le tabelle di controllo. Osservare i modelli d'utilizzo DASD in condizioni operative normali per consentire di prevedere e gestire l'utilizzo di DASD. Se la registrazione su giornale è attiva, tiene in considerazione anche che il volume di registrazione o di giornale aumenta come gli inserimenti e le eliminazioni DB2 dalla tabella UOW e dalle tabelle CD.
- Quando il ricevitore corrente è pieno, il sistema passa ad un nuovo ricevitore; facoltativamente è possibile salvare o eliminare i vecchi ricevitori non più necessari alla replica. Quando un sistema gestisce un grande numero di transazioni, il programma Capture può occasionalmente rallentarsi. Se Capture si rallenta spesso, è possibile dividere le proprie tabelle di origine in periodi multipli per distribuire il carico di lavoro a più istanze del programma Capture.

## Influenza della registrazione per i server di destinazione

Oltre alla registrazione del database di origine, esiste anche la registrazione del database di destinazione, nella quale vengono applicate le righe. L'influenza della registrazione dipende dalla modalità di commit scelta per il programma Apply.

### Modalità tabella

Nell'elaborazione della modalità tabella, il programma Apply effettua un commit singolo dopo che i file inseriti vengono applicati. Il programma Apply non effettua checkpoint intermedi. In questo caso, è necessario stimare la quantità massima di dati che il programma Apply elaborerà durante un intervallo di tempo e regolare lo spazio di registrazione per accordare quella quantità di dati.

### Modalità transazione

Nell'elaborazione della modalità transazione, il programma Apply copia

ogni aggiornamento nell'ordine della transazione d'origine alle tabelle di destinazione ed esegue il commit di queste modifiche su un limite di transazione in un intervallo. Impostare l'intervallo per i commit temporanei impostando il valore di  $x$  nelle opzioni della serie di richieste **commit\_count(x)**. Dopo che il programma Apply ha inserito tutte le serie di risposte, applica il contenuto dei file di trasferimento nell'ordine della sequenza di commit. Questo tipo di elaborazione permette di aprire ed elaborare tutti i file di trasferimento nello stesso momento. Ad esempio, se si imposta il conteggio di commit a 1, il programma Apply esegue il commit dopo ogni transazione, se si imposta il conteggio di commit a 2, esegue il commit dopo ogni serie di due transazioni.

**System i** È necessario anche considerare lo spazio di registrazione (spazio dei ricevitori di giornale) delle tabelle di destinazione. Poiché i ricevitori di giornale per le tabelle di destinazione su System i possono essere creati con i parametri **MNGRCV(\*SYSTEM)** e **DLTRCV(\*YES)** e poiché è necessario inserire nel giornale le colonne post-immagine, utilizzare la seguente formula per stimare il volume dei ricevitori di giornale per le tabelle di destinazione:

```
journal_receiver_volume=target_table_row_length X journal_receiver_threshold
```

## Requisiti di memorizzazione delle tabelle di destinazione e delle tabelle di controllo

È necessario stimare il volume delle nuove tabelle di destinazione. Lo spazio richiesto per una tabella di destinazione solitamente non supera quello della tabella di origine, ma può essere maggiore se la tabella di destinazione viene denormalizzata o se include pre-immagini (oltre a post-immagini) o dati cronologici. La dimensione della tabella di destinazione dipende da cosa si sceglie di replicare, ad esempio, la percentuale della tabella origine di cui si sta eseguendo la replica, il tipo di file delle colonne delle quali si sta effettuando la replica, se si sta effettuando la replica di post- o pre-immagini, se si stanno aggiungendo colonne calcolate, se si sta eseguendo un'impostazione secondaria delle righe, se sono effettuate delle trasformazioni durante la replica.

Anche tabelle CD e alcune tabelle di controllo replica (IBMSNAP\_UOW, IBMSNAP\_CAPTRACE, IBMSNAP\_APPLYTRACE, IBMSNAP\_APPLYTRAIL, IBMSNAP\_CAPMON, IBMSNAP\_ALERTS) influenzano lo spazio richiesto per le database DB2. Queste tabelle possono crescere molto in base all'impostazione dell'ambiente di replica. Lo spazio richiesto per le altre tabelle di controllo replica è generalmente di piccole dimensioni e statico.

Le tabelle CD aumentano di dimensione ad ogni modifica apportata alla tabella di origine fino a che il programma Capture elimina la tabella CD. Per stimare lo spazio richiesto per le tabelle CD, innanzitutto determinare per quanto tempo si vogliono mantenere i dati prima della loro eliminazione, poi specificare la frequenza con cui il programma Capture deve eliminare automaticamente queste tabelle e con quale frequenza verranno eliminate le tabelle utilizzando un comando.

Quando viene calcolato il numero di byte dei dati replicati, è necessario includere 21 byte per il sovraccarico dei dati per ogni riga aggiunta alle tabelle CD dal programma Capture. Determinare il periodo di tempo per il quale il programma Capture dovrebbe essere in grado di mantenere l'acquisizione dei dati all'interno delle tabelle CD, anche quando i dati non possono essere applicati, ad esempio, in caso di un'interruzione di rete. Stimare il numero di inserimenti, aggiornamenti ed

eliminazioni che generalmente dovrebbero essere acquisite per la tabella di origine all'interno del periodo di tempo di contigenza.

Per determinare la dimensione raccomandata per la tabella CD, utilizzare la seguente istruzione:

```
recommended_CD_size =  
  ( (21 bytes) + sum(lunghezza di tutte le colonne registrate) ) X  
  (numero di inserimenti, aggiornamenti, ed eliminazioni dalla tabella di origine  
  durante il periodo di contigenza)
```

## Esempio

Se le righe nella tabella CD sono lunghe 100 bytes (più dei 21 bytes per il sovraccarico) e sono acquisiti 100,000 aggiornamenti durante un periodo di contigenza di 24 ore, la memorizzazione richiesta per la tabella CD è intorno ai 12 MB.

Le colonne registrate in questa formula includono sia le colonne pre-immagini che quelle post. Se gli aggiornamenti vengono convertiti in coppie di operazioni INSERT e DELETE, considerarli quando viene determinato il numero totale di inserimenti, aggiornamenti, ed eliminazioni. Ad esempio, contare ogni aggiornamento alle tabelle di origine come due righe all'interno della tabella CD.

La tabella UOW si espande o si riduce in base al numero di righe inserito dal programma Capture durante un particolare intervallo di commit e sul numero di righe eliminate. Ogni volta che una transazione di applicazione effettua un COMMIT, viene inserita una riga nella tabella UOW e la transazione esegue un'operazione INSERT, DELETE, o UPDATE rispetto ad una tabella di origine di replica registrata. Inizialmente è necessario sovrastimare lo spazio richiesto dalla tabella e controllare lo spazio attualmente utilizzato per determinare se può essere recuperato dello spazio.

## Requisiti di spazio per i file di trasferimento relativi al programma Capture

Se il programma Capture non dispone di memoria sufficiente, scrive (o trasferisce) le transazioni per trasferire i file. Il programma Capture scrive la transazione di dimensioni più elevate nel file; tuttavia, la transazione di dimensioni più elevate non è necessariamente quella che supera il limite di memoria.

### z/OS

I file di trasferimento vengono memorizzati su VIO (virtual I/O).

### Linux UNIX Windows

I file di trasferimento sono sempre su disco. Viene creato un file per transazione nella directory **capture\_path**.

### System i

I file di trasferimento sono creati nella libreria QTEMP, un file di trasferimento per ogni registrazione per il quale è necessario un file di trasferimento.

La dimensione dei file di trasferimento Capture dipende dai seguenti fattori:

### Limite della memoria

Utilizzare il parametro di funzionamento **memory\_limit** per specificare



quanta memoria può essere utilizzata per il programma Capture. Maggiore è la memoria consentita, minore è la possibilità che Capture scriva i file di trasferimento.

#### **Dimensione delle transazioni**

Transazioni di dimensioni più elevate potrebbero aumentare la necessità di scrivere i file di trasferimento.

#### **Numero di transazioni simultanee**

Se il programma Capture elabora più transazioni contemporaneamente, oppure elabora transazioni interconnesse, è necessario che il programma Capture memorizzi più informazioni nella memoria o sul disco.

#### **Intervallo di commit**

Generalmente minore è l'intervallo di commit, minore sarà la necessità di memorizzazione poiché Capture memorizza informazioni nella memoria per un periodo di tempo minore del tempo di esecuzione del commit.

## **Requisiti di spazio per i file di trasferimento relativi al programma Apply**

Il programma Apply richiede uno spazio temporaneo per memorizzare dati. (Se si utilizza il programma di utilità ASNLOAD, è possibile avere un file input di caricamento anziché un file file di trasferimento di caricamento.) Il programma Apply utilizza i file di trasferimento per mantenere gli aggiornamenti fino a che non vengono applicati alle tabelle di aggiornamento. In generale, i file di trasferimento sono file disco; tuttavia, sui sistemi operativi z/OS, è possibile specificare i dati trasferiti alla memoria. A meno che non ci siano vincoli di memoria virtuale, memorizzare i file di trasferimento nella memoria virtuale e non sul disco.

La dimensione del file di trasferimento è proporzionale alla dimensione dei dati selezionati per la replica durante ogni intervallo di replica. Generalmente il file di trasferimento è approssimativamente il doppio della dimensione dei dati. È possibile stimare la dimensione del file di trasferimento confrontandola con l'intervallo di frequenza (o il valore dei dati di blocco) pianificato per il programma Apply con il volume di modifiche nello stesso periodo (o in un periodo di picco di modifica).

**z/OS** **Linux UNIX Windows** La dimensione della riga del file di trasferimento è la dimensione della *riga di destinazione*, incluse tutte le colonne di sovraccarico della replica. La dimensione delle righe non è nel formato compatto interno di DB2, ma è nel formato di carattere interpretato (come quello inserito dal comando SELECT). La riga include anche una lunghezza di riga e dei null terminator sulle stringhe delle singole colonne. Il seguente esempio stima la dimensione dei file di trasferimento richiesta dai dati selezionati per la replica e non tiene conto dell'ulteriore spazio necessario per gli altri dati che vengono memorizzati nel file di trasferimento.

**System i** La riga del file di trasferimento ha una dimensione costante di 32 KB.

### **Esempio**

Se si modificano i picchi di volume a 12,000 aggiornamenti all'ora e la frequenza del programma Apply è pianificata per intervalli di un'ora, è necessario che il file di trasferimento mantenga un valore di aggiornamento ad ogni ora, oppure a

12,000 aggiornamenti. Se ogni aggiornamento rappresenta 100 byte dei dati, il file di trasferimento sarà approssimativamente 1.2 MB al minimo. Ulteriore spazio è richiesto per gli altri dati che vengono memorizzati nel file di trasferimento.

## Requisiti di spazio per i file di log diagnostica (z/OS, Linux, UNIX, Windows)

z/OS

Linux UNIX Windows

I file di log di diagnostica memorizzano informazioni riguardo le attività dei programmi di replica, quali l'ora dell'avvio e dell'arresto del programma e altre informazioni o messaggi di errore. Per impostazione predefinita, il programma aggiunge i messaggi al proprio file di log, anche dopo il riavvio del programma. Verificare che le directory che contengono tali file di log dispongano di spazio sufficiente per memorizzare i file.

L'ubicazione dei file di log di diagnostica dipende dal valore impostato per i parametri di avvio **capture\_path**, **apply\_path** e **monitor\_path** quando è stato avviato il programma Capture, il programma Apply e il programma Replication Alert Monitor.

Per eventuali problemi di memorizzazione, è possibile riutilizzare le registrazioni del programma in modo che ad ogni avvio del programma, la registrazione venga cancellata e ricreate. È possibile specificare se si desidera riutilizzare la registrazione quando si avvia il programma.

---

## Pianificazione rilevazione di conflitti

Se si utilizza una rilevazione di conflitto standard o avanzata, è necessario memorizzare le pre-immagini nelle tabelle CD (o CCD) per la replica delle tabelle di destinazione. Anche le regole d'integrità referenziale sono limitate. Negli scenari peer-to-peer e update-anywhere, o quando il programma Apply utilizza la modalità elaborazione transazione, è necessario definire le regole d'integrità referenziale che sono conservate insieme alle regole di origine. Se si utilizza una replica peer-to-peer o una replica update-anywhere e non si desidera attivare la rilevazione di conflitto, è necessario progettare l'ambiente dell'applicazione per impedire che i conflitti si aggiornino. Se non è possibile che si verifichino conflitti all'interno del proprio ambiente di applicazione, è possibile salvare i cicli di elaborazione senza utilizzare la rilevazione di conflitto.

Utilizzare uno dei seguenti metodi per impedire conflitti nella replica peer-to-peer o update-anywhere:

### Frammentazione per chiave

Progettare l'applicazione affinché la replica di origine sia aggiornata dalle repliche per gamma di chiavi in specifici siti. Ad esempio, il sito di New York può aggiornare i record di vendita solo per quanto riguarda l'est degli Stati Uniti (utilizzando i codici ZIP <sup>1</sup> minori o uguali a 49999 come gamma di chiavi), ma può leggere tutti i record.

### Frammentazione per tempo

Progettare l'applicazione affinché la tabella possa essere aggiornata solo durante intervalli di tempo specifici in specifici siti. I periodi di tempo devono essere sufficientemente distanti per permettere alla replica delle

---

1. codici postali degli Stati Uniti

modifiche richieste di essere effettuate nei siti che stanno diventando la versione principale. Si ricordi di permettere le modifiche di tempo, come Daylight Savings Time o Summer Time e per differenze di time-zone.

---

## Pianificazione origini relazionali Non-DB2

I trigger Capture sono utilizzati invece del programma Capture se si effettua la replica da database relazionali non-DB2. Questi trigger acquisiscono i dati modificati da una tabella relazionale di origine non-DB2 ed eseguono il commit dei dati modificati all'interno delle tabelle CCD.

I trigger Capture influenzano la velocità di trasferimento della transazione e i requisiti di spazio della registrazione. Inoltre, se si dispone di trigger esistenti all'interno dell'ambiente, potrebbe essere necessario unirli con i nuovi trigger Capture. Per ulteriori informazioni, consultare le seguenti sezioni:

### Velocità di trasmissione della transazione per i trigger Capture

Il carico di lavoro della transazione per il sistema di origine aumenterà; l'acquisizione di una modifica basata sul trigger influisce sulla velocità di trasferimento della transazione.

I trigger di Capture aumentano il tempo di risposta per le transazioni in aggiornamento. L'influenza è maggiore per quelle transazioni che aggiornano in grande misura le tabelle di origine che stanno per essere replicate.

### Influenza della registrazione per i server di origine relazionali non-DB2

Per quanto riguarda i server di origine relazionale non-DB2, saranno necessari più spazi di registrazione attiva per le applicazioni di origine poiché il volume di registrazione, approssimativamente, triplica per le tabelle di origine replicate. Le modifiche sono acquisite dai trigger nelle tabelle di origine e memorizzate nelle tabelle CCD, i dati modificati sono scritti senza la stessa durata di commit scope come le tabelle di modifica origine, poi sono eliminati attraverso un meccanismo di eliminazione basato su un trigger.

Ogni origine INSERT, UPDATE, oppure operazione DELETE diventa un'operazione INSERT, UPDATE, o DELETE, più un'operazione INSERT, più un'operazione DELETE. Il volume della registrazione aumenta maggiormente se vengono modificati gli aggiornamenti con coppie di operazioni DELETE e INSERT.

Se lo spazio di registrazione è quasi esaurito e il trigger Capture non può inserire un record nella tabella CCD, la transazione tentata dall'utente o da un programma applicativo non sarà completata con successo.

### Coesistenza di trigger esistenti con trigger Capture

La logica dei trigger Capture è contenuta nello script SQL generato dal Centro di replica quando viene registrata un'origine.

Come impostazione predefinita vengono creati un trigger INSERT, un trigger UPDATE e un trigger DELETE così questi tipi di modifiche (insert, update, delete) possono essere replicate dalla tabella origine. Il nome del trigger è composto dal nome della tabella CCD preceduta da una lettera che descrive il tipo di trigger: I per INSERT, U per UPDATE, D per DELETE. Ad esempio, se il nome della tabella

CCD è undjr02.ccd001, il nome del trigger DELETE generato è undjr02.dccd001. È necessario non modificare i nomi dei trigger che vengono generati nello script.

Se un trigger già esiste sulla tabella che si desidera registrare per la replica e quel trigger ha lo stesso nome di un trigger incluso nello script generato, si riceverà un avviso quando verrà generato lo script. Non avviare lo script generato poiché RDBMS potrebbe sovrascrivere il trigger esistente. Determinare come si desiderano unire i trigger preesistenti con i nuovi trigger e creare uno script che unisce la logica esistente con la logica del trigger generato dal Centro di replica.

Se il tipo di trigger che si desidera creare già esiste sulla tabella che si desidera registrare per la replica e RDBMS permette solo uno di questi trigger per tabella, è necessario unire la logica prima di avviare lo script generato.

## Blocchi per i server di origine Oracle

È necessario che le applicazioni che stanno attualmente aggiornando l'origine di Oracle siano terminate prima che il programma Apply possa iniziare ad applicare i dati.

È necessario che il programma Apply blocchi la tabella CCD, affinché sia in grado di elaborare i dati e impostare il proprio punto di sincronizzazione. I blocchi sulle tabelle CCD vengono conservati fino a quando il programma Apply imposta il proprio punto di sincronizzazione, e non per l'intero ciclo Apply. È necessario che le applicazioni delle quali è necessario aggiornare la tabella origini attendano fino a che il programma Apply sblocchi la tabella CCD.

---

## Pianificazione della traduzione di tabelle codici

I componenti di replica sono delle applicazioni database che si basano sui database DB2 su vari sistemi operativi per gestire la traduzione dei dati che utilizzano tabelle codici differenti.

I componenti di replica operano con i dati utilizzando le istruzioni SQL SELECT, INSERT, UPDATE e DELETE.

## Replica tra database con tabelle codici compatibili

Se la configurazione della replica richiede istruzioni SQL e il passaggio di dati fra sistemi con differenti tabelle codici, i protocolli DB2 sottostanti come ad esempio DRDA gestiscono la traduzione di tabelle codici. Inoltre, se i dati sono trasmessi fra database relazionali DB2 e non-DB2, la replica DB2 si basa sui prodotti database sottostanti per gestire le traduzioni di tabelle codici necessarie.

Se si pianifica di eseguire una replica tra database con tabelle codici differenti, consultare il *Centro informazioni IBM Information Management Software per z/OS Solutions* o il *Centro informazioni DB2* per determinare se le tabelle codici sono compatibili. Per esempio, se si utilizza DB2 per Linux, UNIX e Windows, consultare la sezione relativa alla traduzione dei dati carattere.

Una volta verificato che il database dispone di tabelle codici compatibili, determinare se il database utilizza tabelle codici differenti. Ad esempio, supponiamo che un prodotto per database consenta una tabella codici differente per ogni colonna di una tabella mentre un altro database richieda di specificare la tabella codici solo a livello di database. Una tabella con più tabelle codici nel primo prodotto non può essere replicata ad un singolo database nel secondo prodotto. Per questo motivo, il modo in cui i database gestiscono le tabelle codici

influenza il modo in cui impostare la replica per assicurarsi che i dati siano replicati con successo tra i vari database nel proprio ambiente.

## **Tabelle codici per la replica SQL**

La configurazione di tabelle codici per la replica viene definita all'atto dell'impostazione delle connettività del database fra i sistemi. Tuttavia, se i programmi Capture o Apply sono in esecuzione su sistemi operativi Linux, UNIX o Windows, potrebbero essere necessarie alcune operazioni di configurazione.

Su Linux, UNIX e Windows, il programma Capture deve essere in esecuzione nella stessa tabella codici del database dal quale sta acquisendo i dati. Se il programma Capture non è in esecuzione nella stessa tabella codici, è necessario impostare una variabile d'ambiente DB2 o una variabile di registro chiamata DB2CODEPAGE così Capture utilizza la stessa tabella codici del database.

Quando il programma Apply è in esecuzione su Linux, UNIX o Windows, se le tabelle di origine sono in UNICODE, è necessario che il codice dell'applicazione Apply sia in UNICODE. Se i dati nella tabella di origine sono in ASCII, la codepage dell'applicazione può essere in ASCII o in UNICODE. È anche possibile impostare la variabile DB2CODEPAGE per il programma Apply.

### **Impostazione della variabile codepage**

DB2 ottiene la tabella codici per un'applicazione dall'ambiente attivo nel quale l'applicazione è in esecuzione. Generalmente, quando la variabile DB2CODEPAGE non è impostata, la tabella codici è ottenuta da l'ID linguaggio specificato dal sistema operativo. Nella maggior parte delle situazioni, questo valore è corretto per i programmi Capture o Apply se si utilizza la tabella codici predefinita quando viene creato il database. Tuttavia, se si crea un database con una tabella codici particolare, differente dalla tabella codici predefinita, è necessario impostare la variabile DB2CODEPAGE. Altrimenti, i dati potrebbero non essere tradotti correttamente. È necessario che il valore utilizzato per la variabile DB2CODEPAGE sia lo stesso specificato nell'istruzione CREATE DATABASE. Consultare il DB2 Information Center per dettagli riguardo l'impostazione della variabile DB2CODEPAGE.

### **Replica da una tabella codici**

Se si sta effettuando la replica dei dati di origine con una tabella codici SBCS (single-byte character set) ad una destinazione con Unicode UTF-8, alcuni single-byte characters nel database di origine potrebbero essere tradotti da DB2 a due o più byte nel database di destinazione. Tutti i single-byte characters che hanno un valore esadecimale tra 0x80 e 0xff sono tradotti con i loro equivalenti two-byte 1208. Questo significa che è necessario che le colonne di destinazione siano più grande delle colonne di origine, altrimenti il programma Apply potrebbe ricevere errori SQL da DB2.

Alcuni prodotti database, a differenza di altri, implementano un supporto tabelle codici che può influire sulla configurazione della replica. Ad esempio, DB2 su System i permette di specificare una tabella codici a livello di colonna, ma DB2 per Linux, UNIX e Windows permette di specificare una tabella codici solamente a livello di database. Per questo motivo, se si dispone di una tabella System i con più colonne utilizzando differenti tabelle codici, queste colonne non possono essere replicate ad un singolo database DB2 per Linux, UNIX e Windows a meno che tutte le tabelle codici siano compatibili.

### Impostazione variabile LANG

Se si stanno eseguendo programmi Capture e Apply su un sistema Linux o UNIX, potrebbe essere necessario impostare la variabile d'ambiente LANG. I programmi Capture e Apply utilizzano i contenuti di questa variabile d'ambiente per trovare la loro libreria messaggi per la propria lingua. Ad esempio, se la variabile d'ambiente LANG è impostata su en\_US, il programma Capture cerca la sua libreria messaggi Inglese nella sottodirectory dell'istanza DB2 /sqlib/msg/en\_US . Se Capture non trova la sua libreria messaggi, tutti i messaggi scritti sulla tabella IBMSNAP\_CAPTRACE sono ASN0000S.

---

## Pianificazione della replica per DB2 per z/OS

### z/OS

La replica SQL per DB2 per z/OS supporta nomi di schema e tabella fino a 128 byte.

Per sfruttare il supporto per nomi lunghi:

- Creare le tabelle di controllo Capture, Apply e Monitor in DB2 per z/OS Versione 8 o successiva in modalità nuova funzione.
- Eseguire i server Capture, Apply e Monitor in DB2 per z/OS Versione 8 o successiva in modalità nuova funzione

**Limitazione:** Se si desidera eseguire la replica fra i sottosistemi della modalità nuova funzione di DB2 per z/OS e DB2 per Linux, UNIX, Windows, oppure DB2 per iSeries, è necessario utilizzare nomi schema di 30 byte o meno. Utilizzando nomi schema maggiori di 30 caratteri in modalità nuova funzione di DB2 per z/OS Versione 8, non sarà possibile eseguire la replica fra tale piattaforma e DB2 per Linux, UNIX, Windows, oppure DB2 per iSeries.

---

## Ottimizzazione delle prestazioni

L'ottimizzazione delle prestazioni consiste nell'ottimizzazione dell'ambiente di replica per le prestazioni ottimali.

*WebSphere Information Integration Tuning per SQL Replication Performance* descrive in che modo ottimizzare i componenti principali di un ambiente di replica DB2 per le prestazioni massime. Questo documento è disponibile in linea al sito di supporto WebSphere Information Integration per il proprio prodotto.

---

## Capitolo 2. Requisiti di autorizzazione per la replica SQL

Per utilizzare i programmi di replica SQL, è necessario garantire che gli ID utente che utilizzano i programmi di replica o gli strumenti di replica dispongano dell'autorizzazione corretta sui sistemi locali e remoti.

---

### Requisiti di autorizzazione per l'amministrazione

Per impostare la replica, viene eseguito l'SQL generato per creare oggetti, viene eseguito il bind dei piani e vengono creati i pacchetti SQL (System i). Le autorizzazioni richieste per queste attività variano a seconda del sistema operativo.

Per amministrare la replica, è necessario avere almeno un ID utente su tutti i database interessati nella configurazione della replica e questo ID utente deve avere l'autorizzazione per impostare la replica. Il proprio ID utente non deve lo stesso su tutti i sistemi, sebbene, se lo fosse, potrebbe essere più semplice per l'utente.

z/OS

Linux UNIX Windows

Verificare che gli ID utente utilizzati per impostare la replica possano eseguire le seguenti attività:

- Collegarsi a tutti i server (server di origine, server di controllo Capture, server di controllo Apply, server di controllo Monitor, server di destinazione).
- Selezionare dalle tabelle del catalogo nel server di origine, sul server di controllo Capture, sul server di controllo Monitor e sul server di destinazione.
- Creare tabelle (incluse tabelle di controllo della replica), tablespace e viste nel server di origine, nel server di controllo Monitor, nel server di controllo Capture, e nel server di controllo Apply.
- Se vengono utilizzati programmi di replica per creare nuove tabelle di destinazione: creare tabelle e tablespace nel server di destinazione. (Non richiesto se vengono utilizzate tabelle esistenti come destinazioni).
- Eseguire il bind dei piani o creare pacchetti su ogni database DB2 coinvolto nella replica, incluso il server di origine, il server di destinazione, il server di controllo Monitor e il server di controllo Apply.
- Creare procedure memorizzate utilizzando una libreria condivisa e richiamare le procedure memorizzate (solo Linux, UNIX e Windows).

Per database relazionali non DB2, l'ID utente deve essere in grado di eseguire le seguenti azioni:

- Creare tabelle.
- Creare trigger Capture nelle tabelle di origine e nelle tabelle di controllo.
- Creare procedure.
- Creare nickname nel database federato.
- Creare sequenze (solo per database Oracle).
- Selezionare dalle tabelle del catalogo.

La maggior parte degli amministratori di replica godono di privilegi DBADM o SYSADM. Su DB2 per z/OS l'amministratore di replica deve

essere almeno autorizzato a selezionare dal catalogo e deve godere di tutti i privilegi necessari per creare tabelle con lo schema ASN e per creare tabelle CD e di destinazione con le caratteristiche delle tabelle di origine, inclusi i privilegi per la creazione degli indici.

#### System i

Verificare che gli ID utente utilizzati per impostare la replica possano eseguire le seguenti attività:

- Collegarsi a tutti i server (server di origine, server di controllo Capture, server di controllo Apply, server di controllo Monitor, server di destinazione).
- Selezionare dalle tabelle del catalogo nel server di origine, sul server di controllo Capture, sul server di controllo Monitor e sul server di destinazione.
- Creare tabelle (incluse tabelle di controllo della replica) e viste nel server di origine, nel server di controllo Monitor, nel server di controllo Capture, e nel server di controllo Apply.
- Se vengono utilizzati i programmi di replica DB2 per creare nuove tabelle di destinazione: creare tabelle sul server di destinazione. (Non richiesto se vengono utilizzate tabelle esistenti come destinazioni).
- Eseguire il bind dei piani o creare pacchetti su ogni database DB2 coinvolto nella replica, incluso il server di origine, il server di destinazione, il server di controllo Monitor e il server di controllo Apply.

La maggior parte degli amministratori di replica godono di privilegi DBADM o SYSADM.

Utilizzare il comando GRTPRAUT (Concessione autorizzazione DPR) per autorizzare un utente alla registrazione di origini, alla sottoscrizione di tali origini e alla creazione di tabelle di controllo. Se si effettua la replica solo tra sistemi System i, è necessario utilizzare lo stesso ID utente per tutti i server.

Se il comando GRTPRAUT (Concessione autorizzazione DPR) non è installato su una macchina, è necessario utilizzare il comando GRTOBJAUT (Concessione autorizzazione oggetto).

---

## Requisiti di autorizzazione per il programma Capture

L'ID utente che esegue il programma Capture deve essere in grado di accedere al catalogo di sistema DB2, accedere e aggiornare tutte le tabelle di controllo della replica sul server di controllo Capture ed eseguire i pacchetti del programma Capture.

È possibile utilizzare l'ID utente dell'amministratore di replica per eseguire il programma Capture, ma questo non è un requisito.

#### z/OS

L'ID utente utilizzato per eseguire il programma Capture deve essere registrato con l'accesso a USS. Questo significa che l'ID utente deve essere definito per l'utilizzo di z/OS UNIX o OS/390 UNIX (deve disporre di un segmento OMVS).

Inoltre, verificare che la libreria per il caricamento di Capture sia APF-authorized e che l'ID utente che esegue il programma Capture disponga dei seguenti privilegi:



- WRITE accede ad una directory temporanea; sia la directory /tmp che la directory specificata che la variabile di ambiente TMPDIR.
- I privilegi SELECT, UPDATE, INSERT e DELETE per tutte le tabelle di replica sul server di controllo Capture.
- Il privilegio SELECT per il catalogo DB2 (SYSIBM.SYSTABLES, SYSIBM.SYSCOLUMNS. e SYSIBM.SYSPLAN).
- Privilegio TRACE.
- Privilegi MONITOR1 e MONITOR2.
- Privilegio EXECUTE per i pacchetti del programma Capture.

Inoltre, verificare che l'ID utente abbia l'accesso WRITE alla directory di percorso Capture (USS) o al qualificatore di alto livello (z/OS). Per eseguire il programma Capture in una shell USS, la variabile di sistema STEPLIB deve essere impostata e deve includere la libreria per il caricamento di Capture. Il percorso HFS, /usr/lpp/db2repl\_09\_01/bin, deve essere nel PATH dell'utente.

#### Linux UNIX Windows

Garantire che gli ID utente che eseguono il programma Capture dispongano delle seguenti autorizzazioni e dei seguenti privilegi:

- autorizzazione DBADM o SYSADM.
- Privilegio WRITE nella directory specificata dal parametro **capture\_path**. Il programma Capture crea i file diagnostici in questa directory.

#### Windows

L'ID utente che esegue il programma Capture deve essere autorizzato per creare oggetti globali.

#### System i

Utilizzare il comando GRTPRAUT (Concessione autorizzazione DPR) per autorizzare un utente all'esecuzione del programma Capture su un sistema locale. Se si effettua la replica solo tra sistemi System i, è necessario utilizzare lo stesso ID utente per tutti i server. Se il comando GRTPRAUT non è installato su una macchina, è necessario utilizzare il comando GRTOBJAUT (Concessione autorizzazione oggetto).

---

## Requisiti di autorizzazione per il programma Apply

L'ID utente che esegue il programma Apply deve essere in grado di accedere al catalogo di sistema DB2, accedere e aggiornare tutte le tabelle di controllo della replica sul server di controllo Capture e sul server di destinazione ed eseguire i pacchetti del programma Apply.

È possibile utilizzare differenti ID utente in ogni server nel proprio ambiente di replica. È possibile utilizzare l'ID utente dell'amministratore di replica per eseguire il programma Apply, ma questo non è un requisito.

#### z/OS

Garantire che gli ID utente che eseguono il programma Apply dispongano delle seguenti autorizzazioni e dei seguenti privilegi:

- WRITE accede ad una directory temporanea; sia la directory /tmp che la directory specificata che la variabile di ambiente TMPDIR.
- Privilegi SELECT, UPDATE, INSERT e DELETE per tutte le tabelle di replica sul server di controllo Apply.

- Autorizzazione SELECT per il catalogo DB2 (SYSIBM.SYSTABLES, SYSIBM.SYSCOLUMNS. e SYSIBM.SYSPLAN).

**Nota:** l'ID utente utilizzato per eseguire il programma Apply deve essere registrato con l'accesso a USS. Questo significa che l'ID utente deve essere definito per l'utilizzo di z/OS UNIX o OS/390 UNIX (deve disporre di un segmento OMVS). La libreria per il caricamento deve essere APF-authorized solo se il programma Apply deve essere registrato con ARM. Per eseguire il programma Apply in una shell USS, la variabile di sistema STEPLIB deve essere impostata e deve includere la libreria per il caricamento di Apply. Il percorso HFS, /usr/lpp/db2repl\_09\_01/bin, deve essere nel PATH dell'utente.

#### Linux UNIX Windows

Garantire che gli ID utente che eseguono il programma Apply dispongano delle seguenti autorizzazioni e dei seguenti privilegi:

- Privilegi WRITE per la directory del percorso di Apply
- Privilegi di accesso per le tabelle di origine della replica (incluse le tabelle CD e CCD associate).
- Privilegi di accesso e aggiornamento per le tabelle di destinazione della replica.
- Privilegi di accesso a aggiornamento per tutte le tabelle di controllo generate dai programmi di replica e create nel server di controllo Capture e nel server di controllo Apply.
- Privilegi READ per ogni file di password utilizzato dal programma Apply.

**Nota:** se le proprie tabelle di origine sono in un sistema di gestione database relazionali non DB2: l'ID utente deve disporre di sufficienti privilegi sia nel database federato che nel database relazionale non DB2 per accedere alle tabelle di origine attraverso i nickname, definiti sul database federato.

#### Windows

L'ID utente che esegue il programma Apply deve essere autorizzato per creare oggetti globali.

#### System i

Utilizzare il comando GRDPRAUT (Concessione autorizzazione DPR) per autorizzare un utente all'esecuzione del programma Apply su un sistema locale. Se si effettua la replica solo tra sistemi System i, è necessario utilizzare lo stesso ID utente per tutti i server. Se il comando GRDPRAUT non è installato su una macchina, è necessario utilizzare il comando GRTOBJAUT (Concessione autorizzazione oggetto).

#### database non DB2

Se le proprie tabelle di controllo sono sui database non DB2, l'ID utente che avvicina i dati modificati ad una destinazione relazionale non DB2 o acquisisce i dati da questa deve avere sufficienti privilegi nel database federato e nel database relazionale non DB2.

Per destinazioni relazionali non DB2, non è necessario che l'ID utente che sta eseguendo il programma Apply disponga del privilegio WRITE per scrivere i nickname sul database federato e, attraverso associazioni utente, il privilegio WRITE per scrivere nella destinazione non DB2 attuale.

Per origini relazionali non DB2, è necessari l'ID che esegue il programma Apply disponga dei seguenti privilegi:

- Privilegio READ per leggere e WRITE per scrivere nickname sul database federato e, attraverso associazioni utente, il privilegio READ per leggere e WRITE per scrivere sulle tabelle di controllo Capture.
- Privilegio READ per leggere i nickname sul database federato e, attraverso associazioni utente, il privilegio READ per leggere dalla tabella CCD attuale sul server non DB2.
- Privilegio READ per leggere i nickname sul database federato e, attraverso associazioni utente, il privilegio READ per leggere dalla attuale tabella di origine sul server non DB2.

---

## Requisiti di autorizzazione per i trigger Capture sui database relazionali non DB2

Se si effettua la replica da un database non DB2, i trigger Capture vengono utilizzati per acquisire le modifiche dall'origine. È necessario che gli ID utente remoti (ad esempio, dalle applicazioni utente) che modificano le tabelle di origine remote dispongano delle autorizzazioni per effettuare gli inserimenti nella tabella CCD.

Nella maggior parte dei casi, non è necessaria un'autorizzazione esplicita per eseguire i trigger INSERT, UPDATE, o DELETE perché, dopo aver definito i trigger su una tabella, l'esecuzione dei trigger è trasparente all'applicazione che sta eseguendo i trigger INSERT, UPDATE o DELETE. Nel caso dei database Informix, è necessario che gli ID utente remoti che eseguono le azioni INSERT, UPDATE e DELETE per la tabella di origine registrata dispongano del privilegio EXECUTE PROCEDURE.

Per origini Oracle, è necessario concedere un privilegio SELECT sull'oggetto di sequenza *schema\_remoto.SEQUENCE002*, in cui *schema\_remoto* rappresenta lo schema remoto in base al quale vengono create le tabelle di controllo su Oracle. L'oggetto di sequenza viene creato durante la creazione delle tabelle di controllo Capture per un'origine Oracle e viene utilizzato con i trigger Capture per inserire popolare la tabella CCD.

---

## Managing user IDs and passwords for remote servers (Linux, UNIX, Windows)

### Linux UNIX Windows

Replication and event publishing require a password file in some cases to store user IDs and passwords for connecting to remote servers.

#### Informazioni su questa attività

A password file is required in the following cases:

- The Apply program requires a password file to access data on remote servers (the Capture program does not require a password file).
- The Q Apply program requires a password file to connect to the Q Capture server for Q subscriptions that use the EXPORT utility to load targets.
- The Q Capture program requires a password file to connect to multiple-partition databases.

- If the Q Capture program runs remotely from the source database or the Q Apply program runs remotely from the target database, the programs require password files to connect to the remote database.
- The asntdiff and asntrep commands require password files to connect to databases where the utilities are comparing or repairing table differences.
- The Replication Alert Monitor requires a password file to connect to any Q Capture, Capture, Q Apply, or Apply server that you want to monitor.

**Important note about compatibility of password files:** Password files that are created by the asnpwd command starting with Version 9.5 Fix Pack 2 use a different encryption method and cannot be read by older versions of the replication programs and utilities. If you share a password file among programs and utilities that are at a mixed level, with some older than these fix packs, do not recreate the password file by using an asnpwd command that is at these fix packs or newer. Replication programs and utilities at these fix packs or newer can continue to work with older password files. Also, you cannot change an older password file to use the later encryption method; you must create a new password file.

In general, replication and event publishing support the following scenarios:

- Creating a password file with one version and using it with a newer version. For example, you can create a password file under V8.2 and use it with V9.1 and V9.5.
- Creating a password file with one fix pack and using it with a newer fix pack within the same version. For example, you can create a password file with V9.1 Fix Pack 3 and use it with V9.1 Fix Pack 5.
- Creating a password file on one system and using it on another system as long as the following criteria are met:
  - The systems use the same code page.
  - The systems are all 32 bit or all 64 bit.

Encrypted password files are not supported for x64 Windows until 9.5 Fix Pack 2 or later.

## Procedure

To manage user IDs and passwords for remote servers, follow these guidelines:

- Create an encrypted password file for replication and event publishing programs that are running on Linux, UNIX, and Windows by using the asnpwd command. The password file must be stored in the path that is set by the following parameters:

*Tabella 1. Password file requirements*

Program	Parameter
Apply	<b>apply_path</b>
Q Apply	<b>apply_path</b>
Q Capture	<b>capture_path</b>
Replication Alert Monitor	<b>monitor_path</b>
asntdiff or asntrep command	<b>DIFF_PATH</b>

- If the Q Apply program and Replication Alert Monitor are running on the same system, they can share the same password file. If you want the programs to

share a password file, specify the same path and file name for the programs, or use symbolic links to share the same password file in the different directories.

- The Replication Center does not use the password file that is created with the `asnpwd` command to connect to remote servers. The first time that the Replication Center needs to access a database or subsystem, you are prompted for a user ID and password, which is stored for future use. You can use the Manage Passwords and Connectivity window to store user IDs for servers or systems, as well as to change the IDs that you stored and to test connections. To open the window, right-click the **Replication Center** folder and select **Manage Passwords for Replication Center**.



---

## Capitolo 3. Configurazione dei server per la replica SQL

Prima di poter replicare i dati, è necessario creare e configurare i server e garantire che essi possano eseguire la connessione reciprocamente.

**z/OS** Per ulteriori dettagli sulla configurazione di server in z/OS, consultare *Replication installation and customization for z/OS*.

---

### Requisiti di connettività per la replica SQL

È necessario che le workstation in cui viene eseguito il programma Apply, il Centro di replica, o i comandi di replica siano in grado di connettersi ad un server di origine, ad un server di controllo Capture, ad un server di controllo Apply e a server database di destinazione.

Se si utilizza Replication Alert Monitor, la stazione di lavoro sulla quale è in esecuzione deve essere in grado di connettersi al server di controllo Monitor ed ai server che controlla. Se si desidera utilizzare il Centro di replica per imposta il controllo, garantire che il Centro di replica possa connettersi al server di controllo Monitor.

Se la progettazione della replica interessa dati di staging in un server differente dal database di origine, è necessario prestare attenzione alle comunicazioni tra i vari server. Assicurarsi di limitare l'emulazione dei livelli, dei ponti LAN e dei collegamenti ai router richiesti, poiché questa situazione può influenzare la prestazione di replica.

Quando i database sono connessi ad una rete, la connettività varia a seconda dei sistemi operativi connessi.

I seguenti argomenti forniscono dettagli riguardo i requisiti di connettività.

### Connecting to System i servers from Windows

**System i**

You can administer replication on System i servers by connecting from a Windows workstation.

#### Prima di iniziare

- You must have a DB2 or DB2 Connect installed on your workstation.
- You must have TCP/IP set up on your workstation.

#### Procedure

To connect to a System i server from a DB2 for Windows workstation:

1. Log on to the System i server and locate the relational database:
  - a. Log on to the System i server to which you want to connect.
  - b. Submit a dsprdbdire command, then specify local for \*LOCAL.
  - c. Locate the name of the relational database in the output. For example, in the following output, the database is called DB2400E:

MYDBOS2	9.112.14.67
RCHASDPD	RCHASDPD
DB2400E	*LOCAL
RCHASLJN	RCHASLJN

2. Catalog the System i database in DB2 for Windows:
  - a. From a Windows command prompt, enter db2cmd. The DB2 CLP command window opens.
  - b. In the command window, type the following three commands in exact order:
 

```
db2 catalog tcpip node server_name remote server_name server 446 system
server_name ostype OS400

db2 catalog dcs database rdb_name AS rdb_name

db2 catalog database rdb_name AS rdb_name at node server_name
authentication dcs
```

Where *server\_name* is the TCP/IP host name of the System i system, and *rdb\_name* is the name of the System i relational database that you found in Step 1.
3. In the command window, issue the following command:
 

```
db2 terminate
```
4. Ensure that the System i user profile that you will use to log on to your System i system uses CCSID37:
  - a. Log on to the System i system.
  - b. Type the following command, where *user* is the user profile:
 

```
CHGUSRPRF USRPRF (user) CCSID(37)
```
  - c. Make sure that the DDM server is started on the System i system type:
 

```
STRTCPSVR SERVER(*DDM)
```
5. Make sure that DB2 for Windows and DB2 for System i are connected:
 

```
db2 connect to rdb_name user user_name using password
```

## Connessione ai server relazionali non-DB2

Se si desidera replicare i dati verso o da un server relazionale non-DB2, è necessario essere in grado di accedere al server relazionale non-DB2 e connettersi ad esso.

Prima di tentare di replicare dai server di origine non-DB2, è necessario impostare il server federato e il database. Esistono tre principali procedure di installazione:

1. Definire un wrapper in modo che il database DB2 possa accedere ad un altro database relazionale non-DB2.
2. Definire un database relazionale non DB2 utilizzando un'associazione server.

**Limitazione:** Il Centro di replica e il programma della riga comandi ASNCLP non supportano la creazione delle tabelle di controllo o delle tabelle di destinazione nei database Oracle se l'associazione server ha un Two Phase Commit abilitato.

3. Se l'ID utente e la combinazione password utilizzata per la connessione al database DB2 differisce da quella utilizzata per accedere al database relazionale non-DB2, è necessario creare una associazione utente.



Per ulteriori dettagli sulla configurazione di un ambiente federato, consultare il DB2 Information Center o la *WebSphere Information Integration Federated Systems Guide*.

---

## Creazione delle tabelle di controllo per la replica SQL

Il programma di replica utilizza le tabelle di controllo per memorizzare le informazioni riguardo le tabelle registrate, le serie di richieste, i parametri di funzionamento e le preferenze utente. Vengono create tabelle di controllo prima di definire le origini e le destinazioni per la replica.

### Creazione delle tabelle di controllo per la replica SQL

È possibile utilizzare il Centro di replica o il programma della riga comandi ASNCLP per creare tabelle di controllo per i programmi Capture e Apply.

#### Restrizioni

- È necessario che il Centro di replica o ASNCLP siano in grado di connettersi al server nel quale si desiderano creare le tabelle di controllo.
- In un ambiente con più partizioni del database, è necessario che tutti i tablespace utilizzati dalle tabelle di controllo si trovino nella partizione del catalogo. Se si utilizza un tablespace esistente, esso non deve essere partizionato e deve trovarsi nella partizione del catalogo.

#### Informazioni su questa attività

Se non si personalizza il modo in cui le tabelle di controllo sono create, sono creati due tablespace, uno per la tabella UOW e uno per le altre tabelle di controllo. Se non si desidera utilizzare i tablespace della replica predefinita, è possibile specificare tablespace esistenti, creare nuovi tablespace, o utilizzare il corrente tablespace predefinito DB2.

Se Capture è avviato in un ambiente di più partizioni del database, Capture crea un'ulteriore tabella di controllo (IBMSNAP\_PARTITIONINFO) nello stesso tablespace come la tabella IBMSNAP\_RESTART.

Entrambi gli strumenti di gestione consentono di creare un profilo per identificare i valori predefiniti da utilizzare quando vengono create delle tabelle di controllo per un determinato sistema operativo o ambiente. Dopo aver impostato i profili per queste tabelle di controllo, non è necessario impostarli per ogni serie di tabelle di controllo create. È possibile sovrascrivere i valori predefiniti quando si creano tabelle di controllo. È inoltre possibile modificare il profilo in qualsiasi momento, ma le modifiche intesseranno solo le tabelle di controllo create dopo aver modificato il profilo.

#### Procedure

Per creare tabelle di controllo, utilizzare uno dei seguenti metodi:

Metodo	Descrizione
Programma della riga comandi ASNCLP	Utilizzare il comando CREATE CONTROL TABLES FOR per creare una nuova serie di tabelle di controllo Capture o Apply. Ad esempio, i seguenti comandi impostano l'ambiente e creano tabelle di controllo Capture: <pre>SET SERVER CAPTURE TO DB SAMPLE SET OUTPUT CAPTURE SCRIPT "capctrl.sql"; SET LOG "capctrl.err"; SET RUN SCRIPT LATER; CREATE CONTROL TABLES FOR CAPTURE SERVER IN UW UOW TSUOW100 OTHERS TSASN100;</pre>
Centro di replica	Utilizzare le finestre Create Control Tables o Create Control Tables - Quick per Capture e Apply. Per aprire le finestre, fare clic con il tasto destro del mouse su una cartella Capture Control Servers o Apply Control Servers nella struttura ad albero degli oggetti e fare clic su una delle seguenti opzioni: <ul style="list-style-type: none"> <li>• Crea Capture Control Tables</li> <li>• Crea Capture Control Tables → Quick</li> <li>• Crea Apply Control Tables</li> <li>• Crea Apply Control Tables → Quick</li> </ul>

## Creating control tables (System i)

### System i

Replication control tables are created automatically when you install DB2 DataPropagator for System i. You can also use a command to create control tables.

#### Informazioni su questa attività

During the installation, control tables are created in the DataPropagator default schema (ASN), if they do not already exist. You can create additional sets of control tables if your control tables are accidentally deleted or corrupted. For Capture, you can create the new set of control tables with a different schema. You can create a maximum of 25 schemas.

For a user-defined file system, you can create the replication control tables in the base Auxiliary Storage Pool (ASP) or in Independent Auxiliary Storage Pool (IASP) groups, but not in both. If you create control tables in an IASP group, you must first remove all Capture and Apply control tables from the base ASP. Issue the SETASPGRP command for the ASP group that contains the ASN library (or any other library for a Capture schema) before you start the Capture or Apply programs.

#### Procedure

To create control tables on System i, use the Create DPR Tables (CRTDPRTBL) command.

**Restriction:** Use only the CRTDPRTBL command to create control tables on System i. The ASNCLP command-line program and Replication Center do not support the creation of control tables for System i.

## Creazione delle tabelle di controllo per le origini relazionali non-DB2

Se si desidera utilizzare un database non-DB2 come Informix come origine di una replica, è possibile utilizzare il Centro di replica o il programma della riga di comando ASNCLP per creare le tabelle di controllo.

### Informazioni su questa attività

Per questi tipi di origine, il Centro di replica crea le seguenti tabelle di controllo Capture nel database relazionale non-DB2:

- IBMSNAP\_PRUNCNTL
- IBMSNAP\_PRUNE\_SET
- IBMSNAP\_REG\_SYNCH
- IBMSNAP\_REGISTER
- IBMSNAP\_SEQTABLE on Informix solo
- IBMSNAP\_SIGNAL

I nickname sono creati in un database federato per tutti i tipi di origine tranne che per IBMSNAP\_SEQTABLE. (Questa tabella è utilizzata solamente dai trigger Informix. Il programma Apply non la utilizza). I trigger sono creati automaticamente nella tabella IBMSNAP\_SIGNAL e nella tabella IBMSNAP\_REG\_SYNCH.

**Importante:** Non rimuovere o modificare i trigger creati nella tabelle IBMSNAP\_SIGNAL e IBMSNAP\_REG\_SYNCH.

## Creazione di più serie di tabelle di controllo Capture

Se si desidera utilizzare più di un programma Capture su un server, è necessario creare più di una serie di tabelle di controllo Capture e garantire che ogni serie di tabelle abbia un solo schema Capture univoco.

### Informazioni su questa attività

Questo schema identifica il programma di Capture che utilizza una serie di tabelle. Più schemi di Capture abilitano l'esecuzione di più programmi contemporaneamente.

È possibile che si desideri eseguire più programmi Capture nelle seguenti situazioni:

- Per ottimizzare le prestazioni trattando le tabelle a bassa latenza in modo differente da altre tabelle. Se si dispone di tabelle a bassa frequenza, è possibile che si desideri replicare queste tabelle con il loro programma Capture. In questo modo, è possibile assegnare ad esse differenti priorità di runtime. Inoltre, è possibile impostare i parametri del programma Capture, come l'intervallo di eliminazione e l'intervallo di controllo, per adattare la bassa latenza di queste tabelle.
- Per fornire potenzialmente una velocità di trasmissione di Capture più alta. Questo potrebbe rappresentare un significativo beneficio in un ambiente di origine con più CPU. Il trade-off per la velocità di trasmissione più alta è il sovraccarico della CPU aggiuntiva associato con più programmi di lettura della registrazione.

Se si desidera replicare da più database di origine non-DB2 senza lo stesso database federato, è necessario creare più serie di tabelle di controllo Capture, in cui ogni serie disponga di un proprio schema. Oppure, se si preferisce, è possibile utilizzare database federati separati, in questo caso le tabelle di controllo Capture su ogni server possono utilizzare lo schema ASN predefinito schema.

**z/OS** È possibile utilizzare più schemi di Capture se si desidera lavorare con schemi di codifica UNICODE e EBCDIC separatamente oppure se si desidera eseguire più di un'istanza del programma Capture su un sottosistema.

**System i** Utilizzare il comando CRTDPRTBL (Create DPR Tables) per creare un'ulteriore serie di tabelle di controllo Capture utilizzando il parametro CAPCTLLIB per specificare il nome dello schema.

## Creazione delle tabelle di controllo in un database a partizione multipla

**Linux UNIX Windows**

Quando si creano tabelle di controllo Capture in un database a partizione multipla, le tabelle di controllo devono essere posizionate in uno spazio tabella con un'unica partizione nella partizione del catalogo.

Generalmente, si crea innanzitutto uno spazio tabella con un'unica partizione quindi quando si utilizza il Centro di replica o il programma della riga di comando ASNCLP per creare le tabelle di controllo si specifica quello spazio tabella.

Se si sta avviando un programma Capture per la prima volta e si seleziona la modalità di avvio WARMSI, la tabella IBMSNAP\_PARTITIONINFO non esiste. Il programma Capture crea questa tabella e un indice univoco per essa nel tablespace nel quale è ubicata la tabella IBMSNAP\_RESTART. Dopo la creazione della tabella IBMSNAP\_PARTITIONINFO, il programma Capture inserisce una riga per ogni partizione di database all'interno di essa.

Se non è la prima volta che si avvia il programma Capture e si seleziona una delle modalità di avvio a caldo, la tabella IBMSNAP\_PARTITIONINFO già esiste. Nel Centro di replica, se si seleziona la casella di controllo **una o più partizioni sono state aggiunte dall'ultimo di Capture**, il programma Capture inserisce una riga all'interno della tabella IBMSNAP\_PARTITIONINFO per ogni partizione database che è stata aggiunta dall'ultima esecuzione del programma Capture.

---

## Impostazione dei programmi di replica

Prima di poter effettuare la replica, è necessario impostare il programma Capture, il programma Apply, ed altri programmi di replica per i server nel proprio ambiente.

I seguenti argomenti descrivono l'impostazione richiesta per i programmi di replica.

## Impostazione dei programmi di replica (Linux, UNIX, Windows)

**Linux UNIX Windows**

Per impostare la replica dei programmi è necessario impostare le variabili d'ambiente, preparare il database per il programma Capture e facoltativamente effettuare il bind dei pacchetti.

## Impostazione delle variabili d'ambiente per i programmi di replica (Linux, UNIX, Windows)

Linux UNIX Windows

È necessario impostare le variabili d'ambiente prima di avviare o arrestare il programma Capture, il programma Apply, o il programma Replication Alert Monitor e prima di utilizzare il Centro di replica o i comandi di sistema di replica.

### Procedure

Per impostare le variabili di ambiente:

1. Impostare la variabile d'ambiente per il nome dell'istanza DB2 (DB2INSTANCE) come mostrato:

Sistema operativo	Comando
Linux UNIX	<code>export DB2INSTANCE=<i>nome istanza db2</i></code>
Windows	<code>SET DB2INSTANCE=<i>nome istanza db2</i></code>

2. Se il database di origine è stato creato con una tabella codici differente dal valore di tabelle codici predefinito, impostare la variabile di ambiente DB2CODEPAGE su tale tabella codici. **Nota:** È necessario che il programma Capture sia in esecuzione nella stessa tabella codici del database per il quale si stanno acquisendo i dati. DB2 ottiene la tabella codici di Capture dall'ambiente attivo dove Capture è in esecuzione. Se la DB2CODEPAGE non è impostata, DB2 ottiene il valore della tabella codici dal sistema operativo. Il valore ottenuto dal sistema operativo è corretto per Capture se quando è stato creato il database è stato utilizzato la tabella codici predefinita.
3. Opzionale: Impostare la variabile d'ambiente DB2DBDFT nel server di origine.
4. **Linux UNIX** Assicurarsi che le variabili di sistema percorso della libreria e percorso di esecuzione specifiche per il proprio sistema includano la directory dove sono installati le librerie di replica e i file eseguibili.

## Preparazione del database DB2 per l'avvio del programma Capture (Linux, UNIX, Windows)

Linux UNIX Windows

Per preparare il database DB2 per l'avvio del programma Capture, abilitare la registrazione archivio. È anche possibile impostare altri parametri di configurazione database.

### Procedure

Per preparare il database DB2 per l'avvio del programma Capture:

1. Collegarsi al server database di controllo Capture immettendo il seguente comando:

```
db2 connect to database
```

Dove *database* è il server database di controllo Capture.

2. Abilitare la registrazione archivio e preparare il database del server di controllo Capture per il recupero roll-forward immettendo il comando update database configuration (LOGARCHMETH1 = LOGRETAIN) e il comando backup database.

Per più ambienti di partizione database, è necessario che ogni partizione sia impostata per permettere il recupero roll-forward per tutte le partizioni dalle quali Capture acquisirà le modifiche.

3. Potrebbe essere necessario aumentare i valori di configurazione basati sui propri requisiti di installazione.
  - Per le transazioni con un grande numero di righe o con righe molto lunghe è necessario aumentare il valore del parametro **memory\_limit** del Capture.
  - I seguenti valori di configurazione database sono adeguati per molti scenari di stazioni di lavoro di grandi dimensioni: APPLHEAPSZ 1000, LOGFILSZ 4000, LOGPRIMARY 8, LOGSECOND 40, DBHEAP 1000, LOGBUFSZ 16, MAXAPPLS 200.

## Facoltativo: esecuzione del bind dei pacchetti del programma Capture (Linux, UNIX, Windows)

Linux UNIX Windows

Il bind del programma Capture è effettuato automaticamente Linux, UNIX, e Windows durante l'esecuzione. È possibile eseguire il bind dei pacchetti in modo manuale se si desidera specificare le opzioni di bind, pianificare il bind o controllare che tutti i processi di bind siano completati correttamente.

### Procedure

Per effettuare il bind dei pacchetti del programma Capture:

1. Collegarsi al server database di controllo Capture immettendo il seguente comando:

```
db2 connect to database
```

Dove *database* è il server database di controllo Capture.

2. Passare alla directory dove sono ubicati i file di bind del programma Capture.

Linux UNIX

```
db2homedir/sqllib/bnd
```

Dove *db2homedir* è la directory principale dell'istanza DB2.

Windows

```
unità:...\sqllib\bnd
```

Dove *unità*: è l'unità in cui DB2 è installato.

3. Creare ed effettuare il bind del programma Capture per il server database di origine immettendo il seguente comando:

```
db2 bind @capture.lst isolation ur blocking all
```

Dove *ur* specifica l'elenco in formato UR (uncommitted read) per una migliore prestazione.

Questi comandi creano i pacchetti, i cui nomi sono contenuti nel file capture.lst.

## Facoltativo: esecuzione del bind dei pacchetti del programma Apply (Linux, UNIX, Windows)

Linux UNIX Windows

Il bind del programma Apply è effettuato automaticamente su Linux, UNIX e Windows durante l'esecuzione. È possibile eseguire il bind dei pacchetti in modo manuale se si desidera specificare le opzioni di bind, pianificare il bind o controllare che tutti i processi di bind siano completati correttamente.

### Procedure

Per eseguire il bind dei pacchetti del programma Apply:

1. Passare alla directory dove sono ubicati i file di bind del programma Apply.

Linux UNIX

```
db2homedir/sqllib/bnd
```

Dove *db2homedir* è la directory principale dell'istanza DB2.

Windows

```
unità:\...\sqllib\bnd
```

Dove *unità*: è l'unità in cui DB2 è installato.

2. Per ogni server di origine, server di destinazione, server di controllo Capture e server di controllo Apply al quale il programma Apply si connette, effettuare le seguenti operazioni:

- a. Collegarsi al database immettendo il comando seguente:

```
db2 connect to database
```

Dove *database* è il nome del server di origine, del server di destinazione, o del server di controllo Capture, o del server di controllo Apply. Se il database è catalogato come database remoto, è necessario specificare un ID utente e una password nel comando db2 connect to. Ad esempio:

```
db2 connect to database user ID utente using password
```

3. Creare ed eseguire il bind del pacchetto del programma Apply per il database immettendo i seguenti comandi:

```
db2 bind @applycs.lst isolation cs blocking all grant public
```

```
db2 bind @applyur.lst isolation ur blocking all grant public
```

Dove *cs* specifica l'elenco in formato CS (cursor stability) e *ur* specifica l'elenco in formato UR (uncommitted read).

Questi comandi creano i pacchetti i cui nomi sono contenuti nei file *applycs.lst* e *applyur.lst*.

## Creating SQL packages to use with remote systems (System i)

System i

You need to create packages using the CRTSQLPKG command in some cases on System i.

### Informazioni su questa attività

Use this command to create packages in the following cases:

- When you use remote journaling. Run the CRTSQLPKG command on the system where the Capture program is running and point to the system where the source table is located.
- Before you use the ADDDPRESUB or ADDDPRESUBM command to add a subscription set or subscription set member. Run the CRTSQLPKG command on the target server and use the following guidelines:
  - If the source table is on a different machine, point to the system where the source table is located.
  - If the Apply control server is on a different machine, point to the Apply control server.

The SQL packages allow replication programs to operate in a distributed replication environment, whether that environment is one in which you are replicating between System i systems or between a System i system and some other operating system (such as Linux, UNIX, or Windows).

For information about using the CRTSQLPKG command, see *DB2 for i5/OS SQL Programming*.

The packages are created by using the ASN qualifier. On System i they are created in the ASN library. On other operating systems, they are created in the ASN schema.

## Creating SQL packages for the Apply program (System i)

System i

You must create SQL packages for the Apply program on System i so it can interact with all the remote servers to which it needs to connect.

### Procedure

To create SQL packages for the Apply program:

Run this command on the system where Apply is running to enable it to connect to a remote system:

```
CRTSQLPKG PGM(QDP4/QZSNAPV2) RDB(remote_system)
```

Where *remote\_system* is the relational database entry name for the remote system to which the Apply program needs to connect.

## Creating SQL packages for the Replication Analyzer (System i)

System i

You must create SQL packages for the Replication Analyzer on System i so it can interact with the servers that you are analyzing, such as the Capture control server or the target server.

### Procedure

To create SQL packages for the Replication Analyzer:

Run this command on the system where the Replication Analyzer is running:

```
CRTSQLPKG PGM(QDP4/QZSNANZR) RDB(remote_system)
```

Where *remote\_system* is the name of the system that you are analyzing.



## Granting privileges to the SQL packages (System i)

System i

After you create SQL packages on System i, you must grant \*EXECUTE privileges to all users who will be subscribing to files registered on the source database.

### Procedure

To grant privileges for SQL packages:

Log on to the System i system where the source database resides and use one of the following methods:

Method	Description
<b>GRTOBJAUT command</b>	Use the Grant Object Authority (GRTOBJAUT) command, for example: <code>GRTOBJAUT OBJ(<i>ASN/package_name</i>) OBJTYPE(*SQLPKG) USER(<i>subscriber_name</i>) AUT(*OBJOPR *EXECUTE)</code>
<b>GRANT SQL statement</b>	Use SQL to connect to the source database and run the GRANT SQL statement: <code>CONNECT TO <i>data_server_RDB_name</i> GRANT EXECUTE ON PACKAGE <i>ASN/package_name</i> TO <i>subscriber_name</i></code>
<b>GRTDPRAUT command</b>	Use the GRTDPRAUT command, if it is installed on the local system.

## Impostazione dei programmi di replica (z/OS)

z/OS

È necessario impostare e personalizzare i programmi di replica, quando si installa la replica SQL su z/OS.

Consultare le istruzioni in Replication installation and customization for z/OS.

## Acquisizione di più partizioni del database

Linux UNIX Windows

Se si sta effettuando una replica dei dati su DB2 Enterprise Server Edition, è possibile acquisire le modifiche alle tabelle origine che sono sparse in più partizioni del database.

Il programma Capture conserva un elenco delle partizioni del database appartenenti al proprio gruppo partizione nella tabella IBMSNAP\_PARTITIONINFO. Questa tabella è creata dal programma Capture quando esso viene avviato per la prima volta e rileva l'esistenza di più di una partizione database nel proprio gruppo partizione.

Nel momento in cui il programma Capture è avviato a caldo, Capture legge l'elenco delle partizioni del database per il gruppo partizione nel quale sono ubicate le proprie tabelle di controllo. Capture confronta il numero delle partizioni del database note a DB2 con il numero di partizioni del database elencate nella tabella IBMSNAP\_PARTITIONINFO. Il numero delle partizioni elencate nella tabella IBMSNAP\_PARTITIONINFO deve corrispondere al numero conosciuto da DB2 oppure il programma Capture non verrà eseguito.

Se sono state aggiunte una o più partizioni del database dall'ultima volta che è stato avviato il programma Capture, è necessario indicare al programma Capture le nuove partizioni del database. È possibile eseguire ciò nel Centro di replica, selezionando la casella di controllo **Sono state aggiunte una o più partizioni dall'ultimo avvio di Capture** quando si imposta il parametro **startmode** a qualsiasi modalità di avvio a caldo nella finestra Avvia acquisizione.

## Replica di tabelle partizionate

La replica SQL supporta le tabelle di origine che sono partizionate in base all'intervallo (mediante la clausola PARTITION BY dell'istruzione CREATE TABLE).

Le tabelle partizionate utilizzano uno schema di organizzazione dei dati in cui i dati delle tabelle vengono suddivisi in più oggetti di archiviazione, denominati intervalli o partizioni di dati, a seconda dei valori di una o più colonne della chiave di partizione della tabella.

La replica SQL gestisce tutte le partizioni dati di una tabella di origine come se fosse un'unica tabella. Ad esempio, quando si registra una tabella partizionata, si specifica l'intera tabella invece che una o più partizioni dati della stessa. Tutte le operazioni di riga della tabella vengono replicate a prescindere dalla partizione in cui avvengono.

È possibile eseguire varie modifiche su una tabella partizionata, incluso aggiungere una partizione, collegare una partizione o scollegare una partizione. Queste operazioni ALTER effettuate sulla tabella di origine non vengono replicate sulla tabella di destinazione. Per mantenere lo schema di partizione, è necessario modificare la tabella di destinazione indipendentemente dalla tabella di origine.

La replica SQL gestisce queste operazioni ALTER in maniera differente:

**ADD** Aggiunge una nuova partizione vuota alla tabella di origine. Se si desidera aggiungere la nuova partizione alla tabella di destinazione, occorre procedere manualmente.

### ATTACH

Crea una nuova partizione nell'origine utilizzando una tabella esistente. L'operazione ATTACH non viene replicata e i dati contenuti nella nuova partizione non vengono replicati nella tabella di destinazione. Se si necessita della nuova partizione nella tabella di destinazione, occorre aggiungerla manualmente. Se si necessita dei dati collegati nella tabella di destinazione, occorre caricarli in tale tabella manualmente.

### DETACH

Trasforma una partizione esistente in una tabella distinta. L'operazione DETACH non viene replicata. I dati eliminati dalla tabella di origine dall'operazione DETACH non vengono eliminati dalla tabella di destinazione. Se si desidera modificare la partizione di destinazione in una tabella distinta, occorre procedere manualmente.

## Running DB2 Query Patroller in a SQL replication environment

If you set up replication in an environment that also runs DB2 Query Patroller, you need to take special steps to ensure that replication activities are not compromised.

**Informazioni su questa attività**

Query Patroller monitors the cost of dynamic queries that are issued against a database. If you run the Query Patroller client and the cost of a query exceeds a threshold set by the database administrator, the query is trapped and a message issued.

The replication components can issue many dynamic queries. If Query Patroller is enabled and the client is installed where the replication components are running, these situations might occur:

- Dialog messages are issued on the client system when a dynamic query from replication exceeds the defined threshold.
- If Query Patroller detects an error, the replication components can receive a nonzero SQLCODE from Query Patroller. Most of these SQLCODEs are in the range between SQL29000N and SQL29999N.

### Procedure

To avoid these situations, take one of the following actions:

- Run the replication components from a DB2 client that does not have the Query Patroller client enabled.
- Disable Query Patroller. For example, you can disable it for a given database by setting the DYN\_QUERY\_MGMT parameter to 0 (DISABLE). DYN\_QUERY\_MGMT is a database configuration parameter that determines whether Query Patroller is enabled for a given database.

## Setting up journals (System i)

### System i

DB2 DataPropagator for System i uses the information that it receives from the journals about changes to the data to populate the CD and UOW tables for replication.

DB2 DataPropagator for System i runs under commitment control for most operations and therefore requires journaling on the control tables. (The QSQJRN journal is created when the CRTDPRTBL command creates a collection.)

Administrators must make sure the libraries containing the source table, CD table, and target table contain journals. They must also ensure that all the source tables are journaled correctly.

Before you register a table for replication on System i, the table must be journaled for both before-images and after-images.

The following topics describe the journal setup required for replication.

### Setting up journals for source tables (System i)

#### System i

To set up journaling for a source table, you create a journal receiver, create a journal, and then start journaling.

#### Prima di iniziare

You must have authority to create journals and journal receivers for the source tables to be defined.

## Restrizioni

Use a different journal for the source tables than one of those created by DB2 DataPropagator for System *i* in the library for the ASN schema (or other Capture schema).

## Procedure

To create a source table journal:

1. Create a journal receiver in a library of your choice by using the Create Journal Receiver (CRTJRNRCV) command. Place the journal receiver in a library that is saved regularly. Choose a journal receiver name that can be used to create a naming convention for future journal receivers, such as RCV0001. You can use the \*GEN option to continue the naming convention when you change journal receivers. This type of naming convention is also useful if you choose to let the system manage the changing of your journal receivers. The following example uses a library named JRNLIB for journal receivers.

```
CRTJRNRCV  JRNRCV(JRNLIB/RCV0001)
           THRESHOLD(100000)
           TEXT('DataPropagator Journal Receiver')
```

2. Create the journal by using the Create Journal (CRTJRN) command, as in the following example:

```
CRTJRN  JRN(JRNLIB/DJRN1)
        JRNRCV(JRNLIB/RCV0001)
        MNGRCV(*SYSTEM) DLTRCV(*YES)
        TEXT('DataPropagator Journal')
```

- Specify the name of the journal receiver that you created in Step 1.
- Use the Manage receiver (MNGRCV) parameter to have the system change the journal receiver and attach a new one when the attached receiver becomes too large. If you choose this option, you do not need to use the CRTJRN command to detach receivers and create and attach new receivers manually.
- Use the default attribute MINENTDTA(\*NONE). Other values are not valid for this keyword.
- Specify DLTRCV(\*NO) only if you have overriding reasons to do so (for example, if you need to save these journal receivers for recovery reasons). If you specify DLTRCV(\*YES), these receivers might be deleted before you have a chance to save them.

You can use two values on the RCVSIZOPT parameter of the CRTJRN command (\*RMVINTENT and \*MINFIXLEN) to optimize your storage availability and system performance. See the *System i Programming: Performance Tools Guide* for more information.

3. Start journaling the source table by using the Start Journal Physical File (STRJRNPf) command, as in the following example:

```
STRJRNPf FILE(library/file)
         JRN(JRNLIB/DJRN1)
         OMTJRNE(*OPNCLO)
         IMAGES(*BOTH)
```

Specify the name of the journal that you created in Step 2. The Capture program requires a value of \*BOTH for the IMAGES parameter.

4. Change the source table journaling setup:
  - a. Use IMAGES(\*BOTH) to make sure that the source table is journaled for both before- and after-images.

- b. Make sure that the journal has the following attributes: MNGRCV(\*SYSTEM) and DLTRCV(\*YES).
- c. Make sure that the journal has the MINENTDTA(\*NONE) attribute.
- d. For journals on remote systems, specify the MNGRCV(\*SYSTEM), DLTRCV(\*YES), and MINENTDTA(\*NONE) attributes on the source journal. To define the remote journal, specify the DLTRCV(\*YES) attribute on the ADDRMTJRN command.

## Managing journals and journal receivers (System i)

System i

The Capture program uses the Receive Journal Entry (RCVJRNE) command to receive journals.

The following topics describe how to manage journals and journal receivers.

### Specifying system management of journal receivers (System i):

System i

You should let the System i system manage the changing of journal receivers. This is called *system change journal management*.

#### Restrizioni

When you use the RTVJRNE command to retrieve journal entries, no more than 299 source physical files can use the same journal and Capture schema. If you need to register more than 299 files in the same journal, break your source registrations into multiple Capture schemas.

#### Procedure

To specify system management of journal receivers, specify MNGRCV(\*SYSTEM) when you create the journal, or change the journal to that value. If you use system change journal management support, you must create a journal receiver that specifies the threshold at which you want the system to change journal receivers. The threshold must be at least 5 000 KB, and should be based on the number of transactions on your system. The system automatically detaches the receiver when it reaches the threshold size and creates and attaches a new journal receiver if it can.

### Remote journaling with different system times (System i):

System i

If the system time (QTIME) of the source and target systems do not match in a replication environment that uses remote journaling, you need to take precautions in the initial handshake between the Capture and Apply programs.

When replicating with remote journals, the Capture and Apply programs operate as if the source system is local when it is not. If possible, make sure that the QTIME of the source and target systems match.

If you cannot match system times, take the following precautions:

- If the source system time is ahead of the target system time, make sure that PTF SE23500/SI21622 is installed.

- If the source system time is behind the target system time, the Capture program starts receiving changes based on the target system time. Wait for the source system time to become greater than the time of the full refresh. Then make dummy changes and check to see if the changes are replicated before you allow applications to update the source table.

In general, when the source and target system times are different, avoid operations that will cause a full refresh again, such as CLRPFM and CPYF with \*REPLACE.

After the Capture program has started processing changes for the source table, you could set the column DISABLE\_REFRESH in the IBMSNAP\_REGISTER table to 1 for that table. If a full refresh is needed, the Apply program fails and you could coordinate the full refresh when you are ready to perform the handshake in a controlled manner by setting DISABLE\_REFRESH to 0.

### Changing definitions of work management objects (System i):

System i

You can alter the default definitions for the three types of work management objects that are created during installation of DB2 DataPropagator for System i, or provide your own definitions.

#### Informazioni su questa attività

The installation program creates an SQL journal, an SQL journal receiver for this library, and work management objects.

Tabella 2 lists the objects that are created.

Tabella 2. Work management objects

Description	Object type	Name
Subsystem description	*SBSD	QDP4/QZSNDPR
Job queue	*JOBQ	QDP4/QZSNDPR
Job description	*JOBDD	QDP4/QZSNDPR

If you create your own subsystem description, you must name the subsystem QZSNDPR and create it in a library other than QDP4. See *System i Work Management Guide* (SC41-5306) for more information about changing these definitions.

### Specifying user management of journal receivers (System i):

System i

If you specify MNGRCV(\*USER) when you create the journal (meaning you want to manage changing your own journal receivers), a message is sent to the journal's message queue when the journal receiver reaches a storage threshold, if one was specified for the receiver.

#### Informazioni su questa attività

Use the CHGJRN command to detach the old journal receiver and attach a new one. This command prevents Entry not journaled error conditions and limits the amount of storage space that the journal uses. To avoid affecting performance, do this at a time when the system is not at maximum use.

You can switch journal receiver management back to the system by specifying CHGJRN MNGRCV(\*SYSTEM).

You should regularly detach the current journal receiver and attach a new one for two reasons:

- Analyzing journal entries is easier if each journal receiver contains the entries for a specific, manageable time period.
- Large journal receivers can affect system performance and take up valuable space on auxiliary storage.

The default message queue for a journal is QSYSOPR. If you have a large volume of messages in the QSYSOPR message queue, you might want to associate a different message queue, such as DPRUSRMSG, with the journal. You can use a message handling program to monitor the DPRUSRMSG message queue. For an explanation of messages that can be sent to the journal message queue, see *System i Backup and Recovery*.

### Delete journal receiver exit routine (System i): System i

The *delete journal receiver* exit routine (DLTJRNRCV) helps ensure that journal receivers are not deleted if the Capture program has not processed all the entries they contain.

When you install DB2 DataPropagator for System i, this exit routine is registered automatically. It is called any time a journal receiver is deleted, whether or not it is used for journaling the source tables. This exit routine determines whether or not a journal receiver can be deleted.

To take advantage of the delete journal receiver exit routine and leave journal management to the system, specify DLTRCV(\*YES) and MNGRCV(\*SYSTEM) on the CHGJRN or CRTJRN command.

**Attenzione:** If you remove the registration for the delete journal receiver exit routine, you must change all the journals used for source tables to have the DLTRCV(\*NO) attribute.

If the journal that is associated with the receiver is not associated with any of the source tables, this exit routine *approves* the deletion of the receiver.

If the journal receiver is used by one or more source tables, this exit routine makes sure that the receiver being deleted does not contain entries that have not been processed by the Capture program. The exit routine *disapproves* the deletion of the receiver if the Capture program still needs to process entries on that receiver.

If you must delete a journal receiver and the delete journal receiver exit routine does not approve the deletion, specify DLTJRNRCV DLTOPT(\*IGNEXITPGM) to override the exit routine.





---

## Capitolo 4. Registrazione di tabelle e viste come origini di repliche SQL

Con la replica SQL, si identificano le tabelle e le viste che si desidera utilizzare come origini di replica eseguendone la registrazione.

Quando si registra una determinata tabella o vista per la replica, si crea un'origine di dati disponibili che sarà possibile utilizzare in seguito con diverse destinazioni per vari scopi. Le attività di gestione di questa sezione risultano utili nell'impostazione delle informazioni di controllo che definiscono la modalità di cattura dei dati da ciascuna origine in base agli obiettivi di replica.

Quando si registra un'origine, si identifica la tabella o la vista che si desidera utilizzare come origine di replica, le colonne di tabella che si desidera rendere disponibili per la replica e le proprietà relative a come la replica SQL cattura i dati e le modifiche dall'origine.

Relativamente a repliche SQL, è possibile registrare i seguenti oggetti come origini:

- Una tabella DB2
- Una tabella relazionale non DB2 attraverso un nickname
- Una serie secondaria dei dati in una tabella (relazionale DB2 o non DB2)
- Una vista su una singola tabella (DB2)
- Una vista che rappresenta un collegamento interno di due o più tabelle (DB2)

---

### Registrazione di tabelle DB2 come origini

Quando si registra una tabella DB2 come un'origine di replica, si specifica il server di origine, il nome della tabella di origine e lo schema Capture. Viene creata una tabella CD (Change-Data).

#### Prima di iniziare

- Per tutte le origini DB2 tranne per System i, la tabella di origine DDL richiede l'opzione DATA CAPTURE CHANGES. Non eliminare questa opzione dall'origine in uso.
- Le tabelle di controllo Capture devono esistere già sul server di controllo Capture che elaborerà la tabella che si desidera registrare come un'origine.

#### Restrizioni

##### System i

- Poiché le istruzioni SQL sono limitate ad una lunghezza di 32.000 caratteri, è possibile registrare solo circa 2.000 colonne per tabella; il numero esatto di colonne dipende dalla lunghezza dei nomi di colonna.
- Relativamente ad un solo schema Capture, non registrare più di 300 tabelle di origine che utilizzano lo stesso giornale.
- Le tabelle di origine, le tabelle CD e i giornali per le tabelle di origine devono essere tutti nello stesso ASP (Auxiliary Storage Pool) come le tabelle di controllo Capture che contengono le informazioni di registrazione per tali tabelle di origine.

#### Linux UNIX Windows

- La replica è supportata da database a partizione multipla. Non ci sono limiti al numero di partizioni supportate dalla replica.

### Informazioni su questa attività

La replica SQL supporta i seguenti tipi di tabelle DB2 come origini:

#### z/OS

- Tabelle DB2 gestite dall'applicazione in uso
- Tabelle di catalogo
- Tabelle CCD esterne

#### System i

- Tabelle DB2 gestite dall'applicazione in uso (registrate localmente o in remoto)
- Tabelle CCD esterne

#### Linux UNIX Windows

- Tabelle DB2 gestite dall'applicazione in uso
- Tabelle di catalogo (per replica solo di aggiornamento completo)
- Tabelle di interrogazione materializzate
- Tabelle CCD esterne
- Tabelle partizionate con la clausola DISTRIBUTE BY dell'istruzione CREATE TABLE
- Le tabelle partizionate in base all'intervallo (utilizzando la clausola PARTITION BY dell'istruzione CREATE TABLE).
- Tabelle compresse

È possibile registrare la stessa tabella più volte utilizzando differenti schemi Capture.

### Procedure

Per registrare una tabella DB2, utilizzare uno dei metodi riportati di seguito:

Metodo	Descrizione
<b>Programma della riga comandi ASNCLP</b>	Utilizzare il comando CREATE REGISTRATION per registrare una tabella di origine, vista o nickname. Ad esempio, i seguenti comandi impostano l'ambiente e registrano la tabella STAFF nel database DB2 SAMPLE. <pre>SET SERVER CAPTURE TO DB SAMPLE; SET OUTPUT CAPTURE SCRIPT "register.sql"; SET LOG "register.err"; SET RUN SCRIPT LATER; CREATE REGISTRATION (DB2ADMIN.STAFF) DIFFERENTIALREFRESH STAGE CDSTAFF;</pre>

Metodo	Descrizione
Centro di replica	Utilizzare la finestra Registra tabelle. Nella struttura ad albero dell'oggetto, espandere lo schema Capture scelto, fare clic con il tasto destro del mouse sulla cartella <b>Tabelle registrate</b> e fare clic su <b>Registra tabelle</b> . <b>Suggerimento:</b> per risparmiare tempo quando si registra, è possibile impostare in anticipo un profilo dell'oggetto di origine per il server di controllo Capture. Quando si registra una tabella, il Centro di replica utilizza quindi i valori predefiniti definiti nel profilo dell'oggetto di origine anziché i valori predefiniti del Centro di replica. In questo modo è possibile risparmiare tempo quando si registra in quanto è possibile sovrascrivere i valori predefiniti una volta anziché dover selezionare una tabella alla volta e cambiare manualmente le impostazioni predefinite.
<div style="background-color: #800000; color: white; padding: 2px; text-align: center; font-weight: bold;">System i</div> Comando di sistema ADDDPRREG	Utilizzare il comando ADDDPRREG per registrare una tabella su System i.  Ad esempio, per registrare una tabella di origine denominata EMPLOYEE della libreria HR nello schema BSN Capture e per creare una tabella CD denominata CDEMPLOYEE nella libreria HRCDLIB:  ADDDPRREG SRCTBL(HR/EMPLOYEE) CAPCTLLIB(BSN) CDLIB(HRCDLIB) CDNAME(CDEMPLOYEE)

Quando si registra una tabella come un'origine, il programma Capture associato alla tabella registrata legge la registrazione per l'origine e memorizza modifiche da trasferire che si verificano per colonne registrate nella memoria finché la transazione esegue il commit o il rollback. Relativamente ad un rollback, le modifiche vengono eliminate dalla memoria. Relativamente ad un commit, le modifiche sono inserite nella tabella CD non appena il programma Capture legge il record di registrazione di commit. Tali modifiche sono lasciate in memoria finché il programma Capture esegue il commit delle modifiche dopo ciascun ciclo Capture. Il programma Capture non avvia la cattura dei dati per una tabella di origine DB2 finché non viene emesso un segnale CAPSTART, mediante l'utente o il programma Apply.

**Per tabelle di origine non relazionali:** È possibile registrare tabelle DB2 che contengono dati di sistemi di gestione di database non relazionali, come IMS. A tal fine, è necessaria un'applicazione, come IMS DataPropagator o Data Refresher, per popolare una tabella CCD con i dati del database non relazionale. L'applicazione cattura le modifiche sui segmenti non relazionali nel database IMS e popola una tabella CCD. La tabella CCD deve essere completa, ma può essere concentrata o non concentrata. Come altre origini CCD, non esiste alcun programma Capture associato ad una tabella di origine CCD poiché la tabella memorizza già i dati modificati della tabella di origine non relazionale. I prodotti IMS DataPropagator e Data Refresher gestiscono i valori nella tabella IBMSNAP\_REGISTER, il programma Apply può quindi eseguire correttamente la lettura da questa tabella di origine.

---

## Registrazione di tabelle relazionali non DB2 come origini

Quando si registra una tabella relazionale non DB2, si specifica il nickname della tabella di origine che si desidera registrare. Viene creata una tabella CCD (Consistent-Change Data).

### Prima di iniziare

Sul server di controllo Capture che elaborerà questa origine devono essere già presenti tabelle di controllo Capture.

### Restrizioni

- Se si sta utilizzando un singolo database federato per accedere a più server origine relazionali non DB2, si deve utilizzare un diverso schema Capture per ciascun server origine relazionale non DB2 su tale singolo database federato. Non è possibile che due schemi siano identici. È possibile registrare una tabella relazionale non DB2 sotto un solo schema Capture.
- Non è possibile registrare colonne LOB in tabelle relazionali non DB2. Se si registra una tabella che include questo tipo di dati, è necessario registrare una serie secondaria di colonne.

### Informazioni su questa attività

Per impostazione predefinita, il proprietario CCD è derivato dal nome dello schema della tabella di origine. Se si modifica il proprietario CCD in modo che non corrisponde al nome dello schema, accertarsi che il proprietario della tabella di origine sia autorizzato a scrivere sulla tabella CCD. Se il proprietario della tabella di origine non può aggiornare la tabella CCD, i trigger sulla tabella di origine non saranno in grado di scrivere le modifiche sulla tabella CCD.

### Procedure

Per registrare una tabella relazionale non DB2, utilizzare uno dei metodi riportati di seguito:

Metodo	Descrizione
<b>Programma della riga comandi ASNCLP</b>	<p>Utilizzare il comando CREATE REGISTRATION per registrare una tabella di origine, vista o nickname. Ad esempio, i seguenti comandi impostano l'ambiente e creano una registrazione con le seguenti caratteristiche:</p> <ul style="list-style-type: none"> <li>• Server non IBM che contengono il database Oracle V9ORA</li> <li>• Server federato FEDORADB</li> <li>• Tabella CCD nel database Oracle undjr09.ccdtest</li> <li>• Nickname CCD nel server federato repldba.ccdtestnk</li> <li>• Nickname origine che è registrato repldba.tesnk</li> <li>• Tutte le colonne in repldba.tesnk vengono registrate con post-immagini</li> </ul> <pre>SET SERVER CAPTURE TO DB FEDORADB NONIBM SERVER V9ORA ID repldba PASSWORD "passwd"; SET OUTPUT CAPTURE SCRIPT "ora_reg.sql"; SET CAPTURE SCHEMA SOURCE ASNORA; SET LOG "orareg.out"; CREATE REGISTRATION (repldba.testnk) DIFFERENTIALREFRESH STAGE repldba.ccdtestnk CONDENSED OFF NONIBM undjr09.ccdtest COLS ALL IMAGE AFTER;</pre> <p>L'opzione CONDENSED OFF è richiesta per origini federate.</p>

Metodo	Descrizione
Centro di replica	<p>Utilizzare la finestra Registra nickname. Dalla struttura ad albero dell'oggetto, espandere il database relazionale non DB2 che contiene i nickname che si desidera registrare. Fare clic con il tasto destro del mouse sulla cartella <b>Nickname registrato</b> e selezionare <b>Registra nickname</b>.</p> <p><b>Suggerimento:</b> per risparmiare tempo quando si registra, è possibile impostare in anticipo un profilo dell'oggetto di origine per il server di controllo Capture. Quando si registra una tabella, il Centro di replica utilizza quindi i valori predefiniti definiti nel profilo dell'oggetto di origine per le tabelle CCD e i nickname per le tabelle CCD anziché i valori predefiniti del Centro di replica. In questo modo è possibile risparmiare tempo quando si registra in quanto è possibile sovrascrivere i valori predefiniti una volta anziché dover selezionare una tabella alla volta e cambiare manualmente le impostazioni predefinite.</p>

Quando viene modificata una tabella relazionale non DB2 registrata, i trigger Capture simulano il programma Capture e inseriscono la modifica nella tabella CCD. I trigger Capture avviano l'acquisizione delle modifiche per una tabella di origine relazionale non DB2 quando si registra l'origine.

---

## Opzioni di registrazione per tabelle di origine

Le repliche SQL forniscono diverse opzioni quando si registra una tabella come origine di replica. Tali opzioni fanno parte dell'attività più ampia di registrazione di una tabella.

Una volta scelta la tabella che si desidera registrare, è possibile individuare le colonne che si desidera rendere disponibili per la replica ed è possibile definire le proprietà che determinano come verranno gestiti e memorizzati i dati registrati da tale origine. È possibile inoltre specificare altre opzioni di registrazione, ad esempio come si desidera che il programma Capture memorizzi i dati di origine nella tabella CD (o come si desidera che i trigger Capture memorizzino i dati nella tabella CCD).

### Registrazione di una serie secondaria di colonne (serie secondaria verticale)

È possibile registrare una serie secondaria di colonne della tabella di origine per la replica, ad esempio se non si desidera che tutte le colonne disponibili per le destinazioni da richiedere o se le tabelle di destinazione non supportano tutti i tipi di dati che sono definiti per la tabella di origine.

Per impostazione predefinita, tutte le colonne vengono registrate. Per registrare una serie secondaria delle colonne, selezionare solo le colonne che si desidera rendere disponibili per la replica sulla tabella di destinazione.

Poiché le tabelle CD e CCD devono contenere dati chiave sufficienti ad alcuni tipi di tabelle di destinazione (come quelle di un determinato momento), accertarsi che la serie secondaria contenga le colonne che agiranno come colonne chiave (chiave primaria o indice univoco) per la destinazione.

**Suggerimento:** registrare una serie secondaria delle colonne di origine solo se si è certi che non si eseguirà mai la replica delle colonne non registrate. Se successivamente si desidera replicare le colonne che non sono state registrate, è

necessario modificare le registrazioni per aggiungere le colonne non registrate. (Relativamente a origini relazionali non DB2, è necessario ridefinire completamente le registrazioni per aggiungere nuove colonne ad una registrazione). Se si prevede di avere una tabella CCD interna associata a questa origine, può essere anche più difficile aggiungere colonne in seguito poiché la registrazione di nuove colonne le aggiunge alla tabella CD, ma non alla tabella CCD interna. Per evitare tali problemi, si potrebbe voler registrare tutte le colonne ed utilizzare il programma Apply sulla serie secondaria le cui colonne sono replicate sulle destinazioni.

## **Replica di modifica e acquisizione e copia di aggiornamento completo**

Per impostazione predefinita, viene eseguita la replica solo delle modifiche che sono state eseguite nella tabella di origine dall'ultimo ciclo di replica (replica di modifica e acquisizione). È possibile inoltre eseguire la replica di tutti i dati nella tabella di origine durante ogni ciclo (replica solo di aggiornamento completo).

### **Replica di modifica e acquisizione**

Durante la replica di modifica e acquisizione, solo i dati modificati vengono replicati sulla tabella di destinazione. A seconda del tipo di tabella di destinazione che si sceglie per questa origine, è necessario eseguire un caricamento iniziale della tabella. Nella maggior parte dei casi, il programma Apply esegue un iniziale aggiornamento completo e quindi continua con la replica di modifica e acquisizione.

Se si sceglie di non consentire l'aggiornamento completo delle tabelle di destinazione, è necessario caricare nuovamente manualmente la tabella se le tabelle di origine e destinazione devono essere nuovamente sincronizzate. Una volta caricata la destinazione con i dati di origine iniziali, il programma Capture cattura le modifiche apportate sull'origine e le memorizza nella tabella CD. Nella replica di modifica e acquisizione per origini relazionali non DB2, i trigger Capture catturano le modifiche sull'origine e le memorizzano nella tabella CCD. Il programma Apply legge le modifiche dalla tabella CD o CCD e le applica sulle destinazioni che richiedono l'origine registrata.

Quando si definisce una tabella di origine DB2 per la replica di modifica e acquisizione, si potrebbe non voler memorizzare tutte le modifiche che si verificano sull'origine nella tabella CD. È possibile registrare una serie secondaria di righe (orizzontale) che filtra le modifiche in modo che nella tabella CD ne vengono catturate meno di quelle che si verificano effettivamente sull'origine. È possibile eseguire la selezione dalle due seguenti regole di cattura righe per determinare le righe modificate dalla tabella di origine che il programma Capture registra nella tabella CD:

- Cattura delle modifiche su tutte le righe.
- Cattura delle modifiche solo se la modifica si è verificata in una colonna registrata. (solo per DB2)

Per impostazione predefinita, le modifiche vengono catturate quando viene aggiornata una riga di una qualsiasi colonna (registrata o non registrata) sulla tabella di origine. Se si registra solo una serie secondaria delle colonne, il programma Capture registra i valori della riga per le colonne registrate nella tabella CD ogni volta che si verifica una modifica sulla tabella di origine, anche se le colonne modificate sono diverse dalle colonne registrate. Utilizzare questa opzione predefinita se si desidera conservare una cronologia di tutte le modifiche sulla tabella di origine. Questa è l'unica opzione disponibile per origini relazionali

non DB2, i trigger Capture catturano tutte le righe modificate sull'origine, anche se la modifica si verifica in una colonna non registrata.

**Esempio:** si supponga di avere 100 colonne nella tabella e che se ne registrino 50 per la replica. Per impostazione predefinita, ogni volta che viene eseguita una modifica su una delle 100 colonne nella tabella, il programma Capture scriverà una riga sulla tabella CD (o i trigger Capture scriveranno una riga sulla tabella CCD).

Se si ha un'origine DB2, si potrebbe volere che il programma Capture catturi le modifiche solo per le colonne registrate. In tale caso, il programma Capture scrive una riga sulla tabella CD solo quando le modifiche si verificano sulle colonne registrate.

**Suggerimento:** scegliere di catturare le modifiche per tutte le righe se si necessita di informazioni per fini di controllo oppure se le modifiche nella tabella si verificano quasi sempre solo nelle colonne registrate. Scegliere di catturare modifiche solo per le colonne registrate se si verificano di frequente modifiche che interessano solo le colonne non registrate. Utilizzare questa opzione se non si desidera conservare una cronologia di tutte le modifiche sulla tabella di origine.

## Replica solo di aggiornamento completo

Quando le destinazioni richiedono un'origine registrata per la replica solo di aggiornamento completo, il programma Apply elimina tutti i dati dalla tabella di destinazione, copia i dati presenti nelle colonne registrate sull'origine e popola le destinazioni con i dati di origine durante ogni ciclo di replica. Il programma Capture non è interessato e non esiste alcuna tabella CD; il programma Apply legge i dati direttamente dalla tabella di origine.

### Tablette di piccole dimensioni

Si potrebbe voler scegliere la replica solo di aggiornamento completo se si dispone di una tabella di origine di piccole dimensioni che non necessita di molto tempo o risorse da copiare.

### Tablette di grandi dimensioni

Se si dispone di tabelle di dimensioni più grandi e si desidera utilizzare la replica solo di aggiornamento completo, per caricare più velocemente le tabelle si potrebbe voler utilizzare la routine di chiusura ASNLOAD.

**Limitazione:** Se si prevede di disporre di una tabella di destinazione concentrata che richiede questa origine e non è possibile attivare un indice univoco per tale tabella di destinazione, è necessario registrare l'origine per la replica solo di aggiornamento completo.

## Colonne post-immagine e pre-immagine

Quando si registra un'origine per la replica di modifica e acquisizione, per impostazione predefinita viene catturato solo il valore modificato (post-immagine) in una colonna. È possibile anche scegliere di catturare il valore precedente (pre-immagine).

z/OS

Linux UNIX Windows

È possibile selezionare se catturare valori pre-immagine per singole colonne in una tabella.

È possibile selezionare se catturare pre-immagini per tutte le colonne o nessuna di esse in una tabella. Non è possibile selezionare questa opzione per ogni singola colonna.

#### **Sybase o Microsoft® SQL Server**

Una tabella può contenere solo una colonna di tipo TIMESTAMP. Quando l'origine dati è Sybase o Microsoft SQL Server e la tabella di origine ha una colonna di tipo TIMESTAMP, selezionare post-immagini solo per questa colonna quando la si definisce come parte dell'origine della replica.

**Limitazione:** Non è possibile includere valori pre-immagini nella tabella CD per colonne con tipi di dati LOB.

Le sezioni riportate di seguito trattano quando si deve scegliere ciascuna opzione.

### **Cattura soltanto di valori post-immagine**

Per ogni colonna che si registra per la replica di modifica e acquisizione, è possibile scegliere il programma Capture o i trigger per registrare solo il valore post-immagine per ciascuna modifica. Quando si sceglie di catturare solo valori post-immagine, la tabella CD (o CCD) contiene una colonna per ogni valore modificato, che memorizza il valore della colonna di origine dopo che si è verificata la modifica.

Non sono necessarie pre-immagini se si prevede di utilizzare solo tipi di tabelle di destinazione aggregate di modifica e aggregate di base per questa origine. Le colonne pre-immagine non sono influenti se si prevede di utilizzare la tabella di destinazione per valori calcolati in quanto non esistono pre-immagini per le colonne calcolate. Tutti gli altri tipi di tabella di destinazione possono fare uso delle colonne pre-immagine.

### **Cattura di valori pre-immagine e post-immagine**

Per ogni colonna che si registra per la replica di modifica e acquisizione, è possibile scegliere il programma Capture o i trigger per registrare sia il valore pre-immagine che post-immagine per ciascuna modifica. Quando si sceglie di catturare valori pre-immagine e post-immagine, la tabella CD (o CCD) contiene due colonne per ogni valore modificato: uno per il valore nella colonna di origine prima che si è verificata la modifica e uno per il valore dopo che si è verificata la modifica.

Quando si sceglie di memorizzare sia le pre-immagini che le post-immagini nella tabella CD (o CCD), le colonne pre-immagine e post-immagine possiedono valori diversi per azioni differenti eseguite sulle tabelle di origine:

#### **Inserimento**

La colonna pre-immagine contiene un valore NULL. La colonna post-immagine contiene il valore inserito.

#### **Aggiornamento**

La colonna pre-immagine contiene il valore della colonna precedente all'esecuzione della modifica. Il valore post-immagine contiene il valore della colonna successivo all'esecuzione della modifica.

Quando si sceglie che gli aggiornamenti vengano catturati come coppie di eliminazione e inserimento, la riga di eliminazione contiene la pre-immagine dell'aggiornamento sia nelle colonne pre-immagine che



post-immagine della riga e la riga di inserimento contiene valori NULL nella colonna pre-immagine e la post-immagine nella relativa colonna.

### **Eliminazione**

Le colonne pre-immagine e post-immagine contengono il valore della colonna precedente all'esecuzione della modifica.

**z/OS** Per le tabelle in DB2 per z/OS, è possibile utilizzare nomi di colonna di 18 caratteri, ma la replica sostituirà il diciottesimo carattere con l'identificatore della colonna pre-immagine nelle tabelle di destinazione, è necessario quindi accertarsi che i primi 17 caratteri del nome siano univoci.

**Linux UNIX Windows System i** Per le colonne che hanno pre-immagini definite, la replica DB2 limita i nomi di colonna a 29 caratteri perché il nome completo di colonna può avere solo 30 caratteri. Se il nome di colonna è più lungo, come impostazione predefinita la replica tronca gli ulteriori caratteri dalla destra, a meno che il profilo non è stato impostato per eseguire il troncamento dalla sinistra. Poiché la replica aggiunge un identificatore di colonna pre-immagine (di solito X) nelle colonne di destinazione e ogni nome di colonna deve essere univoco, non è possibile utilizzare nomi di colonna più lunghi di 29 caratteri. Relativamente alle tabelle che non si prevede di replicare, è possibile utilizzare nomi di colonna più lunghi, ma considerare l'uso di nomi di 29 caratteri nel caso si desideri replicare successivamente tali colonne.

L'elenco riportato di seguito descrive i casi in cui si potrebbe voler catturare valori pre-immagine:

### **Per conservare una cronologia dei dati di origine**

Se si desidera conservare i dati ai fini di controllo, si potrebbe voler selezionare sia le pre-immagini che le post-immagini in modo che si disponga di una registrazione di come i dati sono cambiati in un periodo di tempo. Una serie di copie di pre-immagini e post-immagini è utile in alcune industrie che richiedono funzioni di rollback dell'applicazione o controllo.

### **Per configurazioni di aggiornamenti con rilevamento di conflitti**

In configurazioni di aggiornamenti in cui sono possibili conflitti tra le tabelle di replica (dove il rilevamento di conflitti è impostato su qualsiasi valore diverso da None), è necessario registrare sia le colonne post-immagine che pre-immagine per la tabella CD delle repliche in modo che, se si verifica un conflitto, può essere eseguito il rollback delle modifiche.

### **Quando le colonne chiave nella destinazione sono soggette ad aggiornamento**

Quando si registra un'origine, considerare le potenziali tabelle di destinazione che si potrebbero definire utilizzando questa tabella come origine. Di solito, le tabelle di destinazione sono concentrate e richiedono una colonna o una serie di colonne che rendono ciascuna riga in tale tabella di destinazione univoca. Queste colonne univoche costituiscono la chiave di destinazione. Se una di queste colonne chiave di destinazione potrebbero essere aggiornate all'origine, la replica SQL richiede una gestione particolare per accertare che vengano aggiornate le righe corrette della tabella di destinazione. Per accertarsi che la replica SQL aggiorni le righe corrette nella tabella di destinazione con il nuovo valore chiave, è possibile scegliere di catturare post-immagini e pre-immagini per le colonne che comporranno la chiave di destinazione. Il programma Apply necessita dei valori pre-immagine per queste colonne registrate quando

applica le modifiche di colonne di origine non chiave alle colonne di destinazione chiave nella tabella di destinazione. Quando si applicano le modifiche, il programma Apply cerca la riga nella tabella di destinazione, ricercando i valori chiave di destinazione che corrispondono al valore pre-immagine nella tabella CD (o CCD) di origine e quindi aggiorna la riga di destinazione con il valore post-immagine nella tabella CD (o CCD) di origine.

Sebbene quando si registra la vista o tabella di origine si registrano tali valori pre-immagine, la replica non sa che l'applicazione in uso aggiornerà la chiave di destinazione. In seguito, quando si definiscono le destinazioni da richiedere a questa origine (creando serie di richieste), è possibile specificare che il programma Apply esegua determinati aggiornamenti quando si applicano modifiche da colonne non chiave all'origine a colonne chiave alla destinazione.

## Prefisso pre-immagine

Se si catturano colonne post-immagine e pre-immagine, la colonna post-immagine assume il nome della colonna nella tabella di origine mentre la colonna pre-immagine assume il nome della colonna nella tabella di origine con un prefisso di un carattere.

Il prefisso pre-immagine predefinito assegnato dal programma della riga comandi ASNCLP e dal Centro di replica è X. Il valore predefinito per i comandi System i è @.

È possibile modificare il prefisso predefinito. La combinazione del prefisso pre-immagine e del nome colonna CD (o CCD) non possono essere uguali ad un nome di colonna potenziale o attuale nella tabella CD (o CCD).

**Esempio:** se si utilizza X come prefisso pre-immagine e si registra una colonna di origine denominata COL, non è possibile registrare una colonna denominata XCOL in quanto non è chiaro se XCOL è un nome di colonna corrente di un'altra colonna di origine o il nome di una colonna pre-immagine con un nome di colonna COL e un prefisso pre-immagine X.

**Limitazione:** Non è possibile utilizzare uno spazio come prefisso pre-immagine.

Se non si sta eseguendo la replica di colonne pre-immagine per una tabella, è possibile non avere un prefisso pre-immagine e impostare questa proprietà su Null.

## Arresto del programma Capture a seguito di errore

Quando il programma Capture rileva alcuni problemi mentre elabora le registrazioni, si arresta per impostazione predefinita. È possibile scegliere di lasciare il programma in esecuzione.

L'elenco riportato di seguito fornisce informazioni dettagliate su come scegliere la migliore opzione per l'ambiente in uso.

### Arresta Capture a seguito di errore


Con questa opzione, il programma Capture scrive un messaggio di errore nella tabella IBMSNAP\_CAPTRACE e termina.

Il programma Capture si arresta quando si verificano i seguenti errori irreversibili:

- Lo spazio della tabella CD è esaurito.

- L'errore SQLCODE-911 si verifica 10 volte in una riga.
- Si verificano errori SQL impreveduti.

Il programma Capture non si arresta quando si verificano alcuni errori non irreversibili; ad esempio:

- SQLCODES indica una lunghezza non valida dei dati.
-  Il dizionario di compressione non esiste.

Quando si verificano quegli errori non irreversibili, il programma Capture invalida le registrazioni e continua l'esecuzione.

### Non arrestare Capture a seguito di errore

Il programma Capture continua l'esecuzione quando si verificano alcuni errori. Se rileva degli errori la prima volta che cerca di elaborare l'origine, non attiva la registrazione. Se l'origine registrata era già attiva, ne arresta l'elaborazione. La registrazione viene arrestata in ogni caso. Una registrazione arrestata ha un valore "S" (Arrestato) nella colonna STATE della tabella di controllo IBMSNAP\_REGISTER.

Questa opzione non arresta il programma Capture quando si verificano i seguenti errori non irreversibili:

- La registrazione non è definita correttamente.
- Il programma Capture non ha individuato la tabella CD quando ha tentato di inserire le righe dei dati modificati.
- L'opzione DATA CAPTURE CHANGES sulla tabella di origine (non-System i) è stata rilevata come disattiva (OFF) quando il programma Capture è stato avviato o reinizializzato.

Se lo stato registrato di un membro della serie di richieste si trova in stato di arresto a causa di un errore, il programma Apply non sarà in grado di elaborare la serie.

## Opzioni relative alla modalità di memorizzazione degli aggiornamenti del programma Capture

Per impostazione predefinita, gli aggiornamenti alla tabella di origine vengono memorizzati in una singola riga nella tabella CD. In alcuni casi, è necessario indicare ai trigger o al programma Capture di catturare gli aggiornamenti come coppie DELETE e INSERT memorizzate in due righe.

È necessario catturare gli aggiornamenti come istruzioni DELETE e INSERT quando le applicazioni di origine aggiornano una o più colonne a cui viene fatto riferimento da un predicato sul membro della serie di richieste.

**Esempio:** si supponga di pianificare la definizione di una destinazione che richiede solo i dati di origine con un predicato basato su un determinato valore di colonna (ad esempio, WHERE DEPT = 'J35'). Quando si cambia tale colonna (ad esempio, DEPT='FFK'), la modifica catturata non verrà selezionata per la replica sulla destinazione perché non corrisponde ai criteri del predicato. Ovvero, non verrà eseguita la replica del nuovo reparto FFK perché il membro della serie di richieste si basa su Reparto J35. La conversione degli aggiornamenti su una copia DELETE e INSERT assicura che la riga della tabella di destinazione venga eliminata.

Ciascun aggiornamento catturato viene convertito su due righe nella tabella CD (o CCD) per tutte le colonne. Si potrebbe aver necessità di modificare l'assegnazione dello spazio per la tabella CD (o CCD) per organizzare questo aumento di dati catturati.

## Impedimento della ricattura delle modifiche (replica di aggiornamenti)

Relativamente alla replica di aggiornamenti, è possibile utilizzare l'opzione di ricattura per controllare se le modifiche di cui è stata eseguita la replica da un sito vengono ricatturate sul secondo sito per la replica su altri siti.

**Limitazione:** le tabelle di database relazionali non DB2 non possono partecipare ad aggiornamenti. Questa opzione è solo per origini DB2.

Nella replica di aggiornamenti, le modifiche possono aver origine nella tabella principale o nelle tabelle di replica associate. Quando si registra una tabella che si prevede di utilizzare nella replica di aggiornamenti, la replica SQL assume che sarà la tabella principale nella configurazione in uso.

Durante la registrazione, si imposta l'opzione di ricattura per la tabella principale. In seguito, quando si esegue l'associazione della tabella di origine principale con le relative destinazioni di replica, è possibile impostare se le modifiche nella replica vengono ricatturate e inoltrate ad altre tabelle.

Quando si registra la tabella di origine che opererà come principale nella configurazione di aggiornamenti, è possibile selezionare una delle due seguenti opzioni:

### **Ricattura modifiche dalla principale**

Gli aggiornamenti alla principale che hanno avuto origine da una replica vengono ricatturati nella principale e inoltrati alle altre repliche.

### **Non ricatturare modifiche dalla principale**

Gli aggiornamenti alla principale che hanno avuto origine da una replica non vengono ricatturati nella principale e inoltrati alle altre repliche.

Quando si registra la tabella di replica nella configurazione di aggiornamenti, è possibile selezionare una delle due seguenti opzioni:

### **Ricattura modifiche dalla replica**

Gli aggiornamenti alla replica che hanno avuto origine dalla principale vengono ricatturati nella replica e inoltrati alle altre repliche che richiedono tale replica.

### **Non ricatturare modifiche dalla replica**

Gli aggiornamenti alla replica che hanno avuto origine dalla principale non vengono ricatturati nella replica e inoltrati alle altre repliche che richiedono tale replica.

L'impedimento della ricattura delle modifiche può aumentare le prestazioni e ridurre i costi di memoria poiché il programma Capture non ricattura le stesse modifiche per ciascuna replica.

I seguenti argomenti trattano come decidere se ricatturare le modifiche in base alla configurazione di aggiornamenti.

## **Tabelle principali con una replica**

Se si prevede di disporre di una sola replica nella configurazione di aggiornamenti, creare la registrazione in modo che le modifiche non vengano ricatturate nella tabella principale o nella tabella di replica.

Questa impostazione è ottimale se la tabella principale non è un'origine di altre tabelle di replica e la replica non è un'origine di altre repliche (in una configurazione a più livelli). Se sono implicate solo due tabelle, una modifica che ha origine nella replica non deve essere ricatturata in quella principale ed eventuali modifiche che hanno origine in quella principale non devono essere ricatturate sulla singola replica.

### Più repliche che sono partizioni reciprocamente esclusive della principale

Relativamente a repliche che sono partizioni reciprocamente esclusive della principale, creare la propria registrazione in modo che le modifiche non vengano ricatturate nella tabella principale o nelle tabelle di replica.

Se si prevede di avere diverse repliche che sono partizioni della tabella principale, si potrebbe voler impedire la ricattura delle modifiche sia sulla replica che su ciascuna replica. Questa impostazione è ottimale se nessuna delle repliche è un'origine di altre tabelle di replica. Quando le repliche sono partizioni della principale, due repliche non richiedono mai gli stessi dati alla principale. Quindi, qualsiasi modifica che ha origine nella replica non deve essere ricatturata nella principale e inoltrata alle altre repliche poiché solo la replica in cui si è verificata la modifica richiede tali dati di origine.

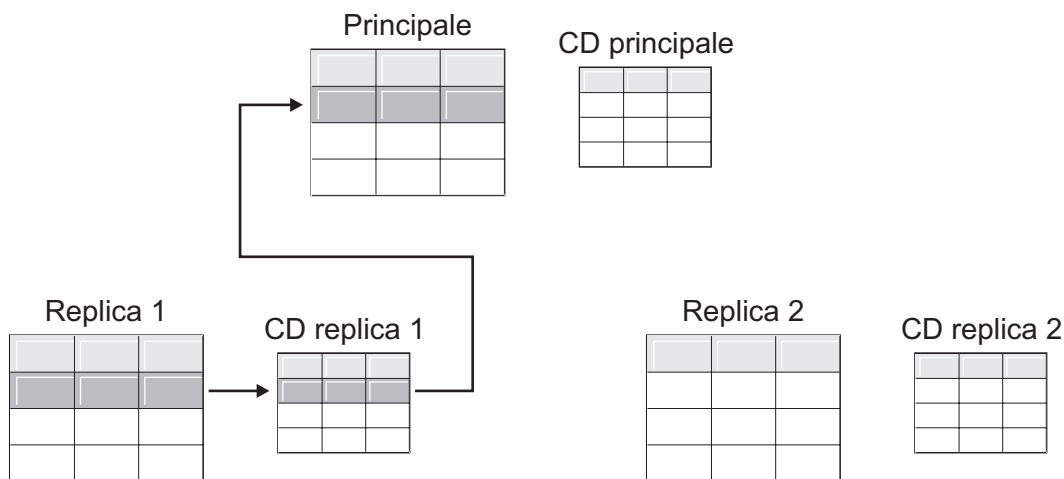


Figura 1. Opzioni di ricattura per repliche che sono partizioni reciprocamente esclusive della principale. Quando si hanno più repliche che non richiedono gli stessi dati nella principale, non è necessario utilizzare l'opzione di ricattura per alcuna delle tabelle.

### Principali che replicano modifiche su più repliche

Relativamente a principali che eseguono la replica di modifiche su più repliche, creare la propria registrazione in modo che le modifiche vengano ricatturate nella tabella principale, ma non nelle tabelle di replica.

Le modifiche che hanno origine su una replica vengono quindi ricatturate sulla principale replicate nelle altre repliche che richiedono i dati principali aggiornati.

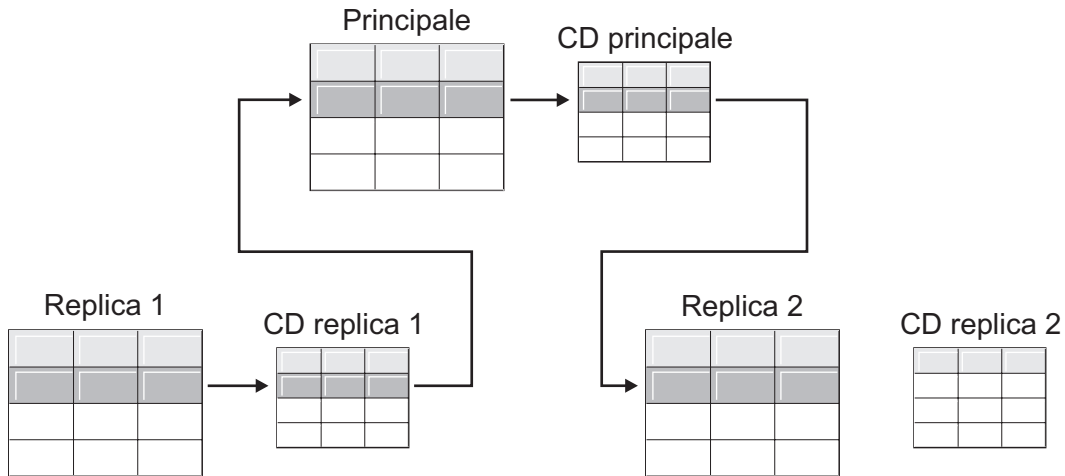
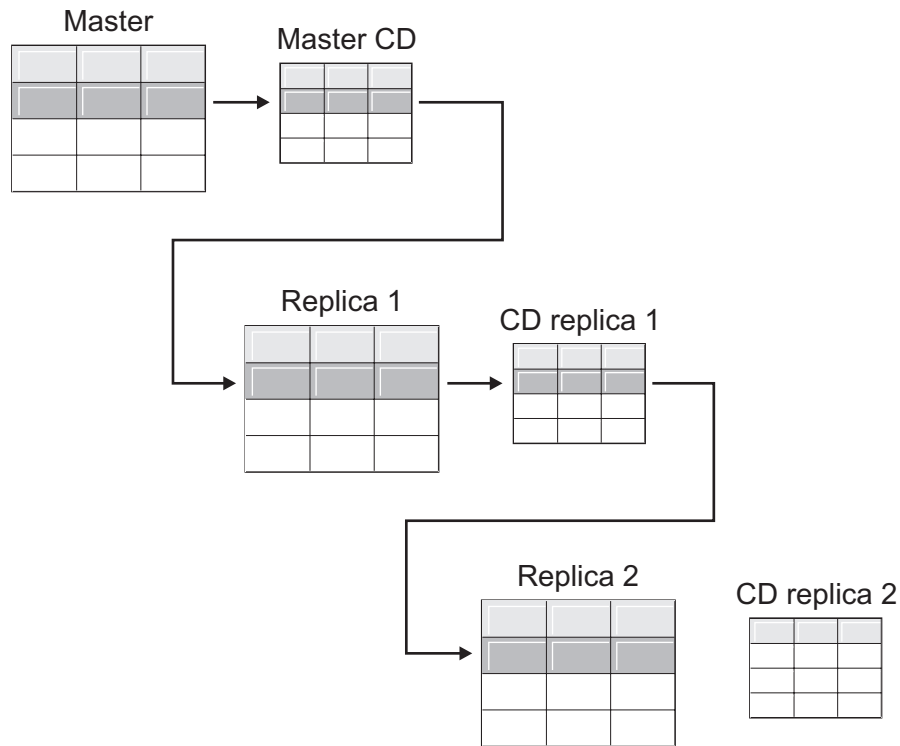


Figura 2. Opzione di ricattura per principali che replicano modifiche su più repliche. Quando si dispone di più repliche che richiedono gli stessi dati nella principale, è possibile utilizzare l'opzione di ricattura sulla principale in modo che le modifiche che si verificano su una replica vengono ricatturate nella principale e inoltrate alle altre tabelle di replica.

### Replica di modifiche su altre repliche (a più livelli)

Relativamente a repliche che eseguono la replica di modifiche su altre repliche (a più livelli), creare la propria registrazione in modo che le modifiche non vengano ricatturate nella tabella principale, ma nelle tabelle di replica.

È possibile avere una configurazione a più livelli in cui la principale (livello 1) opera come un'origine su una replica (livello 2) e quindi anche tale replica opera come un'origine su un'altra replica (livello 3). Se si prevede di avere questo tipo di configurazione, si potrebbe volere che il programma Capture ricatturi le modifiche nella replica intermedia (livello 2) in modo che le modifiche originate sulla principale vengano inoltrate alla replica successiva (livello 3).



*Figura 3. Abilitazione da parte dell'opzione di ricattura a livello 2 alla replica sul livello 3 di modifiche a livello 1. Quando si dispone di una tabella di replica che opera come un livello intermedio in una configurazione a più livelli, è possibile utilizzare l'opzione di ricattura sulla replica in modo che le modifiche che si verificano sulla principale vengono ricatturate nella replica a livello intermedio e inoltrate alla replica nel livello successivo.*

Inoltre, quando la ricattura è impostata per la replica intermedia (livello 2), le modifiche che si originano sulla replica finale (livello 3) vengono ricatturate sulla replica intermedia (livello 2) e inoltrate alla principale (livello 1).

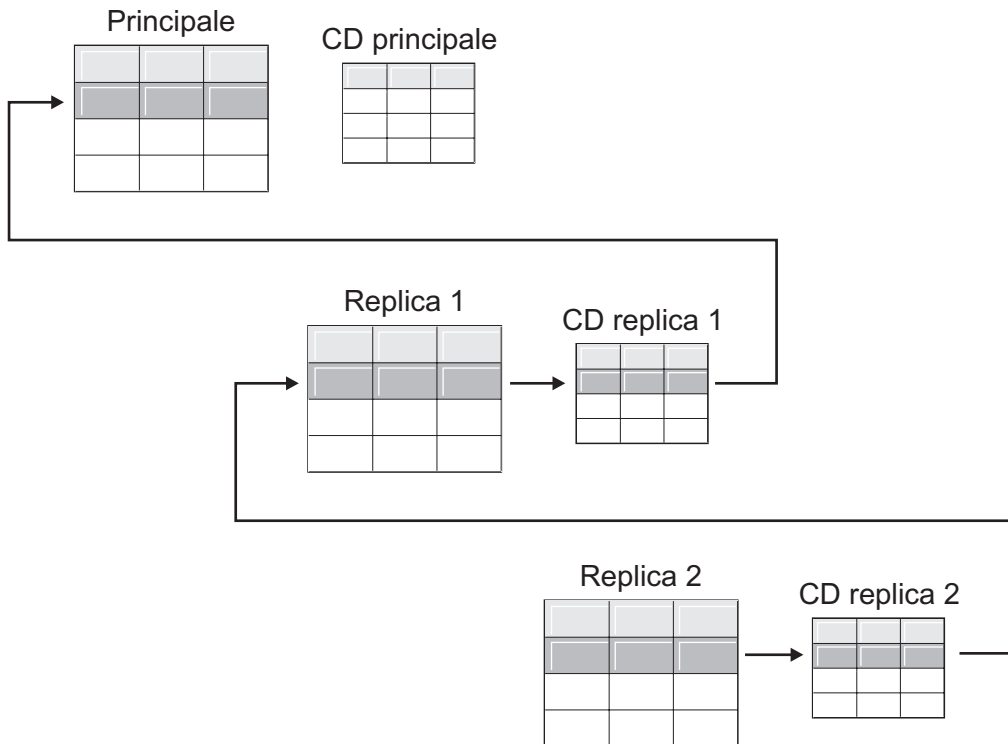


Figura 4. Abilitazione da parte dell'opzione di ricattura a livello 2 alla replica fino al livello 1 di modifiche a livello 3. Quando si dispone di una tabella di replica che opera come un livello intermedio in una configurazione a più livelli, è possibile utilizzare l'opzione di ricattura sulla replica in modo che le modifiche che si verificano sulla replica nel livello successivo vengono ricatturate nella replica nel livello intermedio e inoltrate alla principale.

## Opzioni per la rilevazione di conflitti (replica di aggiornamenti)

In configurazioni di aggiornamenti, possono verificarsi talvolta dei conflitti tra la tabella principale e le relative repliche. Quando si registra un'origine, è possibile selezionare tre livelli di rilevazione di conflitti: nessuno, standard e avanzato.

Possono verificarsi conflitti quando:

- Viene eseguito un aggiornamento su una riga nella tabella principale e viene eseguito un altro aggiornamento sulla stessa riga in una o più tabelle di replica e il programma Apply elabora le modifiche in conflitto durante lo stesso ciclo.
- Violazione dei limiti.

Sebbene si imposti il livello di rilevazione di conflitti per singole origini di replica, il programma Apply utilizza il più elevato livello di rilevazione di conflitti di qualsiasi membro di serie di richieste come livello per tutti i membri della serie.

### Restrizioni:

- Le tabelle di database relazionali non DB2 non possono partecipare ad aggiornamenti; quindi, le origini relazionali non DB2 non dispongono di rilevazione di conflitti.
- Se si dispone di una configurazione di aggiornamenti che include colonne LOB, è necessario specificare Nessuno come livello di rilevazione di conflitti.

In base alla tolleranza di requisiti di prestazioni e transazioni perse o rifiutate, è possibile decidere il tipo di rilevazione da utilizzare:



### **Nessuno**

Nessun rilevamento conflitto. Gli aggiornamenti di conflitto tra la tabella principale e la tabella di replica non verranno rilevati. Questa opzione non è consigliata per repliche di aggiornamenti.

### **Standard**

Rilevamento conflitto minimo.

Durante ogni ciclo Apply, il programma Apply confronta i valori chiave nella tabella CD principale con quelli della tabella CD della replica. Se esiste lo stesso valore chiave in entrambe le tabelle, si verifica un conflitto. In caso di conflitto, il programma Apply annullerà la transazione che aveva eseguito il commit sulla replica dalla tabella CD di replica e conserverà solo le modifiche originate sulla tabella principale.

### **Avanzato**

Rilevamento conflitto che fornisce la miglior integrità dati tra la tabella principale e le repliche.

Come con la rilevazione standard, durante ogni ciclo Apply, il programma Apply confronta i valori chiave nella tabella CD della principale con quelli nella tabella CD della replica. Se lo stesso valore chiave esiste in entrambe le tabelle CD, ciò rappresenta un conflitto. Tuttavia, con il rilevamento potenziato (enhanced), il programma Apply attende il commit di tutte le transazioni in fase di trasmissione prima di verificarne i conflitti. Per assicurare che rilevi tutte le transazioni in fase di trasmissione, il programma Apply blocca tutte le tabelle di destinazione nella serie di richieste per evitare ulteriori transazioni e avvia il rilevamento conflitto una volta acquisite tutte le modifiche nella tabella CD. In caso di conflitto, il programma Apply annullerà la transazione di cui è stato eseguito il commit in precedenza nella replica leggendo dalla tabella CD della replica e conservando solo le modifiche che hanno avuto origine dalla principale.

**Limitazione:** anche se si specifica la rilevazione di conflitti avanzata, quando il programma Apply viene eseguito in un ambiente collegato occasionalmente (avviato con la parola chiave COPYONCE), il programma Apply utilizza la rilevazione di conflitti standard.

Il programma Apply non può rilevare dipendenze lette. Se, ad esempio, un'applicazione legge informazioni che vengono successivamente eliminate (mediante un'istruzione DELETE o una transazione di cui è stato eseguito il rollback), il programma Apply non può rilevare la dipendenza.

Se si imposta una configurazione di replica dove è possibile che si verifichino dei conflitti (selezionando nessuna rilevazione o quella standard), è necessario includere un metodo per identificare e gestire eventuali conflitti che si verificano. Anche se l'infrastruttura della replica ha rilevato e ripristinato gli aggiornamenti di transazioni che risultavano in conflitto, il progettista dell'applicazione deve decidere l'azione da intraprendere relativamente alle transazioni di cui è stato eseguito il commit in contemporanea e di cui è stato eseguito il ripristino. Poiché la routine di chiusura ASNDONE viene eseguita alla fine di ogni ciclo di richieste, il progettista dell'applicazione può utilizzarla come punto di avvio per tale logica specifica dell'applicazione. Le informazioni relative gli aggiornamenti in conflitto di cui è stato eseguito il ripristino rimarranno nelle tabelle CD e UOW finché non sono idonee per l'eliminazione del limite di conservazione.

## **Registering tables that use remote journaling (System i)**

System i

When registering System i tables that use remote journaling, you can specify the remote journal as the replication source instead of the local journal.

By selecting the remote journaling option for replication, you move the CD tables, the Capture program, and the Capture control tables to a System i database server that is separate from the System i server that the source table is on.

When you register tables on System i as sources, the default assumes that you do not want to use remote journaling.

**Recommendation:** Whenever you are replicating data from one System i table to another System i table and you have a remote journal set up, it is highly recommended that you use the remote journaling function when registering. Using remote journaling in replication greatly increases performance. Because the remote journal function makes it possible to move the registration, the Capture program, and the Capture control tables away from the system on which the source table resides, more resources are left available on that system. This reduces processor usage and saves disk space. Also, when you use a remote journal that resides at the target server, the CD table is on the same system as the target table, which allows the Apply program to apply changes directly from the CD table to the target table without using a spill file. Not using a spill file reduces the amount of resources used by the Apply program.

**Recommendation:** Register tables that use remote journals as sources only if the registration resides on the same System i system as the replication target. SQL replication allows you to register remote journals as sources even if the registration does not reside on the same System i system as the target, but then you don't get the performance advantages that you do from having the journal on the target system.

Before you register a System i table that uses remote journaling, make sure that your remote journal is in an active state.

**Restrictions:** The following restrictions apply to registered tables that use remote journaling:

- Replica target table types are not supported in a remote journal configuration.
- The query option `SQL_FAST_DELETE_ROW_COUNT` (also known as fast delete) causes journaling to end and should not be used for registered tables. To avoid fast delete you could use a `WHERE` clause in the delete, or you could set the `SQL_FAST_DELETE_ROW_COUNT` parameter in `QAQQINI` to none. Fast delete does not log the individual deletes.
- Do not reorganize the source table by using `RGZPFM` with `ALWCANCEL *YES`. `RGZPFM` with `ALWCANCEL *YES` will create a CE journal entry, which causes the Capture program to signal a full refresh. Use `RGZPFM` with `ALWCANCEL *NO` to reorganize a replication source table.

For more information about the remote journal function, see "Remote journal management" in the i5/OS Information Center.

## Using relative record numbers (RRN) instead of primary keys (System i)

System i

If you are registering a System i table that does not have a primary key, a unique index, or a combination of columns that can be used as a unique index, you must register the table by using the relative record numbers (RRN).

When you choose to replicate by using the RRN, both the CD table and the target table have an extra column, IBMQSQ\_RRN of type INTEGER, which contains a unique value for each row. This column contains the RRN that corresponds to each source table row.

The RRN is used as a primary key for the source table row as long as the source table is not reorganized. When the source table is reorganized, the RRN of each source table row changes; therefore, the RRN in the CD and target table rows no longer has the correct value that reflects the row's new position in the source table.

Any time that you reorganize a source table (to compress deleted rows, for example), DB2 DataPropagator for System i performs a full refresh of all the target tables in the set of that source table. For this reason, place target tables that use RRN as primary keys in subscription sets with other targets that use RRNs, and not in sets with tables that use some other uniqueness factor.

---

## Comportamento delle viste come origini di repliche

Quando si registrano viste per la replica, queste ereditano le opzioni di registrazione delle relative tabelle sottostanti, in particolare le opzioni di replica di modifica e acquisizione o di aggiornamento completo.

I seguenti argomenti descrivono il comportamento delle viste registrate in vari scenari.

### Viste su una sola tabella

È possibile registrare una vista su una singola tabella se la tabella sottostante è registrata per la replica. La vista eredita il tipo di replica dalla tabella sottostante.

#### Solo aggiornamento completo

Se la tabella sottostante è registrata per la replica di solo aggiornamento completo, la vista ha la replica di solo aggiornamento completo. Non è possibile registrare la vista per la replica di di modifica e acquisizione perchè la tabella sottostante non ha una tabella CD associata ad essa che tenga traccia delle modifiche.

#### Modifica e acquisizione

Se la tabella sottostante è registrata per la replica di modifica e acquisizione, la vista ha la replica di modifica e acquisizione e non può essere registrata per il solo aggiornamento completo.

Quando si registra una vista su una tabella che è registrata per la replica di modifica e acquisizione, viene creata una vista sulla tabella CD della tabella sottostante. Tale vista CD contiene solo le colonne a cui viene fatto riferimento dalla vista registrata.

Nella vista non è possibile registrare una serie secondaria di colonne. Tutte le colonne nella vista vengono registrate automaticamente.

### Viste sull'unione di due o più tabelle

Quando si registra una vista sull'unione di due o più tabelle, deve essere registrata almeno una delle tabelle sottostanti l'unione. È possibile avere anche unioni interne di tabelle CCD che sono registrate come origini.

Quando si registra un'unione come un'origine di replica, la replica SQL aggiunge più righe nella tabella IBMSNAP\_REGISTER con identici valori SOURCE\_OWNER e SOURCE\_TABLE. Tali righe sono distinte dai relativi valori SOURCE\_VIEW\_QUAL. Ognuna di queste voci identifica un componente dell'unione.

**Limitazione:** se si definisce un'unione che include una tabella CCD, tutte le altre tabelle in tale unione devono essere tabelle CCD.

Affinché una vista di unione sia un'origine di replica possibile, è necessario crearla utilizzando un ID di correlazione. (Le viste su singole tabelle non richiedono un ID di correlazione).

**Esempio:**

```
create view REGRES1.VW000 (c000,c1001,c2001,c2002,c1003) as
  select a.c000,a.c001,b.c001,b.c002,a.c003
  from REGRES1.SRC001 a, REGRES1.SRC005 b
  where a.c000=b.c000;
```

VW000 è il nome della vista. SRC001 e SRC005 sono le tabelle che fanno parte della vista e C000, C001, C002 e C003 sono le colonne che fanno parte della vista a condizione che le colonne C000 siano uguali in entrambe le tabelle (SRC001 e SRC005).

Il tipo di replica che la vista eredita dipende dalla combinazione delle relative tabelle sottostanti, ognuna delle quali può essere:

- Registrata per la replica di modifica e acquisizione
- Registrata per la replica di solo aggiornamento completo
- Non registrata

La Tabella 3 mostra le varie combinazioni di tabelle sottostanti e il tipo di vista di origine e CD risultante da ciascuna combinazione.

*Tabella 3. Combinazione delle viste sottostanti per le viste*

Tabella 1	Tabella 2	Descrizione della vista di unione e della vista CD
Registrata per modifica e acquisizione	Registrata per modifica e acquisizione	La vista è registrata per la replica di modifica e acquisizione. Le viste CD contengono le colonne a cui viene fatto riferimento dalla tabella CD della Tabella 1 e dalla tabella CD della Tabella 2.
Registrata per modifica e acquisizione	Registrata per solo aggiornamento completo	La vista è registrata per la replica di modifica e acquisizione. La vista CD contiene le colonne a cui viene fatto riferimento dalla tabella CD della Tabella 1 e quelle a cui viene fatto riferimento dalla Tabella 2. Durante ogni ciclo di replica, verrà eseguita la replica sulla destinazione della vista registrata solo delle modifiche alle colonne presenti nella Tabella 1.
Registrata per solo aggiornamento completo	Registrata per solo aggiornamento completo	La vista è registrata per la replica di solo aggiornamento completo. Non esiste alcuna vista CD.
Registrata per solo aggiornamento completo	Non registrata	La vista è registrata per la replica di solo aggiornamento completo. Non esiste alcuna vista CD.

Tabella 3. Combinazione delle viste sottostanti per le viste (Continua)

Tabella 1	Tabella 2	Descrizione della vista di unione e della vista CD
Registrata per modifica e acquisizione	Non registrata	La vista è registrata per la replica di modifica e acquisizione. La vista CD contiene le colonne a cui viene fatto riferimento dalla tabella CD della Tabella 1 e quelle a cui viene fatto riferimento dalla Tabella 2. Durante ogni ciclo di replica, verrà eseguita la replica sulla destinazione della vista registrata solo delle modifiche alle colonne presenti nella Tabella 1.
Non registrata	Non registrata	La vista non è un'origine di replica valida e non può essere registrata.

## Come evitare eliminazioni doppie

Quando si definisce una vista che include due o più tabelle di origine come origine di replica, è necessario fare attenzione ad evitare eliminazioni doppie. Un'eliminazione doppia si verifica quando si elimina una riga durante lo stesso ciclo di replica da entrambe le tabelle che fanno parte di una vista. Ad esempio, si supponga di creare una vista che contiene la tabella CLIENTI e la tabella CONTRATTI. Si verifica un'eliminazione doppia se si elimina una riga dalla tabella CLIENTI e inoltre si elimina la riga corrispondente (dal punto di unione della vista) dalla tabella CONTRATTI durante lo stesso ciclo di replica. Il problema consiste nel fatto che, poiché la riga è stata eliminata dalle due tabelle di origine dell'unione, la riga non risulta nelle viste (né nella vista di base che in quelle della tabella CD), e quindi l'eliminazione doppia non può essere replicata sulla destinazione.

Per evitare eliminazioni doppie, è necessario definire una tabella CCD per una delle tabelle di origine nell'unione. Tale tabella CCD deve essere concentrata e non completa e deve essere ubicata sul server di destinazione. La definizione di una tabella CCD concentrata e non completa per una delle tabelle di origine nell'unione risolve nella maggior parte delle situazioni il problema dell'eliminazione doppia poiché la colonna IBMSNAP\_OPERATION nella tabella CCD consente di rilevare le eliminazioni. Aggiungere semplicemente un'istruzione SQL alla definizione della serie di richieste che deve essere eseguita dopo il ciclo di richieste. Tale istruzione SQL elimina tutte le righe dalla tabella di destinazione per cui IBMSNAP\_OPERATION corrisponde a "D" nella tabella CCD.

Possono comunque verificarsi ancora problemi con gli aggiornamenti e le eliminazioni se, durante lo stesso ciclo Apply, una riga viene aggiornata sulla tabella di origine che contiene la tabella CCD mentre la riga corrispondente viene eliminata sull'altra tabella nell'unione. Ne consegue che, il programma Apply non è in grado di individuare la riga corrispondente nella tabella unita e non può eseguire la replica del valore aggiornato.

## Registrazione di viste di tabelle come origini

Quando si registra una vista come origine di una replica, la vista eredita le opzioni di registrazione della tabella di origine su cui si basa la vista.

### Prima di iniziare

- Le tabelle di controllo Capture devono esistere già sul server di controllo Capture che elaborerà la vista che si desidera registrare come un'origine.

- Il nome delle viste di origine deve seguire le convenzioni di assegnazione dei nomi di tabella DB2.
- È necessario registrare almeno una delle tabelle base sottostanti la vista come un'origine. Quando si registra la tabella di base, utilizzare lo stesso schema Capture che si prevede di utilizzare quando si registra la vista.

### Restrizioni

- Non è possibile registrare viste di tabelle relazionali non DB2.
- Non è possibile registrare una vista che si trova su un'altra vista.
- Tutte le tabelle CCD che hanno viste definite su di esse devono essere complete e concentrate per essere registrate come origini di replica.
- **System i** Poiché le istruzioni SQL sono limitate ad una lunghezza di 32.000 caratteri, è possibile registrare solo circa 2.000 colonne per vista; il numero esatto di colonne dipende dalla lunghezza dei nomi di colonna.

### Procedure

Utilizzare uno dei seguenti metodi per registrare una vista:

Metodo	Descrizione
Programma della riga comandi ASNCLP	Utilizzare il comando CREATE REGISTRATION e specificare il nome della vista per <i>objowner</i> (proprietario oggetto) e <i>objname</i> (nome oggetto).  Relativamente alle viste, il comando decide se l'origine può essere registrata come aggiornamento completo o differenziale.
Centro di replica	Utilizzare la finestra Registra viste. Espandere lo schema Capture sotto il quale si desidera registrare le viste. Fare clic con il tasto destro del mouse sulla cartella <b>Viste registrate</b> e fare clic su <b>Registra viste.</b>
<b>System i</b> Comando di sistema ADDDPRREG	Utilizzare il comando ADDDPRREG per registrare una vista su System i.

## Gestione di tabelle CCD come origini (IMS)

Se si dispone di tabelle CCD popolate esternamente gestite da un programma come IMS DataPropagator o DataRefresher, è necessario gestire tali tabelle in modo che il programma Apply possa leggere le tabelle CCD come origini.

### Procedure

Per gestire una tabella CCD popolata da uno strumento esterno:

Aggiornare tre colonne nella tabella IBMSNAP\_REGISTER (CCD\_OLD\_SYNCHPOINT, SYNCHPOINT e SYNCHTIME) per ognuno dei seguenti tipi di evento:

Evento	Aggiornamenti richiesti
<b>Aggiornamento completo iniziale o caricamento della tabella CCD</b>	<ul style="list-style-type: none"> <li>• Impostare CCD_OLD_SYNCHPOINT su un valore che rappresenta il valore minimo di IBMSNAP_COMMITSEQ dalla tabella CCD.</li> <li>• Impostare SYNCHPOINT su un valore che rappresenta il valore massimo di IBMSNAP_COMMITSEQ dalla tabella CCD. Non impostare SYNCHPOINT su 0. Se si stanno creando propri valori per la sequenza, iniziare con un valore SYNCHPOINT di 1.</li> <li>• Impostare SYNCHTIME su un valore che rappresenta il valore data/ora massimo di IBMSNAP_LOGMARKER dalla tabella CCD.</li> </ul>
<b>Qualsiasi aggiornamento alla tabella CCD successivo all'aggiornamento completo o caricamento</b>	<ul style="list-style-type: none"> <li>• Non modificare il valore CCD_OLD_SYNCHPOINT.</li> <li>• Impostare SYNCHPOINT su un valore che rappresenta il nuovo valore massimo di IBMSNAP_COMMITSEQ dalla tabella CCD.</li> <li>• Impostare SYNCHTIME su un valore che rappresenta il nuovo valore data/ora massimo di IBMSNAP_LOGMARKER dalla tabella CCD.</li> </ul>
<b>Qualsiasi aggiornamento completo successivo o caricamento della tabella CCD</b>	<ul style="list-style-type: none"> <li>• Impostare CCD_OLD_SYNCHPOINT su un valore che rappresenta il valore minimo di IBMSNAP_COMMITSEQ dalla tabella CCD.</li> <li>• Impostare SYNCHPOINT su un valore che rappresenta il valore massimo di IBMSNAP_COMMITSEQ dalla tabella CCD.</li> <li>• Impostare SYNCHTIME su un valore che rappresenta il valore data/ora massimo di IBMSNAP_LOGMARKER dalla tabella CCD.</li> </ul>

**Importante:** ciò assume che i valori utilizzati nella tabella CCD per IBMSNAP\_COMMITSEQ e IBMSNAP\_LOGMARKER sono sempre valori di incremento. Il programma Apply non rileverà un aggiornamento completo eseguito sulla tabella CCD di origine a meno che il valore CCD\_OLD\_SYNCHPOINT è maggiore del valore SYNCHPOINT applicato più di recente.





---

## Capitolo 5. Sottoscrizione alle origini per la replica SQL

Dopo avere registrato le tabelle o le viste come origini della replica, è possibile definire una sottoscrizione per le proprie viste o tabelle di destinazione, in modo che esse possano ricevere i dati di origine iniziali e le successive modifiche.

Le attività di gestione descritte in questa sezione consentono di configurare le informazioni di controllo utilizzate dai programmi Apply e Capture per copiare i dati di origine o per catturare i dati modificati e replicarli nelle tabelle di destinazione in intervalli appropriati.

I seguenti argomenti forniscono informazioni dettagliate sulla sottoscrizione alle origini.

---

### Pianificazione del modo in cui raggruppare le origini e le destinazioni

Prima di definire quali destinazioni richiedono quali origini, è necessario definire il modo in cui si desidera raggruppare le origini e le destinazioni.

La replica SQL elabora le associazioni tra origini e destinazioni in gruppi. Questi gruppi sono composti da una o più origini che vengono elaborate dallo stesso programma Capture e da una o più destinazioni, che richiedono tutti o parte dei dati di origine, che vengono elaborate dallo stesso programma Apply. Questi gruppi sono detti *serie di richieste* e le associazioni tra origine e destinazione vengono dette *membri della serie di richieste*.

Quando si pianificano le serie di richieste, si osservino le seguenti regole e i seguenti vincoli:

- Una serie di richieste associa un server di origine con un server di destinazione. Un membro della serie di richieste associa una vista o una tabella di origine con una vista o una tabella di destinazione. Le serie di richieste e i membri delle serie di richieste vengono memorizzati sul server di controllo Apply.
- Il programma Apply elabora tutti i membri in una serie di richieste come un singolo gruppo. Per questo motivo, se un membro della serie di richieste richiede una copia dell'aggiornamento completo, vengono aggiornati tutti i membri dell'intera serie.
- Tutte le viste e le tabelle di origine nei membri di una serie devono presentare lo stesso schema Capture.
- Su System i, tutte le tabelle di origine nei membri di una serie di richieste devono essere inserite nello stesso giornale.
- Tutte le tabelle CCD esterne create da IMS DataPropagator che sono membri di una serie di richieste devono presentare lo stesso schema Capture.

Un singolo programma Apply, che presenta un qualificatore Apply univoco, può elaborare una o più serie di richieste. Una singola serie di richieste può contenere uno o più membri della serie di richieste.

Nei seguenti argomenti viene illustrato i controbilanciamenti nel raggruppamento delle serie di richieste per programma Apply e dei membri delle serie di richieste per serie di richieste.

## Pianificazione del numero di membri della serie di richieste

Quando si aggiungono i membri in una serie di richieste, è necessario decidere se raggruppare tutte le coppie origine-destinazione (membri della serie di richieste) in una serie di richieste, creare serie di richieste separate per ciascuna coppia o un piccolo numero di serie di richieste, ciascuna con un numero di coppie.

Poiché il programma Apply replica i membri di una serie di richieste in una transazione (logica), è necessario raggruppare più membri in una serie di richieste nelle seguenti situazioni:

- Se le tabelle di origine sono correlate logicamente tra loro.
- Se le tabelle di destinazione presentano dei vincoli di integrità referenziali.

Raggruppando diversi membri in una serie di richieste, è possibile verificare che la replica per tutti i membri inizi nello stesso momento. Inoltre, viene ridotto il numero di connessioni con il database necessarie per elaborare le serie di richieste e i costi di gestione dell'ambiente della replica. Se la serie di richieste contiene istruzioni SQL o procedure memorizzate, è possibile utilizzare tali istruzioni o procedure per elaborare tutti i membri della serie di richieste.

Se non vi sono relazioni di integrità referenziali o logiche tra le tabelle in una serie di richieste, è possibile raggrupparle in una serie di richieste o in diverse serie di richieste. La riduzione del numero delle serie di richieste consente di semplificare la gestione dell'ambiente della replica. Tuttavia, aumentando il numero di serie di richieste, viene ridotta l'influenza degli errori della replica.

Per individuare più facilmente gli errori che causano un malfunzionamento del programma Apply, aggiungere solo un piccolo numero di membri in una serie di richieste. Con un numero limitato di membri, sarà possibile individuare più facilmente la causa dei problemi. Se un membro della serie di richieste presenta un errore, viene eseguito il rollback di tutti i dati che sono stati applicati agli altri membri della serie. In tal modo, nessun membro potrà completare il ciclo se non lo completano anche tutti gli altri membri. Il programma Apply esegue il rollback di una serie di richieste con errore per ripristinare l'ultimo punto di commit corretto, che potrebbe essere nel ciclo Apply corrente, se all'avvio del programma Apply è stata specificata la parola chiave `commit_count`.

## Pianificazione del numero di serie di richieste per qualificatore Apply

Quando si definisce una serie di richieste, viene specificato il qualificatore Apply per tale serie di richieste. Il qualificatore Apply associa un'istanza del programma Apply con una o più serie di richieste.

Ciascuna serie di richieste viene elaborata da un solo programma Apply, ma ciascun programma Apply può elaborare una o più serie di richieste durante ciascun ciclo di Apply.

È possibile eseguire il numero desiderato di istanze del programma Apply (ciascuna con il proprio qualificatore Apply) e ciascun programma Apply può elaborare il numero desiderato di serie di richieste. Sono disponibili due opzioni di base:

### **Associare ciascun qualificatore Apply con una serie di richieste**

Ciascun programma Apply elabora esattamente una serie di richieste.

Se la velocità è importante, è possibile distribuire le serie tra diversi qualificatori Apply, che consentono di eseguire diverse istanze del programma Apply contemporaneamente.

Se si desidera che un'istanza del programma Apply elabori una serie di richieste, è possibile utilizzare l'opzione di avvio OPT4ONE del programma Apply, che carica in memoria le informazioni relative alle tabelle di controllo per la serie di richieste.

Se si utilizza questa opzione, il programma Apply non legge le tabelle di controllo per le informazioni relative alla serie di richieste per ciascun ciclo di Apply. Pertanto, le prestazioni del programma Apply sono migliori. Tuttavia, maggiore è il numero delle istanze del programma Apply, maggiore sarà il numero delle risorse del sistema utilizzate e più lente saranno le prestazioni generali.

#### **Associare ciascun qualificatore Apply con più serie di richieste**

Ciascun programma Apply elabora molte serie di richieste.

Utilizzando più di un qualificatore Apply, è possibile eseguire più di un'istanza del programma Apply da un singolo ID utente.

Il programma Apply tenta di mantenere il più aggiornate possibile tutte le serie per un determinato qualificatore Apply. Quando viene avviato un ciclo di Apply, il programma Apply determina quella serie di richieste contiene i dati meno aggiornati ed inizia ad elaborare prima quella serie.

Se la velocità non è importante, è possibile replicare molte serie di richieste con un qualificatore Apply. Ad esempio, si consiglia di attendere la fine dell'orario di lavoro prima di eseguire la replica.

Tuttavia, quando un programma Apply elabora più serie di richieste, queste ultime vengono elaborate in sequenza, con la possibilità che venga incrementata la latenza generale della replica.

Se vi sono requisiti specifici per determinate serie di richieste, è possibile associare queste due opzioni. Ad esempio, è possibile che un programma Apply elabori la maggior parte delle serie di richieste, traendo vantaggio dall'utilizzo di un programma Apply per elaborare insieme le serie di richieste correlate, mentre un altro programma Apply elabora una singola serie di richieste, garantendo la minima latenza della replica per quella serie di richieste. Utilizzando le due istanze del programma Apply, viene incrementato il parallelismo generale per le serie di richieste.

---

## **Creazione di serie di richieste**

Prima di replicare i dati da un'origine registrata, è necessario creare le serie di richieste, che sono delle raccolte di membri della serie di richieste (associazioni tra origine e destinazione) che il programma Apply elabora come una serie.

#### **Prima di iniziare**

- Creare le tabelle di controllo Apply sul server di controllo Apply per la serie di richieste.
- Prima di aggiungere i membri della serie di richieste nelle serie di richieste, registrare le tabelle o le viste che si desidera utilizzare come origini. Inoltre, è necessario definire come si desidera raggruppare le serie.

#### **Informazioni su questa attività**

Quando si crea una serie di richieste, vengono specificati i server di origine e destinazione, i programmi Capture e Apply che si desidera utilizzare, quando e come si desidera che il programma Apply elabori la serie.

Non è necessario aggiungere i membri della serie di richieste in una serie di richieste. È possibile creare una serie vuota che non contiene nessuna associazione tra origine e destinazione. È utile creare una serie vuota per i seguenti motivi:

- Si pensa di aggiungere dei membri in una serie successivamente e non si desidera attivare la serie di richieste finché non vengono aggiunti i membri.
- Si desidera che il programma Apply elabori la serie di richieste vuota per chiamare un'istruzione SQL o una procedura memorizzata quando la serie è pronta per essere elaborata.

## Procedure

Per creare una serie di richieste, utilizzare uno dei seguenti metodi:

Metodo	Descrizione
<b>Programma della riga comandi ASNCLP</b>	<p>Utilizzare il comando CREATE SUBSCRIPTION SET. Questo comando può creare solo serie di richieste vuote, mentre il Centro di replica consente di aggiungere i membri nella serie durante la relativa creazione.</p> <p>I seguenti comandi impostano l'ambiente e creano una serie di richieste denominata SET00 con un qualificatore Apply AQ00.</p> <pre>SET SERVER CAPTURE TO DB SAMPLE; SET SERVER CONTROL TO DB TARGET; SET OUTPUT CAPTURE SCRIPT "capsubset.sql" CONTROLSCRIPT "appsubset.sql"; SET LOG "subset.err"; SET RUN SCRIPT LATER; CREATE SUBSCRIPTION SET SETNAME SET00 APPLYQUAL AQ00 ACTIVATE YES TIMING INTERVAL 1 START DATE "2006-10-22" TIME "09:00:00.000000";</pre>
<b>Centro di replica</b>	<p>Utilizzare il blocco note Crea serie di richieste. Per visualizzare il blocco note, espandere il server di controllo Apply su cui verrà definita la serie, fare clic con il tasto destro del mouse sulla cartella <b>Serie di richieste</b> e fare clic su <b>Crea</b>.</p>
<b>System i</b> Comando di sistema ADDDPRSUB	<p>Utilizzare il comando per l'aggiunta della serie di richieste DPR (ADDDPRSUB) per creare una serie di richieste con un membro o nessun membro.</p> <p>Ad esempio, per creare una serie di richieste denominata SETHR nel qualificatore Apply AQHR:</p> <pre>ADDDPRSUB APYQUAL(AQHR) SETNAME(SETHR) SRCTBL(HR/EMPLOYEE) TGTTBL(TGTLIB/TGTEMPL)</pre> <p>Questa serie di richieste, contenente un membro della serie di richieste, replica i dati dalla tabella di origine registrata denominata EMPLOYEE nella libreria HR nella tabella di destinazione denominata TGTEMPL nella libreria TGTLIB.</p>

L'utente fornisce queste caratteristiche di base:

### Alias del server di controllo Apply

L'alias locale del server contenente le tabelle di controllo per il programma Apply che elaborerà la serie di richieste. Definire lo stesso alias per il

server di controllo Apply in ciascun database da cui viene eseguito il Centro di replica, ASNCLP oppure il programma Apply, in modo che gli strumenti di gestione popolino correttamente le tabelle di controllo Apply e ciascun programma Apply si connetta al server corretto mediante un nome alias standard.

#### **Nome della serie di richieste**

Il nome della serie di richieste. Sul server di controllo Apply che elabora questa serie di richieste, il nome della serie deve essere univoco per un determinato qualificatore Apply. Il nome può contenere un numero massimo di 18 caratteri.

#### **Qualificatore Apply**

Il nome di un qualificatore Apply nuovo o esistente, che identifica il programma Apply che elaborerà questa serie di richieste. È possibile utilizzare lo stesso qualificatore Apply per elaborare diverse serie di richieste. Le serie di richieste che presentano lo stesso qualificatore Apply devono essere definite sullo stesso server di controllo Apply.

#### **Alias del server di controllo Capture**

L'alias del server contenente le tabelle di controllo per il programma Capture che elaborerà le origini registrate per la serie di richieste. Definire lo stesso alias per il server di controllo Capture in ciascun database da cui viene eseguito il Centro di replica, ASNCLP oppure il programma Apply, in modo che gli strumenti di gestione popolino correttamente le tabelle di controllo Capture e Apply e ciascun programma Apply si connetta al server corretto mediante un nome alias standard.

#### **Schema Capture**

Il nome dello schema Capture che identifica la serie di tabelle di controllo Capture, che definiscono le origini registrate per la serie di richieste. Tutte le tabelle di origine in una serie di richieste devono risiedere sullo stesso server e solo il programma Capture può catturare le modifiche per esse.

#### **Alias del server di destinazione**

Il nome del server di destinazione contenente le tabelle o le viste cui il programma Apply replicherà le modifiche dall'origine. Definire lo stesso alias per il server di destinazione in ciascun database da cui viene eseguito il Centro di replica, ASNCLP oppure il programma Apply, in modo che gli strumenti di gestione popolino correttamente le tabelle di controllo Apply e ciascun programma Apply si connetta al server corretto mediante un nome alias standard.

Quando si crea una serie di richieste, è possibile utilizzare le impostazioni predefinite per il modo in cui il programma Apply elabora la serie oppure modificare le proprietà della sottoscrizione per soddisfare le proprie esigenze di replica.

---

## **Opzioni di elaborazione per le serie di richieste**

Quando si crea una serie di richieste, vengono definite le opzioni relative al modo in cui il programma Apply elabora la serie.

Di seguito vengono illustrate le impostazioni da selezionare in base alle esigenze della replica.

## Come specificare se la serie di richieste è attiva

È possibile specificare se si desidera che il programma Apply inizi ad elaborare la serie di richieste. Quando si attiva una serie di richieste, il programma Apply avvia un aggiornamento completo della serie.

È possibile scegliere tra tre livelli di attivazione:

**Attiva** Il programma Apply elabora la serie durante il ciclo successivo. Attivare la serie se si desidera che il programma Apply la elabori durante la successiva esecuzione. Se lo si desidera, è possibile aggiungere dei membri nella serie successivamente. Quando si attiva la serie, essa rimane attiva e il programma Apply continua ad elaborarla finché non viene disattivata.

### **Inattiva**

Il programma Apply non elabora la serie. Lasciare la serie inattiva, se non si desidera che il programma Apply la elabori.

### **Attiva solo una volta**

Il programma Apply elabora la serie durante il ciclo successivo e disattiva la serie. Specificare questa opzione se si desidera che la serie venga eseguita solo una volta. Aggiungere tutti i membri della serie di richieste prima di selezionare questa opzione, perché il programma Apply non elabora i membri aggiunti successivamente, a meno che non si riattivi la serie di richieste.

## Specifica del numero di minuti di dati richiamati dal programma Apply

È possibile specificare un numero approssimativo di minuti di dati che verranno richiamati dal programma Apply dall'origine della replica durante ciascun ciclo di Apply.

Questa opzione è utile in diverse situazioni:

- Quando è necessario elaborare una grande quantità di dati in ciascun ciclo della serie di richieste.  
Le serie di richieste che replicano grandi blocchi di modifiche in un ciclo di Apply possono causare un'eccedenza di file di trasferimento o di registrazioni (per il database di destinazione). Ad esempio, gli scenari batch di Apply possono produrre un backlog di transazioni accodate di grandi dimensioni da replicare.
- Un'interruzione estesa della rete può causare l'accumulo di un grande blocco di dati nelle tabelle CD, che può causare un'eccedenza della registrazione della destinazione e del file di trasferimento del programma Apply.

Il numero di minuti specificato viene detto blocco di dati. Il valore di blocco dei dati specificato viene memorizzato nella colonna MAX\_SYNCH\_MINUTES della tabella IBMSNAP\_SUBS\_SET. Se l'accumulo di dati è maggiore della dimensione del blocco di dati, il programma Apply converte un singolo ciclo di Apply in diversi mini-cicli. Se le risorse non sono ancora sufficienti per gestire il fattore di blocco fornito, il programma Apply riduce la dimensione del blocco di dati in modo che corrisponda alle risorse disponibili del sistema. Richiamando serie di dati più piccole, il programma Apply può ridurre il carico della rete e lo spazio temporaneo richiesto per i dati richiamati.

Durante ciascun ciclo di Apply, se il valore MAX\_SYNCH\_MINUTES di una serie di richieste è NULL o è impostato su un valore numerico minore di 1, il programma Apply elabora tutti i dati idonei per quella serie in un singolo ciclo di

Apply. Se le tabelle CD e UOW contengono grandi volumi di dati, è possibile che si verifichino problemi quali il riempimento della registrazione delle transazioni del database o un'eccedenza del file di trasferimento. È possibile impostare MAX\_SYNCH\_MINUTES su un valore non NULL utilizzando le seguenti linee guida:

- Se la colonna SLEEP\_MINUTES della tabella ASN.IBMSNAP\_SUBS\_SET è impostata su 5 minuti (o meno) per una determinata serie di richieste, impostare MAX\_SYNCH\_MINUTES su 5 minuti.
- Se SLEEP\_MINUTES è impostato su 30 minuti (o meno) per una determinata serie di richieste, impostare MAX\_SYNCH\_MINUTES su 60 minuti.
- Se SLEEP\_MINUTES è impostato tra 5 e 30 minuti, impostare il valore di MAX\_SYNCH\_MINUTES in modo che sia uguale a SLEEP\_MINUTES.

Monitorare l'ambiente della replica e regolare il valore di MAX\_SYNCH\_MINUTES in base alle proprie esigenze. Verificare che il valore numerico di MAX\_SYNCH\_MINUTES sia maggiore di zero.

**Esempio:** se si specifica che il programma Apply dovrà richiamare al massimo 10 minuti di dati per mini-ciclo, il programma Apply richiamerà una quantità di dati di cui è stato eseguito il commit dalla tabella CD nell'origine compresi in circa 10 minuti dell'ultimo mini-ciclo.

Oltre ad evitare le eccedenze delle registrazioni e dei file di trasferimento, i mini-cicli presenta molti altri vantaggi. In caso di errori durante il ciclo di replica, il programma Apply deve eseguire il rollback delle sole modifiche apportate durante il mini-ciclo che non è stato eseguito correttamente. Se la replica non viene eseguita correttamente durante un mini-ciclo, il programma Apply tenta di elaborare la serie di sottoscrizioni dall'ultimo mini-ciclo eseguito correttamente. Ciò consente di risparmiare molto tempo per le elaborazioni di una grande quantità di dati. Figura 5 illustra in che modo i dati modificati sono disponibili nei sottoinsiemi delle modifiche.

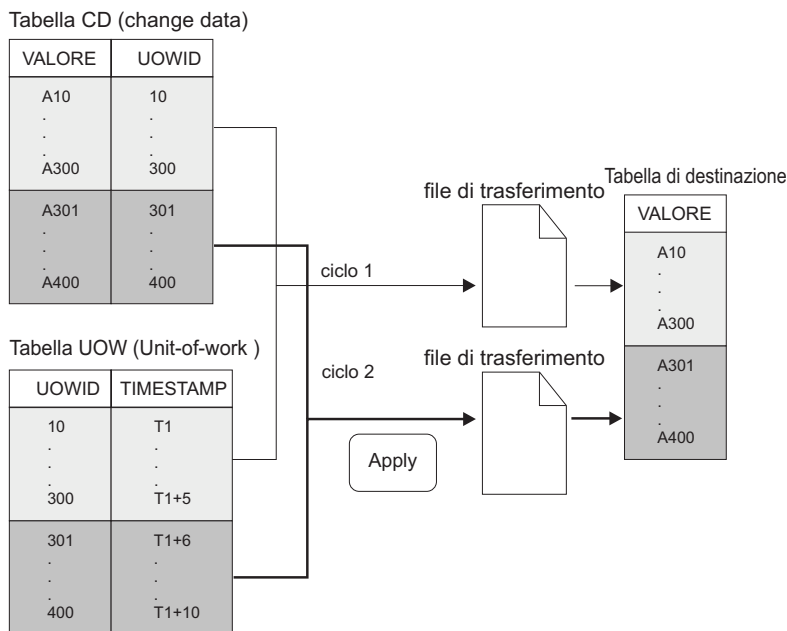


Figura 5. Blocco dei dati. È possibile ridurre la quantità di traffico della rete specificando un valore di blocco dei dati.

Il numero di minuti impostato deve essere abbastanza limitato, in modo che tutte le transazioni per la serie di richieste verificatesi durante l'intervallo possano essere copiate senza causare eccedenze dei file di trasferimento o della registrazione durante il mini-ciclo.

Durante l'elaborazione dei dati, il programma Apply non esegue nessuna delle seguenti operazioni:

- Dividere una UOW (unit of work). Ciò significa che un lungo lavoro batch in esecuzione senza commit non può essere interrotto dal fattore di blocco dati.
- Rollback dei mini-cicli di richieste di cui è stato eseguito il commit.
- Utilizzare il fattore di blocco dati durante un aggiornamento completo.

## **Opzioni di caricamento per le tabelle di destinazione con integrità referenziale**

In alcuni casi, si potrebbe voler rimandare l'aggiunta dei vincoli di integrità referenziale tra tabelle di destinazione a dopo che tali tabelle sono caricate con i dati di origine.

La decisione di come caricare le destinazioni viene intrapresa quando si impostano i parametri di avvio per il programma Apply. Considerare queste alternative per la creazione delle relazioni di integrità referenziale tra le tabelle di destinazione:

### **Prima che le tabelle di destinazione vengano caricate**

Ciò richiede che non vengano apportate modifiche alla tabella di origine durante l'intera fase di estrazione e caricamento della tabella di destinazione. Inoltre, è necessario avviare il programma Apply utilizzando l'opzione di avvio LOADX per evitare il controllo di vincoli referenziali durante il caricamento. Se non si utilizza l'opzione LOADX, gli inserimenti nella tabella di destinazione potrebbero avere esito negativo. Generalmente, se si utilizza l'opzione di avvio LOADX l'aggiornamento completo è molto più rapido.

### **Una volta terminato il caricamento e dopo che il programma Apply ha completato un ciclo di applicazione delle modifiche alle tabelle di destinazione**

Con tale opzione, le modifiche possono essere apportate alla tabella di origine mentre vengono caricate le tabelle di destinazione. È possibile avviare il programma Apply con o senza l'opzione di avvio LOADX, in quanto non sono presenti vincoli che necessitano di essere superati. Durante il popolamento iniziale delle tabelle di destinazione, queste potrebbero non essere sincrone una con l'altra riguardo alla relativa relazione di integrità referenziale. Quando le tabelle vengono caricate, tutte le modifiche vengono catturate per la serie. Una volta che il programma Apply replica la prima serie di modifiche, tutte le tabelle di destinazione conterranno le stesse transazioni e avranno integrità referenziale. A questo punto, è possibile disattivare la serie, aggiungere i vincoli di integrità referenziale e quindi riattivare la serie.

## **Specifiche della modalità in cui il programma Apply replica le modifiche per i membri della serie di sottoscrizioni**

Quando una serie di richieste presenta una replica di modifica/cattura, è possibile decidere se il programma Apply eseguirà il commit delle modifiche nella vista o nella tabella di destinazione una volta per ciascun membro della serie di richieste o dopo avere eseguito una serie di transazioni.



Dopo avere caricato le tabelle di destinazione, il programma Apply inizia la lettura delle tabelle CD (o CCD) e raccoglie le modifiche in file di trasferimento. Il programma, quindi, applica le modifiche in uno dei seguenti modi:

#### **Modalità tabella**

Il programma Apply esegue il commit delle modifiche una volta per ciascun membro della serie di richieste.

Il programma Apply legge tutte le modifiche da un file di trasferimento per una tabella CD (o CCD), applica le modifiche nelle tabelle di destinazione corrispondenti e inizia ad elaborare il file di trasferimento per la tabella CD (o CCD). Al termine della lettura e dopo avere applicato le modifiche in tutte le tabelle CD (o CCD) della serie, esegue il commit del DB2 di tutte le modifiche apportate alle tabelle di destinazione nella serie di richieste.

#### **Modalità transazione**

Il programma Apply esegue il commit delle modifiche dopo avere eseguito una serie di transazioni specificate dall'utente. Utilizzare l'elaborazione in modalità transazione quando sono presenti dei vincoli di integrità referenziale nelle tabelle di destinazione nella serie di richieste.

In questa modalità, il programma Apply visualizza tutti i file di trasferimento una sola volta ed elabora le modifiche nello stesso tempo. Le modifiche vengono applicate nell'ordine in cui sono state eseguite nelle tabelle di origine. La colonna COMMIT\_COUNT nella tabella IBMSNAP\_SUBS\_SET controlla il modo in cui vengono applicate le modifiche e viene eseguito il commit delle modifiche in tutte le tabelle di destinazione per quella serie di richieste.

L'elaborazione in modalità transazione modifica il funzionamento del programma Apply solo per le serie con tabelle di destinazione copia dell'utente e istantanea. Le serie contenenti le tabelle della replica vengono sempre elaborate in modalità transazione.

Un solo commit può ridurre la latenza per la serie di richieste, ma più commit consentono al programma Apply di applicare i dati nella sequenza di commit originale.

È anche possibile utilizzare un misto di elaborazione in modalità transazione e in modalità tabella, a seconda dei tipi di tabella di destinazione nella serie di richieste.

## **Definizione delle procedure memorizzate e delle istruzioni SQL per la serie di richieste**

È possibile definire le istruzioni SQL o le procedure memorizzate che vengono eseguite ogniqualvolta il programma Apply elabora la serie di richieste. Queste istruzioni possono essere utili per eliminare le tabelle CCD o per manipolare i dati di origine prima che essi vengano applicati alle destinazioni.

È possibile specificare quando e dove eseguire le istruzioni SQL o le procedure memorizzate:

- Sul server di controllo Capture prima che il programma Apply applichi i dati.
- Sul server di destinazione prima che il programma Apply applichi i dati.
- Sul server di destinazione dopo l'applicazione dei dati da parte del programma Apply.

Quando si utilizza il Centro di replica per aggiungere le istruzioni SQL in una serie di richieste, è possibile fare clic su **Prepara istruzione** nella finestra Aggiungi istruzione SQL o chiamata di procedura per verificare la sintassi.

## Opzioni per pianificare la replica delle serie di richieste

È possibile specificare la frequenza con cui il programma Apply elabora una serie di richieste per controllare la ricorrenza dei dati nelle tabelle di destinazione. È possibile utilizzare una pianificazione in base all'ora, sugli eventi o una combinazione di queste opzioni.

Ad esempio, è possibile impostare un intervallo di un giorno tra i cicli di Apply e specificare un evento che attivi il ciclo. Se si utilizzano entrambe le opzioni di pianificazione, la serie di richieste sarà idonea per essere eseguita all'ora pianificata e quando si verifica l'evento.

Nella replica di aggiornamento ovunque, è possibile utilizzare la stessa durata o una durata diversa per le serie di richieste da tabella principale a replica e da replica a tabella principale.

Se è necessario replicare una grande quantità di dati durante un intervallo o tra gli eventi, il programma Apply potrebbe non essere in grado di elaborare una serie di richieste finché non finisce di applicare i dati per tutte le serie nell'intervallo precedente o per l'evento precedente. In questo caso, è possibile che non si ottenga la latenza della replica prevista, ma i dati non verranno persi.

### Pianificazione in base all'ora

Il metodo più semplice di controllare quando la serie viene elaborata consiste nell'utilizzare la pianificazione in base all'ora (nota anche come durata relativa o durata dell'intervallo). L'utente determina una data di inizio specifica, l'ora e l'intervallo. L'intervallo può essere specifico (da un minuto ad un anno) o continuo, ma gli intervalli di tempo sono approssimativi.

Il programma Apply inizia ad elaborare una serie di richieste appena può, in base al proprio carico di lavoro e alla disponibilità delle risorse. La scelta di un intervallo non garantisce che la frequenza della replica avverrà esattamente in quell'intervallo. Se si specifica una durata continua, il programma Apply replica i dati il più frequentemente possibile.

### Pianificazione basata sugli eventi

Per replicare i dati mediante la pianificazione basata sugli eventi (detta anche durata degli eventi), l'utente specifica il nome di un evento durante la definizione della serie di richieste. È anche necessario inserire i dati nella tabella IBMSNAP\_SUBS\_EVENT con un timestamp per il nome dell'evento. Quando il programma Apply rileva l'evento, inizia la replica.

La tabella IBMSNAP\_SUBS\_EVENT presenta quattro colonne, come illustrato in Tabella 4.

*Tabella 4. Esempio dei dati memorizzati nella tabella IBMSNAP\_SUBS\_EVENT*

EVENT_NAME	EVENT_TIME	END_OF_PERIOD	END_SYNCHPOINT
END_OF_DAY	2002-05-01- 17.00.00.000000	2002-05-01- 15.00.00.000000	

La colonna EVENT\_NAME contiene il nome dell'evento specificato dall'utente durante la definizione della serie di richieste. EVENT\_TIME è il timestamp che indica quando il programma Apply inizierà ad elaborare la serie. END\_OF\_PERIOD è un valore facoltativo che indica che gli aggiornamenti eseguiti dopo l'ora specificata verranno rimandati fino ad un evento o ad un'ora futuri. END\_SYNCHPOINT è un valore facoltativo che indica che gli aggiornamenti eseguiti dopo il numero di sequenza di registrazione specificato devono essere rimandati fino ad un evento o ad un'ora futuri. Se si specificano dei valori per END\_OF\_PERIOD e END\_SYNCHPOINT, il valore di END\_SYNCHPOINT ha la precedenza. Impostare il valore EVENT\_TIME utilizzando l'orologio del server di controllo Apply e Apply e impostare il valore di END\_OF\_PERIOD utilizzando l'orologio del server di origine. Questa distinzione è importante se i due server hanno fusi orari diversi.

In Tabella 4 a pagina 72, per l'evento denominato END\_OF\_DAY, il valore del timestamp per EVENT\_TIME (2002-05-01-17.00.00.000000) è l'ora in cui il programma Apply deve iniziare ad elaborare la serie di richieste. Il valore del timestamp END\_OF\_PERIOD (2000-05-01-15.00.00.000000) è l'ora dopo la quale gli aggiornamenti non vengono replicati e verranno replicati nel ciclo del giorno successivo. Ciò significa che l'evento replica tutti gli aggiornamenti in sospenso eseguiti prima di quest'ora e rimanda tutti gli aggiornamenti successivi.

L'utente o le applicazioni devono inserire gli eventi nella tabella IBMSNAP\_SUBS\_EVENT utilizzando un'istruzione INSERT SQL per inserire una riga nella tabella per attivare l'evento. Ad esempio, utilizzare il timestamp corrente più un minuto per attivare l'evento indicato da EVENT\_NAME. Le serie di richieste collegate a questo evento diventano idonee per essere eseguite tra un minuto. È necessario inserire manualmente gli eventi per l'aggiornamento completo e per la replica di modifica/cattura.

È possibile inserire gli eventi in anticipo, ad esempio per la prossima settimana, l'anno prossimo o ogni sabato. Se il programma Apply è in esecuzione, esso viene avviato approssimativamente all'ora specificata dall'utente. Se il programma Apply viene arrestato all'ora specificata dall'utente, quando viene riavviato, verifica la tabella degli eventi delle richieste e inizia ad elaborare la serie di richieste per l'evento inserito.

Il programma Apply non elimina la tabella. È necessario inserire i dati nella tabella e gestirla. Inoltre, non è possibile utilizzare il Centro di replica per aggiornare la tabella degli eventi di richieste. È necessario eseguire le istruzioni SQL o definire procedure automatiche per aggiungere gli eventi in questa tabella.

**Esempio:**

```
INSERT INTO ASN.IBMSNAP_SUBS_EVENT
  (EVENT_NAME, EVENT_TIME)
VALUES ('EVENT01', CURRENT_TIMESTAMP + 1 MINUTES)
```

Gli eventi che si verificano prima dell'ora più recente in cui il programma Apply ha elaborato la serie di richieste (come specificato dal valore nella colonna LASTRUN della tabella di controllo della serie di richieste) vengono considerati come eventi scaduti e vengono ignorati. Pertanto, se il programma Apply è in esecuzione, inserire gli eventi che sono lievemente nel futuro per evitare di inserire eventi scaduti.

## Pianificazione della serie di richieste

Definire le informazioni relative alla durata della serie di richieste dopo avere associato le origini con le destinazioni (oppure creare una serie di richieste vuota).

Dopo avere associato le origini con le destinazioni (o avere creato una serie di richieste vuota), definire le informazioni relative alla durata della serie di richieste. Nella pagina Pianifica della finestra Crea serie di richieste, specificare quando la serie di richieste dovrà essere idonea per l'elaborazione. Il valore predefinito è la data e l'ora correnti della macchina locale. Inoltre, specificare la frequenza con cui la serie di richieste dovrà essere idonea per l'elaborazione:

- Replica in base all'ora  
Il programma Apply elabora questa serie di richieste utilizzando un intervallo di tempo regolare.
- Replica basata sugli eventi  
Il programma Apply elabora questa serie di richieste quando si verifica un evento.
- Replica in base all'ora e sugli eventi  
Il programma Apply elabora questa serie di richieste utilizzando un intervallo di tempo regolare e il momento in cui si verifica un evento. In questo caso, la serie di richieste è idonea per l'elaborazione all'ora pianificata e quando si verifica l'evento.

## Creazione dei membri delle serie di richieste

In una serie di richieste è possibile aggiungere le associazioni tra origine e destinazione che il programma Apply elaborerà come un gruppo. Queste associazioni tra origine e destinazione vengono denominate membri della serie di richieste.

### Prima di iniziare

Prima di configurare le destinazioni che richiedono le modifiche alle origini, è necessario registrare le tabelle o le viste che si desidera utilizzare come origini. Inoltre, è necessario creare una serie di richieste e pianificare il numero di membri che si desidera aggiungere in una serie.

### Restrizioni

- La replica SQL non supporta le viste di tabelle relazionali non DB2 come origini.
- Se si definisce una vista di destinazione, è necessario che essa possa essere inserita. Ovvero, è necessario potere aggiornare tutte le colonne nella vista e la selezione completa della vista non può includere le parole chiave UNION ALL.
- Se si utilizza il Centro di replica, non è possibile aggiungere una colonna in un membro della serie di richieste, se tale colonna non esiste già nella tabella di destinazione.
- **z/OS:** Non selezionare le colonne ROWID per la replica tranne quando la colonna ROWID è il solo unico indice specificato per la replica.

**suggerimento:** Utilizzare una colonna IDENTITY invece di una ROWID come indice univoco per la replica.

-   È possibile definire un massimo di 200 membri per ogni serie di richieste.

- **System i** È possibile definire un massimo di 78 membri per ogni serie di richieste.

### Informazioni su questa attività

Quando si definisce un membro della serie di richieste, si specifica la vista o la tabella di destinazione che richiede i dati di origine ed è possibile definire il modo in cui i dati replicati verranno visualizzati nella destinazione.

### Procedure

Per aggiungere un membro della serie di richieste, utilizzare uno dei seguenti metodi:

Metodo	Descrizione
<b>Programma della riga comandi ASNCLP</b>	<p>Utilizzare il comando CREATE MEMBER per aggiungere un membro della serie di richieste in una serie di richieste esistente. Ad esempio, i seguenti comandi:</p> <ul style="list-style-type: none"> <li>• Impostare l'ambiente.</li> <li>• Creare un profilo, TBSPROFILE, per la memorizzazione delle opzioni relative al table space utilizzato dalla tabella di destinazione.</li> <li>• Specificare la serie di richieste SET00, il qualificatore Apply AQ00 e la tabella di origine STAFF.</li> <li>• Specificare che una nuova tabella di destinazione, TRGSTAFF, viene creata come copia dell'utente con tutte le colonne registrate.</li> </ul> <pre>SET SERVER CAPTURE TO DB SAMPLE; SET SERVER CONTROL TO DB TARGET; SET SERVER TARGET TO DB TARGET; SET OUTPUT CAPTURE SCRIPT "capmember.sql" CONTROLSCRIPT "appmember.sql" SET LOG "member.err"; SET RUN SCRIPT LATER; SET PROFILE TBSPROFILE FOR OBJECT TARGET TABLESPACE OPTIONS UW USING FILE "/tmp/db/ts/TSTRG.TS" SIZE 700 PAGES; CREATE MEMBER IN SETNAME SET00 APPLYQUAL AQ00 ACTIVATE YES SOURCE STAFF TARGET NAME TRGSTAFF DEFINITION IN TSTRG00 CREATE USING PROFILE TBSPROFILE TYPE USERCOPY COLSALL REGISTERED;</pre>
<b>Centro di replica</b>	<p>Utilizzare uno dei seguenti blocchi note:</p> <ul style="list-style-type: none"> <li>• Crea insieme di richieste. Utilizzare questo blocco note durante la creazione della serie di richieste. Espandere il server di controllo Apply su cui verrà definita la serie, fare clic con tasto destro del mouse sulla cartella <b>Serie di richieste</b> e fare clic su <b>Crea</b>.</li> <li>• Proprietà della serie di richieste. Utilizzare questo blocco note se è già stata creata la serie di richieste e si desidera aggiungervi uno o più membri della serie di richieste. Fare clic con il tasto destro del mouse sulla serie di richieste e selezionare <b>Proprietà</b>.</li> <li>• Aggiungi i membri nella serie di richieste. Utilizzare questo blocco note per aggiungere un membro in più serie di richieste. Ciascun membro deve utilizzare la stessa origine. Fare clic con il tasto destro del mouse sulle serie di richieste in cui si desidera aggiungere un membro e selezionare <b>Aggiungi membro</b>.</li> </ul>

Metodo	Descrizione
<p style="text-align: center;"><b>System i</b></p> <p>Comando di sistema ADDDPRSUBM</p>	<p>Utilizzare il comando ADDDPRSUBM per aggiungere un membro in una serie di richieste. Ad esempio, per aggiungere un membro della serie di richieste in una serie di richieste denominata SETHR nel qualificatore Apply AQHR:</p> <pre>ADDDPRSUBM APYQUAL(AQHR) SETNAME(SETHR) SRCTBL(HR/YTDTAX) TGTTBL(TGTHR/TGTTAX)</pre>

Per associare un'origine con una destinazione, specificare le seguenti informazioni relative alla vista o alla tabella registrata che si desidera utilizzare come origine:

- La vista o la tabella di origine e la vista o la tabella di destinazione (incluso un tablespace e l'indice per la tabella di destinazione).
- Il tipo di tabella di destinazione.
- Le colonne registrate dalla tabella di origine che si desidera replicare nella tabella di destinazione.

Quando si utilizza il Centro di replica per associare un'origine con una destinazione, le colonne LOB non vengono incluse automaticamente nell'associazione delle colonne. È necessario selezionare esplicitamente tali colonne.

- Le righe dalla tabella di origine che si desidera replicare nella tabella di destinazione (inclusa una clausola WHERE per specificare le righe).

#### Per associare l'origine scelta con una destinazione del DB2

Specificare le seguenti informazioni relative alla vista o alla tabella di destinazione:

- Lo schema.
- Il nome della vista o della tabella da utilizzare come destinazione.

**Impostazione predefinita:** il nome predefinito proviene dal profilo dell'oggetto di destinazione per il server di destinazione, se presente. Se questo profilo non è stato impostato, il valore predefinito è TG seguito dal nome della vista o della tabella di origine. Ad esempio, se il nome della tabella di origine è EMPLOYEE, per impostazione predefinita, il nome della tabella di destinazione sarà TGEMPLOYEE.

- Il tipo di tabella di destinazione

**Impostazione predefinita:** copia dell'utente

Se la tabella di destinazione specificata non esiste, essa verrà creata dagli strumenti di gestione o dal comando del sistema ADDDPRSUBM.

#### Per associare l'origine scelta con una destinazione relazionale non DB2

Specificare le seguenti informazioni relative alla tabella di destinazione:

- Lo schema del nickname
- Il nickname
- Lo schema remoto
- Il nome della tabella remota

**Impostazione predefinita:** il nome predefinito proviene dal profilo dell'oggetto di destinazione per il server di destinazione, se presente. Se questo profilo non è stato impostato, il valore predefinito è TG seguito dal nome della vista o della tabella di origine. Ad esempio, se il nome della tabella di origine è EMPLOYEE, per impostazione predefinita, il nome della tabella di destinazione sarà TGEMPLOYEE.

- Il tipo di tabella di destinazione

**Impostazione predefinita:** copia dell'utente

Quando si aggiunge un membro della serie di richieste, è possibile utilizzare il tipo di tabella di destinazione predefinito della copia utente oppure selezionare un altro tipo di tabella di destinazione per soddisfare le proprie esigenze di replica. Quando si aggiunge un membro della serie di richieste per una tabella di destinazione che non esiste, è possibile utilizzare le impostazioni predefinite oppure modificare le proprietà del membro per soddisfare le proprie esigenze di replica. È possibile selezionare il tipo di tabella di destinazione da utilizzare e successivamente impostare le proprietà relative al modo in cui il programma Apply replicherà i dati alla destinazione.

## Tipi di tabelle di destinazione

Il tipo di tabella di destinazione dipende dal modo in cui si desidera visualizzare i dati e dalla configurazione della replica. È possibile utilizzare una tabella esistente come destinazione oppure creare una nuova tabella.

### Restrizioni

- Gli attributi null delle colonne di destinazione post-immagine devono essere compatibili con gli attributi null delle colonne della vista o della tabella di origine. Utilizzare l'espressione SQL COALESCE per fornire la compatibilità con le colonne esistenti.
- Per le tabelle di origine nei database relazionali non DB2, è possibile definire solo i seguenti tipi di tabelle di destinazione:
  - Tabelle copia dell'utente
  - Tabelle delle istantanee
  - Tabelle CCD esterne
- I nomi di tutte le tabelle di destinazione relazionali non DB2 e degli indici devono seguire le convenzioni di denominazione degli indici e delle tabelle del DB2.
- **System i** Per le tabelle di origine su System i che utilizzano colonne RRN come colonne chiave, è possibile definire solo i seguenti tipi di tabelle di destinazione:
  - Tabelle delle istantanee
  - Tabelle CCD esterne
- **z/OS** Per le tabelle di origine in un sottosistema z/OS, lo schema di codifica per le tabelle CD e UOW deve essere identico, se il programma Apply unirà queste tabelle per soddisfare una clausola WHERE della serie di richieste per una tabella copia dell'utente.

### Tipi di destinazione

È possibile selezionare i seguenti tipi di tabelle di destinazione:

#### Copia dell'utente

Una tabella di destinazione di sola lettura che include solo le colonne definite nel membro della serie di richieste. Una tabella copia dell'utente può presentare la stessa struttura della tabella di origine oppure una serie secondaria di colonne di origine con o senza colonne calcolate o pre-immagini. La replica SQL presume che sia l'unica applicazione che scrive nelle tabelle di destinazione copia dell'utente. Le modifiche dirette nelle tabelle copia dell'utente da parte degli utenti finali o delle applicazioni possono essere sovrascritte dalla replica SQL e possono causare una mancata corrispondenza tra i dati nelle tabelle di origine e di

destinazione. Se è necessario aggiornare le tabelle di origine e di destinazione, si consiglia di utilizzare la replica di aggiornamento ovunque.

#### **Punto nel tempo**

Una tabella di destinazione di sola lettura che include le colonne definite nel membro della serie di richieste e una colonna timestamp. Una tabella di istantanee può presentare la stessa struttura della tabella di origine oppure una serie secondaria di colonne di origine con o senza colonne calcolate o pre-immagini.

#### **Aggregazione di base**

Una tabella di destinazione di sola lettura che utilizza le funzioni delle colonne SQL (ad esempio, SUM e AVG) per calcolare i riepiloghi dell'intero contenuto della tabella di origine.

Una tabella di aggregazione di base riepiloga il contenuto di una tabella di origine. Una tabella di aggregazione di base include anche un timestamp del momento in cui il programma Apply ha eseguito l'aggregazione. Utilizzare la tabella di aggregazione di base per tracciare lo stato di una tabella di origine su base regolare.

#### **Aggregazione di modifiche**

Una tabella di destinazione di sola lettura che utilizza le funzioni delle colonne SQL (ad esempio, SUM e AVG) per calcolare i riepiloghi dell'intero contenuto delle modifiche recenti apportate alla tabella di origine, che vengono memorizzate nella tabella CD o in una tabella CCD interna.

Una tabella di aggregazione di modifiche riepiloga il contenuto di una tabella CD o di una tabella CCD interna, piuttosto che della tabella di origine. Una tabella di aggregazione di modifiche include inoltre due timestamp per indicare l'intervallo di tempo in cui sono state catturate le modifiche (scritte nella tabella CD o CCD). Utilizzare la tabella di aggregazione di modifiche per tracciare le modifiche (operazioni UPDATE, INSERT e DELETE) eseguite tra i cicli di replica.

#### **CCD (consistent-change data)**

Una tabella di destinazione di sola lettura con colonne aggiuntive per le informazioni relative al controllo della replica. Queste colonne includono un numero di record di registrazione (o numero di record del diario), un indicatore che indica se la tabella di origine è stata modificata mediante un'istruzione SQL INSERT, DELETE o UPDATE, infine il numero di record della registrazione e il timestamp dell'istruzione commit associata all'istruzione insert, delete o update. Facoltativamente, è possibile includere le colonne pre-immagine e le colonne della tabella UOW.

#### **Replica**

Una tabella di destinazione di lettura/scrittura per la replica di aggiornamento ovunque. Una tabella della replica è l'unico tipo di tabella di destinazione che i programmi applicativi e gli utenti possono aggiornare direttamente. Una tabella della replica riceve le modifiche dalle tabelle principale e dai programmi applicativi locali o dagli utenti. Le tabelle della replica possono presentare la stessa struttura della tabella di origine oppure una serie secondaria di colonne di origine, ma non includono ulteriori colonne di controllo della replica (ad esempio, i timestamp). Le tabelle della replica sono supportate solo per i database DB2.

Di seguito viene illustrato come utilizzare ciascun tipo di destinazione e come impostare le proprietà delle tabelle di destinazione per soddisfare le esigenze della replica:



## **Tabelle di destinazione di sola lettura**

A seconda del modo in cui si desidera che i dati di origine vengano visualizzati nella destinazione, è possibile definire che le tabelle di destinazione di sola lettura contengano una copia della vista o della tabella di origine, una cronologia delle modifiche o un riepilogo calcolato.

I seguenti argomenti forniscono ulteriori dettagli su questi tipi di destinazioni di sola lettura.

### **Destinazioni copia utente e punto nel tempo:**

Per impostazione predefinita, quando viene definito un membro di impostazione richieste verrà creata una tabella copia utente come tipo di destinazione. Selezionare punto nel tempo come tipo di destinazione per tenere traccia dell'ora alla quale sono state apportate le modifiche alla destinazione.

#### **Copia dell'utente**

Utilizzare il tipo predefinito se si desidera che la tabella di destinazione corrisponda alla tabella origine nel momento in cui viene effettuata la copia. La tabella copia utente non contiene nessuna ulteriore colonna controllo replica, ma può contenere una serie secondaria di righe o colonne nella tabella origine o ulteriori colonne che non vengono replicate.

#### **Punto nel tempo**

Selezionare punto nel tempo come tipo di destinazione se si desidera tenere traccia dell'ora alla quale vengono apportate le modifiche alla destinazione. Una destinazione punto nel tempo contiene gli stessi dati della propria tabella origine, con un'ulteriore colonna data/ora aggiunta affinché si possa conoscere quando il programma Apply ha assegnato ogni riga alla destinazione. La colonna data/ora è originariamente vuota. Le tabelle orarie possono contenere una serie secondaria di righe o colonne nella tabella di origine o ulteriori colonne che non vengono replicate.

**Limitazione:** DB2 impedisce che i valori siano inseriti in colonne di una tabella DB2 che vengono definite AS IDENTITY GENERATED ALWAYS. Per evitare la restrizione, è possibile:

- Creare la tabella destinazione senza la IDENTITY CLAUSE
- Creare la tabella di destinazione con la colonna AS IDENTITY GENERATED BY DEFAULT

### **Destinazioni aggregati di modifica o aggregati di base:**

È possibile creare tabelle di destinazione che contengono riepiloghi dell'interno contenuto delle tabelle di origine o delle modifiche più recenti eseguite sui dati della tabella di origine.

Relativamente ai tipi tabella di destinazione aggregati, è possibile definire colonne di destinazione utilizzando funzioni di colonna SQL aggregate, come COUNT, SUM, MIN, MAX e AVG. Tali colonne non contengono i dati di origine, ma contengono i valori calcolati della funzione SQL che si definisce. Il programma Apply non crea aggregazioni durante l'aggiornamento completo; le righe vengono aggiunte nel tempo quando il programma Apply elabora la serie. Un vantaggio nell'uso di una tabella di aggregati consiste nella possibilità di replica da parte della replica SQL delle sole informazioni di riepilogo anziché di ogni singola riga, salvando quindi sia la larghezza di banda di rete che spazio nella tabella di destinazione.

## Destinazioni aggregati di base

Utilizzare una tabella di destinazione aggregati di base per tenere traccia di una tabella di origine durante ogni ciclo di replica. Relativamente ad una tabella di destinazione aggregati di base, il programma Apply esegue l'aggregazione (legge ed esegue calcoli) dalla tabella di origine. Una tabella aggregati di base include inoltre data/ora di quando il programma Apply ha eseguito l'aggregazione.

Se una tabella di origine registrata ha solo una tabella aggregata di base come relativa destinazione, non è necessario catturare le modifiche per la tabella di origine.

**Esempio:** si supponga che si desidera conoscere il numero medio di clienti che si hanno ogni settimana. Se la tabella di origine dispone di una riga per ciascun cliente, il programma Apply è in grado di calcolare la somma del numero di righe nella tabella origine su base settimanale, nonché di memorizzare i risultati in una tabella di aggregati di base. Se si esegue l'aggregazione ogni settimana, la tabella di destinazione avrà 52 voci che mostrano il numero di clienti avuto per ogni settimana dell'anno.

## Destinazioni aggregati di modifica

Utilizzare una tabella di destinazione aggregati di modifica per tenere traccia delle modifiche (operazioni UPDATE, INSERT e DELETE) eseguite tra cicli di replica nella tabella di origine. Relativamente ad una tabella di destinazione aggregati di modifica, il programma Apply esegue l'aggregazione (legge ed esegue calcoli) dalla tabella CD o CCD interna. Una tabella aggregati di modifica include inoltre due data/ora per contrassegnare l'intervallo di tempo relativo a quando il programma Capture ha inserito le modifiche nella tabella CD o CCD.

**Esempio:** si supponga che si desideri sapere quanti nuovi clienti sono stati acquisiti ogni settimana (INSERT) e quanti clienti esistenti sono stati persi (DELETE). È possibile effettuare il conteggio del numero di righe inserite nella tabella CD su base settimanale, nonché di memorizzare tale numero in una tabella di aggregati di modifica.

**Importante:** Se la tabella di origine relativa ad un membro di serie di sottoscrizioni è registrata per la replica solo di aggiornamento completo, non è possibile avere una tabella di destinazione aggregati, che richiede una tabella CD o CCD all'origine.

## Destinazioni CCD:

Le tabelle CCD (Consistent-change-data) forniscono dati transazionali con commit che possono essere letti e utilizzati da altre applicazioni, come ad esempio WebSphere DataStage. È inoltre possibile utilizzare una tabella CCD per il controllo dei dati di origine o per mantenere una cronologia delle modalità di utilizzo dei dati.

Ad esempio, è possibile tenere traccia dei confronti dei dati precedenti e successivi, di quando si sono verificate le modifiche, e dell'ID utente che ha aggiornato la tabella di origine.

Per definire una tabella di destinazione di sola lettura che conservi la cronologia della tabella di origine, definire la tabella CCD di destinazione in modo che includa i seguenti attributi:

### Non concentrata

Per conservare un record di tutte le modifiche di origine, definire la tabella CCD in modo che sia non concentrata. In tal modo, essa memorizzerà una riga per ciascuna modifica che viene eseguita. Poiché le tabelle non concentrate contengono diverse righe con lo stesso valore chiave, non definire un indice univoco. Una tabella CCD non concentrata contiene una riga per ciascuna operazione UPDATE, INSERT o DELETE, in tal modo, conserva una cronologia delle operazioni eseguite nella tabella di origine. Se si catturano le operazioni UPDATE come operazioni INSERT e DELETE (per le colonne della chiave di partizione), la tabella CCD conterrà due righe per ciascun aggiornamento, una riga per DELETE e una riga per INSERT.

### Completa o incompleta

È possibile scegliere se si desidera che la tabella CCD sia completa o incompleta. Poiché le tabelle CCD incomplete inizialmente non contengono una serie completa di righe di origine, creare una tabella CCD incompleta per conservare la cronologia degli aggiornamenti in una tabella di origine (gli aggiornamenti da quando il programma Apply ha iniziato ad inserire i dati nella tabella CCD).

### Includi colonne UOW

Per migliorare la funzione di controllo, includere le colonne aggiuntive dalla tabella UOW. Per una maggiore identificazione orientata verso l'utente, le colonne per l'ID correlazione di DB2 per z/OS e l'ID autorizzazione principale o il nome del lavoro di System i e il profilo dell'utente sono disponibili nella tabella UOW.

Per definizione, una tabella CCD include sempre le seguenti colonne in aggiunta alle colonne replicate dalla tabella di origine:

Colonna	Descrizione
IBMSNAP_INTENTSEQ	<b>Tipo di dati:</b> CHAR(10) PER DATI BIT; <b>Impostabile su null:</b> No  Un numero di sequenza che identifica una modifica in modo univoco. Tale valore è ascendente in una transazione.  <b>z/OS</b> Il numero della sequenza della registrazione (LRSN o RBA) di ciascun aggiornamento, eliminazione e inserimento.
IBMSNAP_OPERATION	<b>Tipo di dati:</b> CHAR(1); <b>Impostabile su null:</b> no  Un flag che indica il tipo di operazione: I (INSERT), U (UPDATE) o D (DELETE).
IBMSNAP_COMMITSEQ	<b>Tipo di dati:</b> CHAR(10) PER DATI BIT; <b>Impostabile su null:</b> No  Un numero di sequenza per ciascuna riga all'interno di una transazione.  <b>z/OS</b> Il numero della sequenza della registrazione (LRSN or RBA) del record di commit dell'origine.

Colonna	Descrizione
IBMSNAP_LOGMARKER	<b>Tipo di dati:</b> TIMESTAMP; <b>Impostabile su null:</b> no  L'ora in cui è stato eseguito il commit dei dati.

Quando si crea una tabella CCD incompleta (COMPLETE=N) con il programma della riga comandi ASNCLP o con il Centro di replica, è possibile specificare ulteriori colonne di controllo. Nella seguente tabella vengono descritte queste colonne:

Colonna	Descrizione
IBMSNAP_AUTHID	<b>Tipo di dati:</b> VARCHAR(30); <b>Impostabile su null:</b> sì  L'ID utente che ha aggiornato la tabella di origine.  <b>z/OS</b> Questa colonna è l'ID di autorizzazione primario.
IBMSNAP_AUTHTKN	<b>Tipo di dati:</b> VARCHAR(30); <b>Impostabile su null:</b> sì  Il token di autorizzazione associato alla transazione.  <b>z/OS</b> L'ID di correlazione (normalmente un nome lavoro) che ha eseguito l'aggiornamento dell'origine.
IBMSNAP_PLANID	<b>Tipo di dati:</b> VARCHAR(8); <b>Impostabile su null:</b> Sì  <b>z/OS</b> Il nome del piano associato alla transazione. Questa colonna sarà nulla per DB2 per Linux, UNIX e Windows.
IBMSNAP_UOWID	<b>Tipo di dati:</b> CHAR(10) FOR BIT DATA; <b>Impostabile su null:</b> Sì  L'identificatore UOW (unit-of-work) dal record di registrazione per una riga.  <b>z/OS</b> L'identificatore unit-of-work, a volte chiamato URID (unit-of-recovery ID) della transazione.

#### Destinazioni CCD interne:

Se le modifiche sono frequenti in una tabella di origine, è possibile creare una tabella CCD interna per riepilogare le modifiche di cui è stato eseguito il commit, verificatesi nell'origine dall'ultimo ciclo di Apply.

Poiché la tabella CD è in continuo cambiamento quando il programma Capture aggiunge le modifiche dalla registrazione, la cache locale delle modifiche dell'origine nella tabella CCD rappresenta un'origine più stabile per le destinazioni.

Quando la tabella di origine originale viene aggiornata, il programma Capture legge le modifiche frequenti nella registrazione di origine e le aggiunge nella tabella CD di origine. Da tale tabella CD, un programma Apply legge le modifiche nella tabella CD e inserisce i dati nella tabella CCD interna. È possibile definire la tabella CCD interna in modo che contenga solo la modifica più recente per ciascuna riga nella tabella CD verificatasi durante l'ultimo ciclo. Pertanto, la tabella CCD è statica tra i cicli di Apply (per il programma Apply che replica dalla tabella CD nella tabella CCD) e rappresenta un'origine più stabile per le destinazioni. Concentrando le modifiche dall'origine, è possibile migliorare le prestazioni generali della replica, poiché non viene eseguita la replica di molti aggiornamenti per la stessa riga nella tabella di destinazione.

Poiché il programma Capture aggiunge costantemente nuove modifiche nella tabella CD, un secondo programma Apply legge le modifiche dalla tabella CCD interna, invece della tabella CD, in tal modo, non replica le diverse modifiche in diverse destinazioni e può mantenere le destinazioni sincronizzate tra loro. Il secondo programma Apply utilizza la tabella di origine originale per gli aggiornamenti completi e utilizza la tabella CCD interna per la replica di modifica/cattura.

**Importante per gli aggiornamenti:** Se si definisce una tabella CCD interna, il programma Apply la ignora quando elabora una serie di richieste con una replica come destinazione e applica le modifiche alla replica dalla tabella CD di origine principale.

### Consigli

- Definire un membro della serie di richieste tra la tabella di origine e la tabella CCD interna prima di definire altri membri della serie di richieste tra la tabella di origine e le altre tabelle di destinazione. In tal modo, il programma Apply utilizzerà la tabella CCD interna invece della tabella CD per replicare le modifiche dalla tabella di origine. Se si definiscono altri membri della serie di richieste e si inizia la replica utilizzando tali membri prima di definire la tabella CCD interna per la tabella di origine, può essere necessario eseguire un aggiornamento completo per tutte le destinazioni della tabella di origine.
- Associare tutte le tabelle CCD interne in una serie di richieste per garantire che tutte le tabelle di destinazione per il database di origine siano sincronizzate tra loro.
- Anche se si desidera applicare alle altre destinazioni solo una serie secondaria delle colonne di origine che cambiano frequentemente, utilizzare l'impostazione predefinita in base alla quale tutte le colonne di origine registrate vengono replicate nella tabella CCD interna. In tal modo, è possibile utilizzare la tabella CCD interna come origine per le future tabelle di destinazione che potrebbero richiedere i dati da altre colonne registrate nella tabella di origine originale. Solo le colonne contenute nella tabella CCD interna saranno disponibili per la replica di modifica/cattura per le destinazioni future.

### Attributi delle tabelle CCD interne

Le tabelle CCD interne vengono utilizzate come un'origine implicita per la replica. Non è possibile definirle esplicitamente come origine della replica. Quando si

aggiunge un membro della serie di richieste, si associa la tabella di origine originale (non la tabella CCD interna) con la tabella di destinazione. Una tabella CCD interna presenta i seguenti attributi:

#### **Interna**

La tabella CCD rappresenta un'alternativa per la tabella CD di origine. Le informazioni relative alla tabella CCD interna vengono memorizzate nella stessa riga della relativa tabella di origine nella tabella IBMSNAP\_REGISTER. Una tabella CCD interna non presenta una propria riga nella tabella register. Il programma Apply replica automaticamente le modifiche da una tabella CCD interna, se presente, piuttosto che dalle tabelle CD. Per ciascuna origine della replica può esistere una sola tabella CCD interna.

**Limitazione:** La tabella utente non include le colonne calcolate, pertanto non inserire colonne calcolate nelle richieste CCD.

**Locale** La tabella CCD è contenuta nello stesso database della tabella di origine.

#### **Incompleta**

Poiché il programma Apply utilizza la tabella di origine originale per gli aggiornamenti completi e non la tabella CCD interna, quest'ultima è incompleta perché la destinazione successiva conterrà già una copia iniziale di tutte le righe di origine.

#### **Concentrata**

La tabella CCD interna è concentrata, ciò significa che contiene una riga per ciascun valore chiave. In tal modo, il programma Apply applica la modifica più recente per ciascuna riga nella tabella CCD, invece di applicare una riga per ciascuna modifica.

#### **Nessuna colonna UOW**

Le tabelle CCD interne non supportano colonne aggiuntive della tabella UOW. Non è possibile utilizzare una tabella CCD interna se è già stata definita una tabella CCD di destinazione contenente le colonne UOW.

### **Definizione di livelli intermedi in una configurazione con livelli multipli**

Il modello di replica di base prevede due modelli, con una singola origine e una o più destinazioni. È anche possibile impostare le configurazioni con tre o più livelli.

#### **Restrizioni**

Il livello intermedio nelle configurazioni a livelli multipli deve essere una tabella DB2.

#### **Informazioni su questa attività**

Una configurazione con livelli multipli presenta una tabella di origine e una tabella di destinazione e quest'ultima funge da origine per le altre tabelle di destinazione.

Un motivo per configurare un ambiente di replica con livelli multipli consiste nel spostare il sovraccarico della distribuzione dal sistema di origine a un secondo sistema. È anche possibile evitare molte delle connessioni del database sul sistema di origine, spostando i costi delle connessioni al secondo livello. Inoltre, poiché è possibile raccogliere le modifiche dal livello 1 nelle tabelle CCD del livello 2, è possibile controllare la frequenza di replica delle modifiche in ciascun livello e ridurre il numero di modifiche replicate alla destinazione (livello 3).

Ad esempio, in un modello a tre livelli, il primo livello (livello 1) è il database di origine, il secondo livello (livello 2) è la destinazione per il livello 1. Il livello 2 è anche un'origine per un terzo livello delle destinazioni (livello 3) e può distribuire le modifiche ad uno o più database del livello 3. Quando si dispone di più di due livelli nella configurazione della replica, i livelli intermedi, che fungono da origini e destinazioni, sono le tabelle CCD.

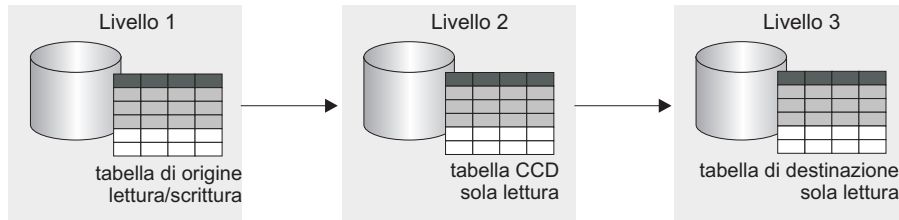


Figura 6. Modello di replica a tre livelli. È possibile replicare i dati da una tabella di origine in una tabella di destinazione e da questa tabella in un'altra tabella di destinazione.

Questa procedura si applica anche per le tabelle della replica. Generalmente, le tabelle CCD vengono utilizzate per le replica di sola lettura, ma le tabelle della replica vengono utilizzate per le repliche di aggiornamento ovunque.

### Procedure

Per configurare una replica con livelli multipli, in modo che la tabella di destinazione funga da origine per le tabelle successive:

1. Registrare la tabella di origine (livello 1) per la replica. Il programma Capture per questa origine cattura le modifiche verificatesi al livello 1 e le memorizza nella tabella CD del livello 1.
2. Creare una serie di richieste tra il server di origine e il server di destinazione (per il livello 2). Il programma Apply per questa serie di richieste applica le modifiche dal livello 1 alla tabella CCD del livello 2.
3. Definire un membro della serie di richieste che associa la tabella di origine (livello 1) e una tabella di destinazione CCD (livello 2).

Quando si definisce la tabella di destinazione per questo membro, selezionare come tabella CCD la tabella di destinazione con i seguenti attributi:

#### Origine esterna registrata

È necessario definire l'origine come una tabella di destinazione esterna e registrare la tabella, in modo che funga da origine per il livello successivo. Come le altre origini registrate, una tabella CCD esterna presenta una propria riga nella tabella IBMSNAP\_REGISTER. Le tabelle CCD esterne, che fungono anche da origini, possono essere popolate solo da una singola tabella di origine.

È necessario registrare tutte le tabelle CCD esterne in una serie di richieste mediante lo stesso schema Capture.

È possibile replicare su una tabella CCD esterna senza unire la tabella dei dati di modifica (CD) e la tabella IBMSNAP\_UOW. Il nuovo tipo di tabella è specificato con un valore 9 nella colonna TARGET\_STRUCTURE della tabella IBMSNAP\_SUBS\_MEMBER. Sebbene la tabella CCD tipo 9 includa la colonna IBMSNAP\_LOGMARKER, il programma Apply non richiede una congiunzione della tabella CD e della tabella IBMSNAP\_UOW per ottenere la data/ora di commit dell'origine per quella colonna. Al contrario, il programma Apply genera lo stesso valore nella colonna

IBMSNAP\_LOGMARKER per tutte le righe nello stesso ciclo. Il nuovo tipo di tabella CCD ha la stessa struttura della tabella CCD di tipo 3. La tabella contiene quattro colonne IBM® obbligatorie oltre alle colonne utente:

```
IBMSNAP_COMMITSEQ  
IBMSNAP_INTENTSEQ  
IBMSNAP_OPERATION  
IBMSNAP_LOGMARKER  
user_columns
```

Questo tipo di tabella di destinazione può essere registrata come tabella di origine per una configurazione di replica a tre livelli.

**Attenzione:** Per le tabelle CCD tipo 9, il fattore di blocco dati (MAX\_SYNCH\_MINUTES nella tabella di controllo IBMSNAP\_SUBS\_SET) non deve avere alcuna impostazione (NULL).

### Completa

È necessario utilizzare una tabella CCD completa, perché il programma Apply utilizzerà questa tabella per eseguire l'aggiornamento completo e la replica di modifica/cattura per il livello successivo.

### Concentrata

Utilizzare una tabella CCD concentrata, ovvero una tabella che contiene una riga per ciascun valore chiave, per garantire che solo le modifiche più recenti verranno replicate nel livello successivo. Il programma Apply applica la modifica più recente per ciascuna riga nella tabella CCD, invece di applicare una riga per ciascuna modifica. Poiché le tabelle concentrate richiedono valori chiave univoci per ciascuna riga, è necessario definire un indice univoco.

4. Poiché la tabella CCD è registrata, creare le tabelle di controllo Capture nel database di livello intermedio, se non esistono già.
5. Creare una serie di richieste tra il server del livello 2 contenente la tabella CCD registrata e il successivo server di destinazione (per il livello 3). Il programma Apply per questa serie applica le modifiche dalla tabella CCD nelle tabelle di destinazione nel livello successivo. Il programma Apply utilizza la tabella CCD per l'aggiornamento completo e la replica di modifica/capture. Generalmente, l'utente utilizza un qualificatore Apply diverso da quello utilizzato per inserire i dati nella tabella CCD, ma è possibile utilizzare lo stesso qualificatore.
6. Definire un membro della serie di richieste associando la tabella di origine CCD (livello 2) e la successiva tabella di destinazione (livello 3). È possibile configurare diversi membri con tabelle di destinazione che richiedono questa tabella di origine CCD. Se questo è il livello finale nella configurazione con livelli multipli, la tabella di destinazione può essere di qualsiasi tipo. Tuttavia, se si desidera avere più di tre livelli, definire la tabella di destinazione del livello 3 come specificato nel passo 3 e ripetere i passi 4 e 5 per aggiungere altri livelli.

**Importante:** Se nella tabella CCD esterna si verifica un aggiornamento completo, (livello intermedio), i programmi Apply per tutti i livelli successivi che utilizzano la tabella CCD esterna come origine eseguiranno aggiornamenti completi. Si tratta di un *aggiornamento completo a cascata*.



## **Definizione delle destinazioni di lettura/scrittura (aggiornamento ovunque)**

Nella replica di aggiornamento ovunque, le modifiche apportate alla tabella di origine principale vengono replicate nelle tabelle di destinazione dipendenti e le modifiche apportate nelle tabelle della replica possono essere replicate nella tabella di origine principale.

### **Prima di iniziare**

- È necessario utilizzare i vincoli di integrità referenziale dichiarativi perché nessun programma applicativo singolo aggiorna le tabelle principale e della replica. Le violazioni di integrità referenziale non possono essere individuate nella logica dell'applicazione.
- È necessario includere tutti i vincoli referenziali esistenti tra le tabelle principali nelle tabelle della replica per evitare violazioni di integrità referenziale. Se vengono omessi alcuni vincoli referenziali, un aggiornamento apportato ad una tabella della replica può causare una violazione di integrità referenziale quando viene replicato nella tabella principale. Gli strumenti di gestione non copiano le definizioni dei vincoli referenziali da una tabella di origine nelle tabelle di destinazione, né possono creare nuovi vincoli.
- Per ignorare la verifica di integrità referenziale durante l'aggiornamento completo, è necessario utilizzare la routine di chiusura ASNLOAD.

### **Restrizioni**

- I tipi di tabella di destinazione della replica non sono supportati in una configurazione remota del giornale.
- Non è possibile utilizzare le tabelle CCD come origini o destinazioni nella replica di aggiornamento ovunque.
- Per consentire alle colonne del tipo di dati LOB di partecipare alla replica di aggiornamento ovunque, è necessario impostare su 0 il valore di CONFLICT\_LEVEL nella tabella register.
- I database non DB2 non possono presentare i tipi di tabelle di destinazione della replica, pertanto, non possono partecipare alla replica di aggiornamento ovunque.

### **Informazioni su questa attività**

Nella replica di aggiornamento ovunque, la tabella principale e le relative repliche sono tabelle di lettura/scrittura che fungono da origini e da destinazioni.

### **Procedure**

Per impostare una configurazione di aggiornamento ovunque tra una tabella principale e una o più tabelle della replica (dove ciascuna tabella della replica è contenuta in un database separato):

1. Creare le tabelle di controllo Capture in ciascun database che conterrà una tabella della replica, se non esistono già.
2. Registrare la tabella di origine (la tabella principale) per la replica.
3. Creare una serie di richieste tra il database principale e il database di destinazione che conterrà una o più repliche.

Se tutte le tabelle della replica sono contenute nello stesso database e tutte le tabelle principali sono contenute in un altro database, è necessaria una sola

serie di richieste. Se le tabelle della replica sono contenute in più database, è necessario un numero di serie di richieste pari al numero di database della replica.

4. Definire un membro della serie di richieste per ciascuna associazione tra ciascuna tabella principale e la relativa tabella della replica associata.

In questa configurazione, è presente un solo programma Apply, che generalmente viene eseguito sul server contenente le tabelle della replica. Il programma Apply per questa serie estrae le modifiche dalla tabella CD principale e le applica alle tabelle della replica. Inoltre, il programma Apply estrae le modifiche dalla tabella CD della tabella della replica e le applica alla tabella principale.

**Importante:** Poiché la tabella principale e le tabelle della replica nelle configurazioni di aggiornamento ovunque replicano i dati reciprocamente, le tabelle di destinazione della replica devono contenere le stesse colonne della tabella di origine. È possibile creare una destinazione della replica contenente una serie secondaria delle colonne nella tabella principale solo se le colonne mancanti sono definite come impostabili su null o NOT NULL WITH DEFAULT nella sede principale, ma è necessario non aggiungere nuove colonne o rinominare le colonne nella replica.

5. Definire le proprietà di origine per la tabella della replica. Quando si crea un membro della serie di richieste con una tabella della replica, la replica SQL registra automaticamente la tabella della replica come un'origine della replica. Poiché le tabelle di destinazione della replica fungono da origini, esse presentano delle proprietà che è possibile impostare, oltre alle proprietà comuni delle tabelle di destinazione, che determinano il modo in cui il programma Capture gestisce le modifiche nella replica. Tuttavia, due proprietà vengono ereditate dalla tabella principale e non possono essere modificate per la tabella della replica: il livello di rilevamento dei conflitti e gli aggiornamenti completi sono disabilitati. Il programma Capture per questa origine cattura le modifiche nella tabella della replica e le memorizza nella tabella CD della replica.

**Importante:** Anche se la tabella principale e la replica fungono da origini e destinazioni, la copia dell'aggiornamento completo viene eseguita solo dalla tabella principale nella replica, non dalla replica nella tabella principale.

Per evitare i conflitti, è necessario che la chiave di destinazione per le tabelle della replica sia identica alla chiave principale della tabella di origine principale o dell'indice univoco. Poiché la tabella principale può aggiornare le repliche e le repliche possono aggiornare la tabella principale, è possibile che si verifichino dei conflitti se viene eseguito un aggiornamento in una riga della tabella principale e un aggiornamento diverso viene eseguito nella stessa riga di una o più tabelle della replica tra i cicli di Apply (in modo che le modifiche siano presenti nella tabella CD principale e nella tabella CD della replica). La tabella della replica eredita il livello di rilevamento dei conflitti dalla vista o dalla tabella di origine principale. Si consiglia di progettare l'applicazione in modo non si verifichi mai un conflitto quando i dati vengono replicati dalla tabella principale in tutte le tabelle della replica. Quando è stata registrata l'origine principale, l'utente ha scelto tra tre livelli di rilevamento dei conflitti.

Se sono stati definiti dei vincoli di integrità referenziale per la tabella di origine, è necessario definire gli stessi vincoli di integrità referenziale per la tabella della replica per impedire le violazioni di integrità. Se si verifica una violazione di integrità referenziale, il ciclo di richieste viene automaticamente rieseguito.

## Utilizzo di una tabella esistente come tabella di destinazione

È possibile definire un membro della serie di richieste che includa una tabella di destinazione esistente definita al di fuori della replica SQL.

Le tabelle di destinazione definite dall'utente possono essere qualsiasi tipo di tabella di destinazione valido per la replica (copia dell'utente, istantanea, aggregazione di modifiche o di base, CCD o replica), se la struttura della tabella è valida. Ad esempio, una tabella di istantanee definita dall'utente deve includere una colonna di tipo `TIMESTAMP` denominata `IBMSNAP_LOGMARKER`.

### Requisiti

- Se la definizione del membro della serie di richieste contiene un numero di colonne minore rispetto alle colonne contenute nella tabella di destinazione esistente, le colonne della tabella di destinazione che non sono implicate nella replica devono consentire valori null o devono essere definite come `NOT NULL WITH DEFAULT`.
- Per le tabelle CCD concentrate, istantanee, copia dell'utente e replica è necessario definire un indice univoco. Quando si definisce il membro della serie di richieste mediante la tabella di destinazione esistente, è possibile utilizzare l'indice univoco esistente oppure specificarne uno nuovo.

### Restrizioni

- La definizione del membro della serie di richieste non può contenere un numero di colonne maggiore di quelle contenute nella tabella di destinazione esistente.
- Se si utilizza il Centro di replica, non è possibile aggiungere una colonna in un membro della serie di richieste, se tale colonna non esiste già nella tabella di destinazione.

La replica verifica la presenza di incoerenze tra la tabella di destinazione esistente e la definizione del membro della serie di richieste.

**Importante per i livelli multipli:** se si desidera impostare una configurazione con livelli multipli con una tabella di origine come livello 1, una tabella CCD come livello 2 e una tabella esistente come livello 3, definire la tabella CCD in modo che corrisponda agli attributi specificati per la tabella di destinazione esistente durante la definizione del membro della serie di richieste tra il livello 1 e il livello 2. Quindi, definire un membro della serie di richieste per la tabella di destinazione esistente in cui la tabella CCD è la tabella di origine.

## Proprietà delle colonne per tutti i tipi di tabella di destinazione

È possibile impostare le proprietà durante la creazione di una tabella di destinazione, indipendentemente dal tipo, in base all'ambiente di replica desiderato.

Nei seguenti argomenti vengono illustrate le caratteristiche comuni che è possibile definire per il modo in cui i dati di origine vengono associati alle tabelle di destinazione.

### Replica di una serie secondaria di colonne di origine

Per impostazione predefinita, la tabella di destinazione contiene tutte le colonne di origine registrate, tranne le colonne LOB. È possibile che l'utente non voglia replicare tutte le colonne o che la tabella di destinazione non supporti tutti i tipi di dati definiti nell'origine.

In questo caso, selezionare solo le colonne di origine che si desidera replicare nella tabella di destinazione. Le colonne registrate nella tabella di origine che non vengono selezionate sono disponibili per gli altri membri della serie di richieste, ma non vengono incluse per l'associazione corrente tra origine e destinazione.

È anche possibile aggiungere le colonne calcolate in una tabella di destinazione. Queste colonne possono essere definite dalle funzioni scalari SQL, ad esempio SUBSTR oppure possono essere colonne derivate, ad esempio la divisione del valore della colonna A per il valore della colonna B (colA/colB). Queste colonne calcolate possono fare riferimento a qualsiasi colonna dalla tabella di origine.

### **Replica di una serie secondaria delle righe di origine**

Per impostazione predefinita, la tabella di destinazione contiene tutte le righe della tabella di origine. È possibile che l'utente non voglia replicare tutte le righe o che voglia replicare le righe contenenti diversi tipi di dati in diverse tabelle di destinazione.

È possibile definire una serie secondaria di righe (orizzontale) nel membro della serie di richieste contenente le righe che soddisfano una determinata condizione (una clausola SQL WHERE).

Il predicato SQL può contenere identificativi delimitati o ordinari. Per ulteriori informazioni sulle clausole WHERE, consultare *DB2 SQL Reference*.

Ad esempio, è possibile definire una clausola WHERE per replicare tutte le righe per una divisione di un'azienda. Oppure è possibile definire una clausola WHERE in un membro della serie di richieste per replicare tutte le colonne LOB (più la colonna della chiave principale) in una tabella di destinazione e una clausola WHERE in un altro membro della serie di richieste per replicare tutte le altre colonne in una tabella di destinazione separata. Quindi, il database di destinazione può contenere tutti i dati dalla tabella di origine, ma denormalizzare la tabella di origine nel database di destinazione per regolare le prestazioni delle query per un data warehouse.

### **Restrizioni dei predicati delle righe**

- Non digitare WHERE nella clausola, è implicito. Digitare WHERE nella clausola solo per le istruzioni subselect.
- Non terminare la clausola con un punto e virgola (;).
- Se la clausola WHERE contiene l'espressione booleana OR, inserire il predicato tra parentesi, ad esempio (COL1=X OR COL2=Y).
- Se la tabella di destinazione è una tabella di aggregazione di modifiche e contiene delle colonne pre-immagine, è necessario inserire le colonne pre-immagine in una clausola GROUP BY.

### **Esempi**

Nei seguenti esempi vengono illustrate le clausole WHERE che è possibile utilizzare per filtrare le righe della tabella di destinazione. Questi esempi sono generici e rappresentano un modello di riferimento.

#### **Clausola WHERE per specificare le righe con valori specifici**

Per copiare solo le righe contenenti un valore specifico, ad esempio MGR per gli impiegati con ruolo di responsabile, utilizzare una clausola WHERE come la seguente:

```
EMPLOYEE = 'MGR'
```

### **Clausola WHERE per specificare le righe con un intervallo di valori**

Per copiare solo le righe contenute in un intervallo, ad esempio il numero di impiegati tra 5000 e 7000 nella tabella di destinazione, utilizzare una clausola WHERE come la seguente:

```
EMPID BETWEEN 5000 AND 7000
```

### **Associazione delle colonne di origine con le colonne di destinazione**

Per impostazione predefinita, i nomi delle colonne in una tabella di destinazione creata dalla replica SQL corrispondono ai nomi delle colonne nella tabella di origine. È possibile modificare la lunghezza dei nomi e dei dati della maggior parte delle colonne di destinazione ed associarle alle colonne di origine.

È possibile modificare i nomi di tutte le colonne nelle tabelle di destinazione, ad eccezione delle colonne di controllo delle repliche (che iniziano con IBMSNAP o IBMQSQ). Se esiste la tabella di destinazione, il Centro di replica associa le colonne per nome.

Le colonne della tabella di destinazione possono presentare lunghezze diverse rispetto alle colonne di origine. Se la colonna di destinazione è più corta della colonna di origine, è possibile utilizzare un'espressione nel membro della serie di richieste per associare i caratteri dalla colonna più lunga alla colonna più corta oppure registrare una vista che includa l'espressione. Ad esempio, se la colonna di origine è char(12) e la colonna di destinazione è char(4), è possibile utilizzare la seguente espressione per troncatura i valori da COL1 durante la replica:

```
substr(col1, 1,4)
```

Se il nome della colonna di destinazione è più lungo, inserire degli spazi nel nome della colonna di destinazione.

**Nota:** Esistono delle restrizioni per associare le colonne LONG VARCHAR in DB2 per Linux, UNIX e Windows con DB2 per z/OS e DB2 per i5/OS.

### **Utilizzo del Centro di replica**

Quando si crea una tabella di destinazione mediante il Centro di replica, è possibile rinominare le colonne nella destinazione indipendentemente dal tipo di tabella di destinazione. Inoltre, è possibile modificare gli attributi della colonna (tipo di dati, lunghezza, scala, precisione e se è impostabile su null) dove gli attributi sono compatibili.

Non è possibile utilizzare il Centro di replica per rinominare le colonne di tabelle di destinazione esistenti. Se le colonne di origine e di destinazione non corrispondono, è possibile utilizzare il Centro di replica per associare le colonne dall'origine alla destinazione oppure creare una vista della tabella di destinazione contenente una corrispondenza con i nomi delle colonne di origine.

### **Associazione con tabelle relazionali non DB2**

Se si associa una tabella del DB2 con una tabella relazionale non DB2 con un nickname esistente per la tabella relazionale non DB2, i tipi di dati di alcune colonne potrebbero non essere compatibili. Se i tipi di dati delle colonne di origine non sono compatibili con i tipi di dati nelle colonne di destinazione, è possibile modificare il tipo di dati nella destinazione per renderli compatibili con l'origine:

- È possibile aggiungere colonne calcolate per regolare i tipi di dati dall'origine in modo che corrispondano al tipo di dati richiesto per la destinazione.
- È possibile modificare il nickname per una tabella di destinazione relazionale non DB2 per modificare le conversioni del tipo di dati.

**Esempio:** si desidera replicare i dati da una tabella di origine del DB2 con una colonna DB2 del tipo di dati DATE con una tabella di destinazione di Oracle con una colonna Oracle del tipo di dati DATE.

*Tabella 5. Associazione di una colonna DATE del DB2 con una colonna DATE di Oracle*

Colonna del DB2	Associazione dei dati di nickname	Colonna di Oracle
A_DATE DATE	A_DATE TIMESTAMP A_DATE DATE	A_DATE DATE

La tabella di destinazione di Oracle viene creata con un tipo di dati DATE di Oracle (che può contenere i dati relativi alla data e al timestamp). Il nickname iniziale per un tipo di dati DATE di Oracle in un database federato associa il tipo di dati del DB2 come un TIMESTAMP. Il Centro di replica DB2 e i comandi di System i per la replica modificano il tipo di dati del nickname su DATE, in modo che in Oracle venga replicato DATE, non un TIMESTAMP.

### Chiave di destinazione

Quando una tabella di destinazione compressa è interessata nella replica di modifica e acquisizione, il programma Apply necessita che essa abbia una chiave principale o un indice univoco, denominato *chiave di destinazione*.

È possibile scegliere quale colonna utilizzate come indice univoco per la propria tabella di destinazione. I seguenti tipi di tabelle di destinazione vengono compressi e richiedono una chiave di destinazione:

- Copia dell'utente
- Punto nel tempo
- Replica
- CCD compresso

Se si crea una nuova tabella di destinazione, è possibile utilizzare il nome indice e lo schema predefiniti o modificare i valori predefiniti in modo da corrispondere alle proprie convenzioni di denominazione.

Il nome predefinito deriva dal profilo dell'oggetto di destinazione per il server di destinazione, se ne esiste uno. Se non è stato impostato questo profilo, il valore predefinito è IX più il nome della tabella di destinazione. Ad esempio, se il nome della tabella di destinazione è TGEMPLOYEE, il nome della propria tabella di destinazione corrisponde a IXTGEMPLOYEE.

### Opzioni per gli indici univoci

Le opzioni per la creazione di indici univoci dipendono dal fatto che si sta creando una nuova tabella di destinazione o si sta utilizzando una tabella di destinazione esistente.

#### Nuova tabella di destinazione

Per creare un indice univoco per una nuova tabella di destinazione, sono disponibili due opzioni:

- Specifica le colonne che si desiderano come indice univoco per la tabella contatti.
- Far scegliere l'indice univoco alla replica SQL.

Se non vengono selezionate colonne per l'indice univoco, la replica SQL controlla la tabella di origine per una delle seguenti definizioni, nel seguente ordine:

1. Una chiave principale
2. Un vincolo univoco
3. Un indice univoco

Se la replica SQL trova una di queste definizioni per la tabella di origine e le colonne associate vengono registrate e sono parte della tabella di destinazione, la replica SQL utilizza la chiave principale della tabella di origine (o l'indice univoco o RRN) come chiave di destinazione. Nel caso di un vincolo univoco, la replica SQL crea un indice univoco per la tabella di destinazione utilizzando le colonne del vincolo.

**System i** Per una tabella di origine System i che non presenta una chiave primaria o un indice univoco, modificare la registrazione per la tabella che utilizza l'RRN (relative record number) come fattore di univocità. Quando viene definito il membro di impostazione richieste, specificare la colonna RRN come indice univoco per la tabella destinazione.

**System i** Per le tabelle di destinazione su System i che utilizzano l'RRN come chiave di destinazione, occorre avviare il programma Apply su System i per eseguire la replica sulle tabelle di destinazione.

### Tabella di destinazione esistente

Per le tabelle di destinazione esistenti, è necessario selezionare l'indice univoco. È possibile selezionare una delle opzioni riportate di seguito:

- Utilizzare un indice già esistente per la tabella di destinazione.

Per utilizzare un indice esistente, selezionare le colonne che rappresentano l'indice nel Centro di replica. Se il Centro di replica trova una corrispondenza esatta imposta solamente una chiave di destinazione per il programma Apply da utilizzare, altrimenti crea un indice univoco e imposta una chiave di destinazione per il programma Apply da utilizzare.

- Creare un altro indice per la tabella di destinazione.

Se l'indice univoco ancora non esiste, sarà creato e sarà impostata la chiave di destinazione per il programma Apply da utilizzare.

**Importante:** Se viene selezionata una chiave per la tabella di destinazione che include colonne che possono essere aggiornate nella tabella di origine, è necessario indicare al programma Apply di apportare gli aggiornamenti speciali alle nuove colonne della chiave di destinazione.

### In che modo il programma Apply aggiorna le colonne della chiave di destinazione con l'opzione modifica della chiave di destinazione

Se si sceglie l'opzione modifica chiave di destinazione quando si definisce un membro d'impostazione sottoscrizione, il programma Apply apporta modifiche speciali alle colonne della chiave di destinazione quando la chiave di destinazione cambia.

## Prerequisito

È necessario che le colonne origine che sono parte della chiave di destinazione vengano registrate con le colonne pre-immagine nella tabella CD (o CCD), in modo che il programma Apply aggiorni le colonne delle chiavi di destinazione. Se non si definisce la registrazione di origine che acquisisce i valori pre-immagine delle colonne che compongono la chiave di destinazione, è necessario modificare la propria registrazione in modo da includerli prima di richiederli ad una tabella di destinazione con una differente chiave.

## Restrizioni

- Non è possibile utilizzare l'opzione modifica chiave di destinazione per le tabelle origine che vengono registrate per acquisire aggiornamenti come coppie elimina/inserisci.
- Non è possibile associare un'espressione in una tabella di origine ad una colonna chiave in una tabella di destinazione se il programma Apply aggiorna la tabella di destinazione basata sulle pre-immagini della colonna chiave di destinazione (ovvero, se la colonna TARGET\_KEY\_CHG della tabella IBMSNAP\_SUBS\_MEMBR presenta un valore Y per quella tabella di destinazione).

Dopo aver garantito che i valori pre-immagine delle colonne della chiave di destinazione sono nella tabella CD (o CCD), selezionare l'opzione membro d'impostazione sottoscrizione per il programma Apply per utilizzare i valori pre-immagine quando si aggiornano le colonne chiave di destinazione.

Se non viene specificato al programma Apply di utilizzare i valori pre-immagine durante l'aggiornamento delle colonne della chiave di destinazione, la replica SQL non replica correttamente i dati quando vengono aggiornate le colonne della tabella origine che sono parte della chiave di destinazione.

Il programma Apply tenta di aggiornare la riga nella tabella di destinazione con il nuovo valore, ma non trova il nuovo valore di chiave nella tabella di destinazione per aggiornarlo. Il programma Apply quindi converte l'aggiornamento in un INSERT e inserisce il nuovo valore di chiave nella tabella di destinazione. In questo caso, la vecchia riga con il vecchio valore di chiave resta nella tabella di destinazione (e non è necessaria).

Quando viene specificato che si desidera vengano effettuate modifiche alle colonne della chiave di destinazione utilizzando valori pre-immagine, il programma Apply è in grado di trovare la riga con il vecchio valore di chiave e aggiornare la riga utilizzando i nuovi valori. Ad esempio, se la variabile *target\_key\_chg* è impostata su N, l'istruzione SQL per l'operazione di aggiornamento è:

```
UPDATE targettable SET <non-key columns>= after-image values
WHERE <key columns> = after-image values
```

Se la variabile *target\_key\_chg* è impostata su Y, l'istruzione SQL per l'operazione di aggiornamento è:

```
UPDATE targettable SET <all columns> = after-image values
WHERE <key columns> = before-image values
```



---

## Capitolo 6. Replica di tipi di dati speciali nella replica SQL

Quando si replicano tipi di dati speciali, ad esempio tipi di dati LOB, ROWID o non DB2, è necessario tenere presente alcune condizioni e restrizioni. In alcuni casi, è necessario eseguire delle operazioni di configurazione aggiuntive affinché la replica SQL funzioni con questi tipi di dati.

Nei seguenti argomenti vengono fornite informazioni sulla replica di tipi di dati speciali:

---

### Restrizioni generali relative ai dati per la replica

La replica SQL presenta delle restrizioni specifiche per alcuni tipi di dati incluse le restrizioni per la codifica e il tipo di dati.

#### Restrizioni per la codifica dei dati

La replica SQL può replicare alcuni tipi di dati codificati.

##### EDITPROC

La replica SQL supporta le tabelle di origine del DB2 per z/OS definite con una routine di modifica (EDITPROC) per fornire ulteriore sicurezza dei dati. Per utilizzare queste tabelle come origini per la replica, è necessario che il sottosistema DB2 contenente le tabelle sia una Versione 8 o successiva con APAR PK13542 o successivo.

##### Funzione scalare di codifica in DB2 per Linux, UNIX e Windows

I dati delle colonne possono essere codificati e decodificati utilizzando la funzione scalare di codifica in DB2 per Linux, UNIX e Windows. Per utilizzare questa funzione con la replica, il tipo di dati deve essere VARCHAR PER DATI BIT all'origine. Tali dati vengono replicati con esito positivo se l'origine e la destinazione utilizzano la stessa tabella codici e sono disponibili le funzioni di decodifica. La replica delle colonne con dati codificati deve essere utilizzata solo su server che supportano la funzione DECRYPT\_BIN o DECRYPT\_CHAR.

##### **z/OS** FIELDPROC

La replica SQL supporta le colonne definite nelle tabelle DB2 per z/OS con procedure di campo (FIELDPROC) per la trasformazione dei valori. Il sottosistema DB2 contenente le tabelle aventi colonne FIELDPROC deve essere APAR PK75340 o successivo.

Se possibile, per migliorare le prestazioni creare il seguente indice nella propria tabella SYSIBM.SYSFIELDS:

```
CREATE INDEX "SYSIBM"."FIELDSEX"  
ON "SYSIBM"."SYSFIELDS"  
(TBCREATOR ASC,  
TBNAME ASC,  
NAME ASC)  
USING STOGROUP SYSDEFLT PRIQTY 100 SECQTY 100  
CLOSE NO;  
COMMIT;
```

#### Restrizioni per i tipi di dati

La replica SQL non può replicare i seguenti tipi di dati:

- Colonne LOB da origini relazionali non DB2
- Qualsiasi colonna in cui sia definito un VALIDPROC.

La replica SQL può replicare i seguenti tipi di dati in alcune circostanze:

- Dati LONG VARGRAPHIC (long variable graphic), se le tabelle di origine e di destinazione risiedono in DB2 per z/OS.
- I dati LONG VARCHAR e LONG VARGRAPHIC (long variable character) richiedono che le tabelle di origine risiedano in DB2 per z/OS o che le tabelle di origine e di destinazione siano in DB2 per Linux, UNIX e Windows. Quando si specifica DATA CAPTURE CHANGES per una tabella di origine, al momento della creazione della tabella, tutte le colonne LONG VARCHAR e LONG VARGRAPHIC sono automaticamente abilitate per la replica. Se si aggiungono colonne LONG VARCHAR alla tabella mediante l'istruzione ALTER TABLE e la tabella non aveva in precedenza alcuna colonna LONG, è necessario utilizzare l'istruzione ALTER TABLE per abilitare DATA CAPTURE CHANGES INCLUDE LONGVAR COLUMNS per le nuove colonne LONG VARCHAR o LONG VARGRAPHIC.

La replica SQL non può replicare una tabella contenente tipi di dati astratti.

La replica SQL può replicare le tabelle con colonne del tipo di dati spaziali, ma non può replicare le colonne reali del tipo di dati spaziali.

I tipi di dati definiti dall'utente (tipi di dati distinti in DB2) vengono convertiti nel tipo di dati di base nella tabella CD (change-data) prima della replica. Inoltre, se la replica SQL crea la tabella di destinazione come parte della definizione del membro della serie di richieste, i tipi definiti dall'utente vengono convertiti nel tipo di dati di base nella tabella di destinazione e nella tabella CD.

---

## Tipi di dati LOB (large object)

La replica SQL supporta i tipi di dati LOB, inclusi i dati BLOB (binary LOB), CLOB (character LOB) e DBCLOB (double-byte character LOB).

In questa sezione i tipi di dati BLOB, CLOB e DBCLOB vengono indicati come dati LOB

Il programma Capture legge il descrittore LOB nei record della registrazione per determinare se i dati nella colonna LOB sono stati modificati e devono essere replicati, ma non copia i dati LOB nelle tabelle CD (change-data). Quando viene modificata una colonna LOB, il programma Capture imposta un indicatore nella tabella CD. Quando il programma Apply legge questo indicatore, copia l'intera colonna LOB (non solo le parti modificate delle colonne LOB) direttamente dalla tabella di origine nella tabella di destinazione.

Poiché una colonna LOB può contenere un numero massimo di due GB di dati, è necessario verificare di disporre di sufficiente larghezza di banda della rete per il programma Apply. Allo stesso modo, le tabelle di destinazione devono disporre di spazio su disco sufficiente per ospitare i dati LOB.

### Restrizioni:

- Il programma Apply copia sempre la versione più corrente di una colonna LOB direttamente dalla tabella di origine (non la tabella CD), anche se tale colonna è più corrente delle altre colonne nella tabella CD. Pertanto, se la colonna LOB nella riga di destinazione cambia, è possibile che tale colonna LOB sia

incongruente con il resto dei dati in quella riga di destinazione. Per ridurre questa possibilità di incoerenza dei dati nella riga di destinazione, verificare che l'intervallo tra i cicli di Apply sia abbastanza breve per l'applicazione.

- È possibile replicare fino a 10 colonne LOB per tabella. Se si registra una tabella con più di 10 colonne LOB, il programma Apply restituisce un messaggio di errore. Il Centro di replica restituisce un messaggio di errore se si tenta di registrare più di 10 colonne LOB per tabella.
- È possibile copiare i dati LOB nelle tabelle della replica se viene disabilitato il rilevamento dei conflitti.
- Per copiare i dati LOB tra DB2 per OS/390 Versione 6 (o successiva) e DB2 per Linux, UNIX e Windows, è necessario che sia installato DB2 Connect Versione 7 o successiva.
- Non è possibile fare riferimento ai dati LOB utilizzando i nickname.
- I valori pre-immagine per le colonne LOB o ROWID non sono supportati.
- La replica non è supportata per DB2 Extenders per testo, audio, video, immagine o altri extender in cui i file di controllo aggiuntivi associati ai dati della colonna LOB dell'extender vengono gestiti al di fuori del database.
- La replica SQL può replicare solo un LOB completo. Non può replicare parti di un LOB.
- Non è possibile replicare le colonne LOB se si utilizza una configurazione del giornale remota nell'ambiente della replica su System i.
- Quando si utilizzano i LOB nella replica di aggiornamento ovunque, è necessario impostare il livello di conflitto su 0.

---

## Replica di nuovi tipi di dati DB2 Versione 9.7 (Linux, UNIX, Windows)

### Linux UNIX Windows

La replica SQL supporta nuovi tipi di dati introdotti con DB2 per Linux, UNIX e Windows Versione 9.7 al fine di agevolare la migrazione delle applicazioni a DB2.

Alcuni dei nuovi tipi di dati richiedono considerazioni particolari in relazione all'ambiente di replica. Le seguenti sezioni forniscono ulteriori dettagli:

- "TIMESTAMP con precisione ampliata"
- "DATE con opzione di compatibilità" a pagina 98
- "NUMBER" a pagina 98

### TIMESTAMP con precisione ampliata

La replica SQL supporta la replica di dati TIMESTAMP con precisione ampliata, da TIMESTAMP(0) a TIMESTAMP(12). È possibile associare colonne con livelli di precisione non corrispondenti. Se sia il database di origine che quello di destinazione e Capture e Apply sono Versione 9.7 o successiva, i dati di origine vengono completati o troncati a livello del database di destinazione.

In un ambiente di livello misto in cui solo il DB2 di origine è Versione 9.7, anche le colonne TIMESTAMP potrebbero dover essere completate o troncate. La replica di tali colonne può verificarsi solo quando sia Capture che Apply sono Versione 9.7 o successiva. Ad esempio, se è stata replicata un'origine V9.7 in una destinazione V9.5 e in una tabella registrata era inclusa una colonna TIMESTAMP(12), l'Apply V9.7 tronca sei cifre dalla parte relativa ai secondi frazionari del valore TIMESTAMP. Il troncamento è necessario poiché DB2 Versione 9.5 non supporta una precisione ampliata e quindi per i database V9.5 i valori TIMESTAMP hanno

una parte relativa ai secondi frazionari equivalente alla precisione predefinita del `TIMESTAMP(6)` di V9.7. Tabella 6 mostra il valore nell'origine e il conseguente valore troncato nella destinazione.

**z/OS** **Nota:** quando gestisce questi nuovi tipi di dati, la replica SQL tratta l'origine o la destinazione DB2 per z/OS come se si trattasse di DB2 per Linux, UNIX e Windows Versione 9.5 o precedente.

Tabella 6. Troncamento di `TIMESTAMP(12)` durante la replica

Valore di origine nel <code>TIMESTAMP(12)</code>	Valore di destinazione nel <code>TIMESTAMP(6)</code>
2009-07-10-10.33.42.458499823012	2009-07-10-10.33.42.458499

Se il database di destinazione è precedente a V9.7, i valori `TIMESTAMP` con precisione inferiore al `TIMESTAMP(6)` predefinito vengono completati automaticamente da DB2 affinché la parte relativa ai secondi frazionari contenga sei cifre.

## DATE con opzione di compatibilità

L'opzione di compatibilità della data memorizza il tipo di `DATE` con una parte aggiuntiva relativa all'ora (`HH:MM:SS`). Questo formato è conforme alla rappresentazione della data di altri sistemi di gestione dei database relazionali quali Oracle, in cui il tipo di dato `DATE` include `YYYY-MM-DD HH:MM:SS`.

La replica SQL gestisce i database senza compatibilità della data come database DB2 precedenti a V9.7 e come sottosistemi DB2 per z/OS. Quando la compatibilità della data è abilitata, DB2 gestisce le colonne definite come `DATE` così come gestisce le colonne definite come `TIMESTAMP(0)`.

Abilitare `DATE` come supporto `TIMESTAMP(0)` impostando il numero di posizione bit 7 (0x40) della variabile di registro `DB2_COMPATIBILITY_VECTOR` prima di creare un database. Con la replica SQL è possibile creare le seguenti associazioni colonne tra `DATE` e `TIMESTAMP(0)`:

### Da `DATE` a `TIMESTAMP(0)`

Se il database di origine non ha alcuna compatibilità data abilitata, il valore di destinazione viene completato in `YYYY-MM-DD-00:00:00`.

### Da `TIMESTAMP(0)` a `DATE`

Se il database di destinazione non ha alcuna compatibilità data abilitata, il valore `TIMESTAMP(0)` viene troncato in `YYYY-MM-DD`.

## NUMBER

Il tipo di dati `NUMBER` supporta applicazioni che utilizzano il tipo di dati `NUMBER` di Oracle. DB2 gestisce i dati `NUMBER` internamente come `DECFLOAT` se non è stata specificata alcuna precisione o scala e come `DECIMAL` con precisione o scala se questi attributi sono specificati.

Poiché la replica SQL supporta già `DECFLOAT` e `DECIMAL`, le colonne definite con uno qualsiasi di questi tre tipi numerici possono associarsi tra loro: `NUMBER` con `DECFLOAT` o `DECIMAL`, `DECFLOAT` con `NUMBER` o `DECIMAL` e `DECIMAL` con `NUMBER` o `DECFLOAT`.

---

## Replica delle tabelle con colonne identità

La replica SQL consente l'utilizzo di colonne identità sia nelle tabelle di origine che in quelle di destinazione, ma a causa delle restrizioni di DB2 potrebbe essere necessario eseguire una procedura aggiuntiva nel caso in cui la tabella di origine abbia colonne definite con la clausola `AS IDENTITY GENERATED ALWAYS`.

Le colonne identità vengono gestite dalla replica in maniera differente a seconda che si trovino nella tabella di origine o in quella di destinazione:

### Tabella origine

Se la colonna identità si trova nella tabella di origine e si desidera replicarla in una tabella di destinazione, registrarla e richiederla nella tabella di origine secondo la procedura abituale. Le tabelle CD e di destinazione vengono create con colonne numeriche che contengano i valori. Ad esempio, una colonna di origine definita come `GENERATE ALWAYS` potrebbe essere replicata in una colonna `BIGINT` della tabella di destinazione. Le colonne della tabella CD e di destinazione non possono essere colonne identità, pertanto non è possibile replicare una colonna identità di una tabella di origine in una colonna identità di una tabella di destinazione.

### Tabella di destinazione

Se la colonna identità si trova nella tabella di destinazione, non includerla nella configurazione della replica durante la definizione del membro della serie di richieste. La colonna viene popolata automaticamente nel momento in cui la replica effettua degli inserimenti nella tabella di destinazione o la aggiorna. Il comportamento della colonna identità è lo stesso degli inserimenti e degli aggiornamenti effettuati da qualsiasi altra applicazione. Se si replica la stessa tabella di origine in più tabelle di destinazione aventi colonne identità, i valori di identità di tali tabelle sono tra loro indipendenti.

DB2 non consente inserimenti in colonne definite con la clausola `AS IDENTITY GENERATED ALWAYS` pertanto questa clausola non è supportata per le tabelle di destinazione della replica SQL. Tuttavia, esistono diverse opzioni per la replica di tali colonne:

- Creare una tabella di destinazione senza la clausola `IDENTITY`.
- Creare la tabella di destinazione con una colonna definita come `AS IDENTITY GENERATED BY DEFAULT`.

Per le colonne definite con la clausola `AS IDENTITY GENERATED BY DEFAULT`, occorre distinguere tra l'intervallo di valori di origine e quello di destinazione, poiché DB2 non garantisce l'univocità delle colonne identità tra due differenti database DB2.

Ad esempio, la colonna identità di un sito potrebbe essere impostata con numeri pari (`START WITH 2, INCREMENT BY 2`), mentre nell'altro sito la colonna identità potrebbe essere impostata con numeri dispari (`START WITH 1, INCREMENT BY 2`). Inoltre, è possibile assegnare degli intervalli ai siti (ad esempio, da 1 a 10.000 in un sito e da 20.000 a 40.000 nell'altro). L'approccio pari-dispari garantisce che, in caso di conflitto, le due righe differenti per cui è stata generata accidentalmente la stessa chiave di identità non si sovrascrivano quando l'azione di conflitto deve forzare la modifica.

Il tipo di dato della colonna identità (SMALLINT, INTEGER o BIGINT) deve essere determinato da esigenze applicative, ad esempio il numero maggiore atteso nella colonna.

Se i numeri non possono essere riutilizzati, le colonne identità devono essere NO CYCLE. Pianificare la procedura da eseguire al raggiungimento del valore massimo (SQLSTATE 23522). Se si utilizza CYCLE, assicurarsi che il riutilizzo di un numero non comporti alcun problema per qualsiasi utilizzo esistente del numero, incluso ciò che accade durante la replica.

---

## Capitolo 7. Creazione di serie secondarie di dati in un ambiente di replica SQL

Generalmente, la replica implica la creazione di serie secondarie. Può implicare la scelta di alcune colonne e righe da replicare da una tabella di origine quando si registra un'origine della replica. Può implicare la scelta di alcune colonne registrate da replicare in ciascuna tabella di destinazione quando si creano le serie di richieste.

A seconda dei requisiti di replica, è possibile creare delle serie secondarie dei dati nell'origine durante la registrazione o nella destinazione durante la sottoscrizione:

- Se si dispone di una sola destinazione per un'origine, o se più destinazioni richiedono gli stessi dati, è possibile creare delle serie secondarie o manipolare i dati durante la registrazione, perché non è necessario considerare diverse esigenze per diverse destinazioni.
- Se si dispone di un'origine e di destinazioni multiple e le destinazioni multiple presentano diversi requisiti per i dati da applicare, non è possibile creare una serie secondaria durante la registrazione. In questo caso, verrà creata una serie secondaria di dati durante la sottoscrizione.

Le viste vengono utilizzate per l'organizzazione dei dati in serie secondarie all'atto della registrazione, mentre i predicati delle query vengono utilizzati per l'organizzazione dei dati in serie secondarie all'atto della sottoscrizione. In molte situazioni, l'utilizzo dei predicati delle sottoscrizioni o delle viste registrate dipende dalle proprie preferenze. Alcuni fattori potrebbero esercitare una certa influenza:

- Le viste potrebbero esistere già e soddisfare le qualifiche per essere una vista registrata per la replica.
- Le viste potrebbero costituire un approccio più semplice alla verifica della serie secondaria definita per la replica.
- I predicati delle sottoscrizioni sono memorizzati nelle tabelle di controllo della replica, eliminando quindi la necessità di creare e gestire le viste.

Non utilizzare queste tecniche se si sta eseguendo la replica nelle tabelle di destinazione della replica. La tabella principale e le tabelle della replica nelle configurazioni di aggiornamento ovunque replicano i dati reciprocamente. Le tabelle della replica possono presentare una serie secondaria delle colonne della tabella di origine, se le colonne che non vengono utilizzate sono impostabili su null. Altrimenti, le tabelle della replica devono contenere le stesse colonne della tabella di origine, quindi non è possibile creare serie secondarie di colonne, aggiungere nuove colonne o rinominare le colonne.

---

### Impostazione secondaria di dati durante la registrazione

Alcune tecniche avanzate sono utili durante l'impostazione secondaria dei dati prima o dopo che vengano catturati da un'origine registrata. Tali tecniche sono particolarmente utili se si desidera catturare la stessa serie secondaria di dati una volta e replicarla su molte tabelle di destinazione.

È possibile scegliere l'impostazione secondaria dei dati prima o dopo che vengano catturati da un'origine registrata. Le tecniche in questa sezione possono essere utilizzate in tutte le configurazioni di replica ad eccezione di repliche di aggiornamenti o peer-to-peer.

L'impostazione secondaria dei dati durante la registrazione può migliorare le prestazioni della replica in quanto riduce la quantità di dati che il programma Capture aggiunge alla tabella CD e la quantità che il programma Apply legge. Riduce inoltre la memoria in quanto nella tabella CD ci sono meno righe.

## **Impostazione secondaria dei dati di origine utilizzando le viste**

Quando si registra un'origine, si scelgono le colonne che si desidera rendere disponibili per la replica. Le colonne selezionate vengono catturate per la replica. In alcuni casi, dopo che si registra un'origine per modificare la replica, si potrebbe voler registrare una vista dell'origine.

Ad esempio, si supponga che il reparto di Risorse umane gestisca una tabella che contiene i dati del personale, comprese le informazioni salariali. Per gestire un database di backup, l'intera tabella del personale viene registrata e richiesta sul sito di backup. Tuttavia, se un altro sito di destinazione desidera richiedere la tabella del personale, si potrebbe voler nascondere le informazioni salariali a questo secondo richiedente. La soluzione consiste nella registrazione di una vista sulla tabella del personale e consentire privilegi di accesso solo alla vista registrata per il secondo richiedente, in modo che le informazioni salariali risultano con accesso protetto. Una sottoscrizione può essere creata su questa vista registrata.

È possibile inoltre registrare viste che includono due o più tabelle di origine. Ad esempio, se si ha una tabella clienti e una tabella filiale, l'unica modalità di impostazione secondaria adeguata dei clienti sulla destinazione potrebbe risultare l'unione delle due tabelle in modo che solo i clienti di una certa filiale vengono replicati su una determinata destinazione. In tale caso, è necessario fare attenzione di evitare doppie eliminazioni.

## **Definizione dei trigger sulle tabelle CD per impedire l'acquisizione di righe specifiche**

In alcuni scenari di replica, si potrebbe desiderare di impedire alcune modifiche alle righe da acquisire e replicare nella tabella di destinazione. Per escludere alcune modifiche da acquisire, definire i trigger nella propria tabella CD.

Quando viene registrata un'origine, gli strumenti di gestione consentono di selezionare quali colonne si desidera acquisire, ma non consentono di impedire che alcune modifiche in quelle righe vengano replicate. In alcuni scenari di replica, si potrebbe desiderare di impedire alcune modifiche alle righe da acquisire e replicare nella tabella di destinazione. Ad esempio, se si desidera che la propria tabella di destinazione contenga tutte le righe e si desidera che non venga mai eliminata nessuna riga fra di esse, non si desidera replicare le eliminazioni dall'origine.

Per escludere l'acquisizione di alcune modifiche, definire i trigger nella propria tabella CD. Questi trigger specificano quali modifiche il programma Capture dovrebbe escludere impedendo l'aggiunta di righe corrispondenti alle modifiche apportate nella tabella CD. Non è possibile creare questi trigger utilizzando il Centro di replica, ma è possibile crearli manualmente per una tabella CD esistente (ovvero, dopo che l'origine è registrata). Il programma Capture ignora gli errori che mostrano un SQLSTATE di 99999 e la riga non è inserita nella tabella CD.



Ad esempio, si ipotizzi che si desidera che vengano ignorate tutte le operazioni DELETE della tabella di origine durante la replica dalla tabella SAMPLE.TABLE, dove la tabella CD è SAMPLE.CD\_TABLE. I seguenti trigger escludono le righe che sono operazioni DELETE dall'essere inserite nella tabella CD:

```
CREATE TRIGGER SAMPLE.CD_TABLE_TRIGGER
NO CASCADE BEFORE INSERT ON SAMPLE.CD_TABLE
REFERENCING NEW AS CD
FOR EACH ROW MODE DB2SQL
WHEN (CD.IBMSNAP_OPERATION = 'D')
SIGNAL SQLSTATE '99999' ('CD INSERT FILTER')
```

Si potrebbe desiderare di aggiungere l'istruzione trigger creata all' SQL generato durante la registrazione. È necessario avviare l' SQL modificato per completare la registrazione e per creare i trigger sulle tabelle CD.

Questi trigger vengono eseguiti ogni volta che il programma Capture tenta di inserire una riga nella tabella CD, quindi bisogna considerare se l'utilizzo dei trigger in questa posizione offre la miglior prestazione nella propria configurazione della replica. È possibile aumentare o diminuire la velocità di trasmissione dati aggiungendo trigger alla tabella CD. Utilizzare i trigger sulla tabella CD per escludere un gran numero di modifiche all'origine. Se si pianifica l'acquisizione della maggior parte delle modifiche, ma si vogliono escludere alcune di esse dalla replica, è possibile che si desideri escludere le righe non desiderate durante la sottoscrizione.

---

## Creazione di serie secondarie di dati durante la sottoscrizione

La creazione di serie secondarie di dati durante la sottoscrizione può migliorare le prestazioni della replica, riducendo la quantità di dati raccolti dal programma Apply. Un numero minore di righe nelle tabelle di destinazione riduce anche i requisiti di memoria.

Il programma Apply utilizza i predicati per determinare i dati da copiare durante l'aggiornamento completo e la replica di modifica/cattura. Il Centro di replica e ASNCLP consentono di specificare i valori dei predicati per l'aggiornamento completo e la replica di modifica/cattura. L'utente può aggiungere ulteriori informazioni sui predicati per utilizzare solo la replica di modifica/cattura, perché tali informazioni non sono disponibili durante l'aggiornamento completo. È necessario aggiungere queste ulteriori informazioni sui predicati nella tabella IBMSNAP\_SUBS\_MEMBR nella colonna UOW\_CD\_PREDICATES mediante l' SQL fornito dall'utente.

Ad esempio, si immagini di disporre di una tabella registrata denominata ALL.CUSTOMERS e che la tabella CD associata sia denominata ALL.CD\_CUSTOMERS. Si consideri di volere che la tabella della sottoscrizione contenga solo una serie secondaria di ALL.CUSTOMERS dove la colonna ACCT\_BALANCE è maggiore di 50000, e che vengano conservati i dati cronologici nella tabella di destinazione (non si desidera eliminare i dati dalla tabella di destinazione). È possibile creare il membro della serie di richieste con un valore PREDICATES di 'ACCT\_BALANCE > 50000'.

Non è possibile utilizzare il Centro di replica o ASNCLP per impedire le eliminazioni nella tabella di destinazione, poiché le informazioni relative al tipo di operazione vengono memorizzate nella tabella CD e non sono disponibili nella vista o nella tabella di origine. Pertanto, è necessario creare il predicato di modifica/cattura utilizzando un'istruzione SQL che includa le seguenti

informazioni. A seconda dello scenario utilizzato, può essere necessario aggiungere delle colonne nell'istruzione update per essere certi di aggiornare una singola riga nella tabella IBMSNAP\_SUBS\_MEMBR:

```
UPDATE ASN.IBMSNAP_SUBS_MEMBR SET UOW_CD_PREDICATES = 'IBMSNAP_OPERATION <>'D''
WHERE APPLY_QUAL = 'apply_qual' AND SET_NAME = 'set_name' AND
SOURCE_OWNER = 'ALL' AND SOURCE_TABLE = 'CUSTOMERS'
```

È necessario configurare manualmente la colonna UOW\_CD\_PREDICATES per i predicati del membro della serie di richieste che fanno riferimento alle colonne non disponibili durante l'aggiornamento completo, incluse le colonne pre-immagine nella tabella CD, le colonne della tabella CD o le colonne della tabella UOW.

Per impostazione predefinita, il programma Apply non unisce la tabella UOW e la tabella CD per le tabelle di destinazione copia dell'utente; raccoglie e applica i dati direttamente dalla tabella CD. Se il predicato deve fare riferimento alla tabella UOW e la tabella di destinazione è una copia utente, è necessario impostare il valore della colonna JOIN\_UOW\_CD su Y nella tabella IBMSNAP\_SUBS\_MEMBR. Impostando questo indicatore, si garantisce che il programma Apply unirà le tabelle UOW e CD.

Se si desidera specificare dei predicati che superano 1024 byte (la capacità della colonna PREDICATES della tabella IBMSNAP\_SUBS\_MEMBR) per una serie secondaria di righe, è necessario utilizzare una vista di origine.

Se si utilizzano istruzioni con predicati complessi per una serie di richieste, racchiudere l'intera espressione tra parentesi. Ad esempio, quando si utilizzano le clausole AND e OR in un'istruzione del predicato, racchiudere l'espressione nel seguente modo:

```
((TOSOURCE = 101 AND STATUS IN (202,108,109,180,21,29,32,42))
OR (SOURCE = 101))
```

---

## Capitolo 8. Manipolazione dei dati in un ambiente di replica SQL

È possibile trasformare o migliorare i dati di origine prima che vengano replicati nelle tabelle di destinazione.

Ad esempio, è possibile manipolare i dati in uno dei seguenti modi:

- Eseguire la ripulitura dei dati
- Eseguire l'aggregazione dei dati
- Inserire i dati nelle colonne nella tabella di destinazione che non esistono nell'origine

Utilizzare il programma Apply per manipolare i dati, prima o dopo l'applicazione dei dati nella destinazione da parte di Apply, in uno dei seguenti modi:

- Utilizzo di procedure memorizzate o istruzioni SQL
- "Associazione delle colonne di origine e di destinazione che presentano nomi diversi" a pagina 107
- "Creazione di colonne calcolate" a pagina 107

È possibile manipolare i dati prima o dopo la relativa cattura. Manipolare i dati durante la registrazione e non durante la sottoscrizione, se si desidera eseguire la manipolazione una volta e replicare i dati trasformati in molte tabelle di destinazione. Manipolare i dati durante la sottoscrizione e non durante la registrazione se si desidera catturare tutti i dati di origine ed applicare selettivamente i dati trasformati sulle singole destinazioni.

In alcuni scenari di replica, è possibile che l'utente desideri manipolare il contenuto dei dati di origine memorizzati nella tabella CD. Un trigger, un'espressione mediante la sottoscrizione o una vista di origine possono essere utilizzati per eseguire lo stesso lavoro. Ciascun metodo presenta dei vantaggi e degli svantaggi. Un trigger può essere troppo costoso per i cicli di CPU richiesti. Una vista consente di configurare la funzione una volta piuttosto che in diverse richieste.

Ad esempio, se nella tabella di origine manca un determinato valore, l'utente può chiedere al programma Capture di catturare i valori null.

È anche possibile utilizzare i trigger nella tabella CD per specificare le condizioni necessarie al programma Capture per migliorare i dati durante l'inserimento dei dati nella tabella CD. In questo caso, è possibile specificare che il programma Capture inserisca un valore predefinito nella tabella CD quando rileva un valore null nell'origine. È possibile utilizzare il seguente codice per creare un trigger che fornisce un valore predefinito ambiguo se i dati mancano nell'aggiornamento della tabella di origine:

```
CREATE TRIGGER ENHANCECD
NO CASCADE BEFORE INSERT ON CD_TABLE
REFERENCING NEW AS CD
FOR EACH ROW MODE DB2SQL
WHEN (CD.COL1 IS NULL)
SET CD.COL1 = 'MISSING DATA'
END
```

Invece del trigger, è possibile utilizzare la funzione scalare COALESCE del DB2 in una vista di origine registrata o in un'espressione della sottoscrizione. In una vista registrata, la funzione coalesce restituisce il primo valore non null.

#### Esempio parziale che utilizza una vista di origine

```
CREATE VIEW SAMPLE.SRCVIEW (columns) AS SELECT
... COALESCE(A.COL1, 'MISSING DATA') ...
FROM SAMPLE.TABLE A
```

#### Esempio parziale utilizzando un'espressione

```
COALESCE(CD.COL1, 'MISSING DATA')
```

---

## Ottimizzazione dei dati mediante le procedure memorizzate o le istruzioni SQL

Quando si definiscono le informazioni relative alla serie di richieste, è anche possibile definire le istruzioni di elaborazione runtime utilizzando le istruzioni SQL o le procedure memorizzate che si desidera vengano eseguite dal programma Apply ogniqualvolta elabora una serie specifica. Questi processi runtime consentono la manipolazione dei dati durante la replica.

Tali istruzioni sono utili per eliminare le tabelle CCD e per controllare la sequenza in cui vengono elaborate le serie di richieste. È possibile eseguire le istruzioni di elaborazione runtime sul server di controllo Capture prima che venga elaborata una serie di richieste oppure sul server di destinazione prima o dopo l'elaborazione di una serie di richieste. Ad esempio, è possibile eseguire le istruzioni SQL prima di richiamare i dati, dopo averli replicati nelle tabelle di destinazione o in entrambi i casi.

**Restrizioni per i nickname:** Le tabelle del DB2 federate (che utilizzano nickname) generalmente vengono aggiornate in una singola UOW (Unit Of Work). Quando si aggiunge un'istruzione SQL in una serie di richieste che viene eseguita dopo l'applicazione di tutti i dati alle destinazioni da parte del programma Apply, è necessario premettere all'istruzione SQL un'istruzione SQL COMMIT nelle seguenti situazioni:

- L'istruzione SQL inserisce, aggiorna o elimina un nickname da un server diverso da quello su cui sono presenti le tabelle di destinazione o i nickname di destinazione per la serie di richieste.
- L'istruzione SQL inserisce, aggiorna o elimina una tabelle presente sul server di controllo Apply, ma i nickname di destinazione per la serie di richieste sono contenuti su un server remoto.

L'istruzione COMMIT esegue il commit del lavoro del programma Apply prima che esso elabori l'istruzione SQL aggiunta.

Le procedure memorizzate utilizzano l'istruzione SQL CALL senza parametri. Il nome della procedura deve avere una lunghezza massima di 18 caratteri (per System i il massimo è 128). Se la tabella di origine o di destinazione è contenuta in un database relazionale non DB2, le istruzioni SQL vengono eseguite nel database DB2 federato. Le istruzioni SQL non vengono mai eseguite in un database non DB2. Le procedure runtime di ciascun tipo vengono eseguite insieme, come una singola transazione. È anche possibile definire dei valori SQLSTATE accettabili per ciascuna istruzione.

Utilizzare la routine di chiusura ASNDONE, se si desidera manipolare i dati al termine dell'elaborazione di ciascuna serie (e non al termine dell'elaborazione di una specifica serie).

---

## Associazione delle colonne di origine e di destinazione che presentano nomi diversi

Quando si utilizza il Centro di replica o il programma della riga comandi ASNCLP per definire un membro della serie di richieste e la tabella di destinazione indicata non esiste, è possibile rinominare le colonne nella destinazione, indipendentemente dal tipo di tabella di destinazione. È anche possibile modificare gli attributi delle colonne compatibili.

Inoltre, è possibile modificare gli attributi delle colonne (tipo di dati, lunghezza, scala, precisione e la possibilità di impostazione su null) quando essi sono compatibili. Non è possibile utilizzare gli strumenti di gestione della replica per rinominare le colonne delle tabelle di destinazione esistenti.

Gli strumenti di gestione tentano di associare le colonne per nome, se esiste la tabella di destinazione indicata dal membro della serie di richieste. Se le colonne di origine e di destinazione non corrispondono, è possibile utilizzare gli strumenti per associare le colonne dall'origine alla destinazione oppure creare una vista della tabella di destinazione contenente una corrispondenza con i nomi delle colonne di origine.

---

## Creazione di colonne calcolate

Sebbene non sia possibile modificare i nomi delle colonne nelle tabelle di destinazione esistenti, è possibile modificare le espressioni delle colonne di origine, in modo che vengano associate correttamente o siano compatibili con le colonne nelle tabelle di destinazione esistenti.

Utilizzando le espressioni SQL, è anche possibile derivare nuove colonne dalle colonne di origine esistenti. Per i tipi di tabelle di destinazione di aggregazione, è possibile definire nuove colonne utilizzando le funzioni di aggregazione, ad esempio COUNT o SUM. Per gli altri tipi di tabelle di destinazione è possibile definire nuove colonne utilizzando le funzioni scalari nelle espressioni. Se le colonne nelle tabelle di origine e di destinazione si differenziano solo per il nome, ma sono compatibili, è possibile utilizzare il Centro di replica o ASNCLP per associare una colonna con l'altra.

Ad esempio, si immagini di disporre della tabella di origine esistente (SRC.TABLE) e della tabella di destinazione (TGT.TABLE):

```
CREATE TABLE SRC.TABLE (SRC_COL1 CHAR(12) NOT NULL, SRC_COL2 INTEGER,
    SRC_COL3 DATE, SRC_COL4 TIME, SRC_COL5 VARCHAR(25))
CREATE TABLE TGT.TABLE (TGT_COL1 CHAR(12) NOT NULL,
    TGT_COL2 INTEGER NOT NULL, TGT_COL3 TIMESTAMP, TGT_COL4 CHAR(5))
```

Eeguire le seguenti operazioni per associare la tabella di destinazione desiderata mediante le colonne calcolate durante la richiesta:

1. Utilizzare il Centro di replica per associare SRC\_COL1 dalla tabella di origine con TGT\_COL1 nella tabella di destinazione. Poiché queste colonne sono compatibili, non è necessario utilizzare un'espressione per associare l'una all'altra.

2. Utilizzare l'espressione `COALESCE(SRC_COL2, 0)` per calcolare i valori della colonna e associarli a `TGT_COL2`. Poiché `SRC_COL2` è impostabile su null e `TGT_COL2` è `NOT NULL`, è necessario eseguire questa operazione per verificare che per `TGT_COL2` venga fornito un valore `NOT NULL`.
3. Utilizzare l'espressione `TIMESTAMP(CHAR(SRC_COL3) CONCAT CHAR(SRC_COL4))` per calcolare i valori delle colonne ed associarli a `TGT_COL3`. L'espressione di questa colonna fornisce i dati da associare alla colonna timestamp nel database di destinazione.
4. Utilizzare l'espressione `SUBSTR(SRC_COL5, 1,5)` per calcolare i valori delle colonne e associarli a `TGT_COL4`.

---

## Capitolo 9. Funzionamento del programma Capture per la replica SQL

Questa sezione è relativa all'acquisizione basata su registrazione per database DB2. Se si sta utilizzando la cattura basata su trigger, i trigger vengono creati alla registrazione e non è possibile eseguire le operazioni descritte in questa sezione.

---

### Avvio del programma Capture (Linux, UNIX, Windows e z/OS)

z/OS

Linux UNIX Windows

Avviare il programma Capture per iniziare la cattura dei dati dalla registrazione per i database DB2. Se si sta utilizzando la cattura basata su trigger per un'origine relazionale non DB2, i trigger vengono creati alla registrazione e non è necessario avviare il programma Capture.

#### Prima di iniziare

- Configurare le connessioni sul server origine e il server di controllo Capture.
- Accertarsi di disporre dell'autorizzazione adeguata.
- Creare le tabelle di controllo per lo schema Capture appropriato.
- Definire le registrazioni.
- Configurare i programmi Capture e Apply.

#### Informazioni su questa attività

**Nota:** Il programma Capture non acquisisce le modifiche fatte dalle utilità DB2, poiché queste non registrano le modifiche in un modo visibile al programma Capture.

Quando si avvia il programma Capture, è possibile anche specificare i parametri di avvio.

Una volta avviato, il programma Capture potrebbe non iniziare la cattura dei dati. Inizierà tale cattura solo dopo che il programma Apply gli segnala che ha aggiornato completamente una tabella destinazione. Quindi il programma Capture inizia la cattura delle modifiche dalla registrazione per una data tabella di origine.

#### Procedure

Per avviare il programma Capture su Linux, UNIX, Windows e z/OS, utilizzare uno dei seguenti metodi:

Metodo	Descrizione
Centro di replica	Utilizzare la finestra Avvia Capture. Per visualizzare la finestra, fare clic sulla cartella <b>Server di controllo Capture</b> nel ramo <b>Operazioni</b> della struttura ad albero dell'oggetto e nel pannello dei contenuti fare clic con il tasto destro del mouse sul server di controllo Capture su cui è ubicato il programma Capture che si desidera avviare. Selezionare <b>Avvia Capture</b> .

Metodo	Descrizione
<p><b>z/OS</b> <b>Linux UNIX Windows</b></p> <p>Comando di sistema asncap</p>	<p>Utilizzare questo comando per avviare il programma Capture e facoltativamente specificare i parametri di avvio.</p>
<p><b>z/OS</b></p> <p>Console z/OS o TSO</p>	<p>Su z/OS, è possibile avviare il programma Capture utilizzando JCL oppure come un'attività avviata dal sistema. È possibile specificare nuovi valori del parametro di richiamo quando si avvia un programma Capture con JCL.</p> <p>In z/OS, la lunghezza totale dei parametri che può essere specificata nel campo PARM= ha un limite di 100 byte. Per superare questo limite, adesso programmi di replica consentono di specificare tutti parametri aggiuntivi necessari nel set di dati SYSIN.</p> <p>Quando l'istruzione SYSIN DD è inclusa nel JCL di chiamata, il programma Capture concatena automaticamente ciò che viene specificato nel set di dati SYSIN con i parametri PARM=. I parametri Capture possono essere specificati solo nel set di dati SYSIN. Qualsiasi parametro LE deve essere specificato nel campo PARM= o in LE _CEE_ENVFILE=DD, seguito da una barra (/).</p> <p><b>Esempio:</b></p> <pre>L'asterisco /* indica una riga di commento // CAP EXEC PGM=ASNCAP,PARM='LE/Parametri Capture' /* I parametri possono essere rappresentati da uno qualsiasi /* o nessuno dei parametri LE e Capture //SYSIN DD * /* parametri Capture aggiuntivi, uno o più /* parametri su ciascuna riga CAPTURE_SERVER=DSN!! CAPTURE_SCHEMA=CAPCAT DEBUG=Y LOGSTDOU=N</pre>
<p><b>Windows</b></p> <p>Servizi Windows</p>	<p>È possibile creare un servizio di replica DB2 sui sistemi operativi Windows per avviare automaticamente il programma Capture all'avvio del sistema.</p>

Per verificare se un programma Capture è avviato, utilizzare uno dei metodi riportati di seguito:

- z/OS** Se si sta eseguendo in modalità batch, ricercare nella console z/OS o nella registrazione lavoro z/OS dei messaggi che indicano che il programma è stato avviato.
- Ricercare nel file di log della diagnostica di Capture (*capture\_server.capture\_schema.CAP.log* su z/OS e *db2instance.capture\_server.capture\_schema.CAP.log* su Linux, UNIX e Windows) un messaggio che indica che il programma sta catturando le modifiche. Ad esempio:

```
ASN0104I Change capture has been started for the source
table "REGRESS.TABLE1" for changes found in the log beginning
with log sequence number "0000:0275:6048".
```
- Ricercare nella tabella IBMSNAP\_CAPTRACE un messaggio che indica che il programma sta catturando le modifiche.
- Utilizzare la finestra Messaggi Capture nel Centro di replica per visualizzare un messaggio che indica che il programma è stato avviato. Per visualizzare la



finestra, fare clic con il tasto destro del mouse sul server Capture che contiene il programma Capture di cui si desidera visualizzare i messaggi e selezionare **Prospetti** → **Messaggi Capture**.

- Utilizzare la finestra Controlla stato nel Centro di replica o il comando asncmd status per visualizzare lo stato di tutti i thread di Capture. Per visualizzare la finestra, fare clic con il tasto destro del mouse sul server Capture che contiene il programma Capture che si desidera verificare e selezionare **Controlla stato**.

---

## Avvio del programma Capture da un punto conosciuto nella registrazione DB2

È possibile richiedere al programma Capture di rileggere la registrazione di recupero DB2 da un punto conosciuto e rielaborare i record di registrazione già catturati e applicati.

### Informazioni su questa attività

**Importante:** Questa procedura deve essere utilizzata solo quando la tabella di destinazione è una copia utente.

### Procedure

1. Arrestare i programmi Capture e Apply.
2. Impostare i valori Capture RETENTION\_LIMIT e LAG\_LIMIT al massimo, come mostrato nella seguente istruzione SQL:  

```
UPDATE ASN.IBMSNAP_CAPPARMS SET RETENTION_LIMIT=99999,LAG_LIMIT=99999;
```
3. Se i valori SYNCHPOINT delle tabelle IBMSNAP\_UOW, CD, IBMSNAP\_REGISTER e IBMSNAP\_PRUNCNTL sono superiori al valore LSN da cui si intende avviare Capture, utilizzare SQL per impostare il valore al punto da cui si intende avviare le transazioni di ricattura. Nel seguente esempio, 00000006F5638E60000 è il numero di sequenza di registrazione e 2009-09-05-09.55.43.316970 è la data/ora a cui si riporta indietro il programma Capture affinché avvii la lettura della registrazione.

```
UPDATE ASN.IBMSNAP_REGISTER SET SYNCHPOINT = x'00000006F5638E600000',  
SYNCHTIME=TIMESTAMP('2009-05-05-09.55.43.316970');
```

```
UPDATE ASN.IBMSNAP_REGISTER SET CD_OLD_SYNCHPOINT=x'00000006F5638E600000',  
CD_NEW_SYNCHPOINT=x'00000006F5638E600000',  
CCD_OLD_SYNCHPOINT=x'00000006F5638E600000'  
WHERE GLOBAL_RECORD='N';
```

```
UPDATE ASN.IBMSNAP_SUBS_SET SET LASTRUN=TIMESTAMP('2009-09-05-09.55.43.316970'),  
LASTSUCCESS=TIMESTAMP('2009-05-05-09.55.43.316970'),  
SYNCHPOINT=x'00000006F5638E600000', SYNCHTIME=TIMESTAMP('2009-05-05-09.55.43.316970')  
WHERE WHOS_ON_FIRST='S' AND SET_NAME='BACK1';
```

```
UPDATE ASN.IBMSNAP_PRUNCNTL SET SYNCHPOINT =x'00000006F5638E600000',  
SYNCHTIME=TIMESTAMP('2009-05-05-09.55.43.316970');
```

```
UPDATE ASN.IBMSNAP_PRUNE_SET SET SYNCHPOINT =x'00000006F5638E600000',  
SYNCHTIME=TIMESTAMP('2009-05-05-09.55.43.316970');
```

```
DELETE FROM ASN.IBMSNAP_UOW;
```

```
INSERT INTO ASN.IBMSNAP_RESTART (MAX_COMMITSEQ, MIN_INFLIGHTSEQ,  
MAX_COMMIT_TIME,CURR_COMMIT_TIME,CAPTURE_FIRST_SEQ)  
VALUES (x'00000006F5638E600000',x'00000006F5638E600000',  
'2009-05-05-09.55.43.316970','2009-05-05-09.55.43.316970',  
x'00000006F5638E600000');
```

4. Avviare il programma Capture in modalità WARMNS e avviare il programma Apply con i normali parametri di avvio.

---

## Starting the Capture program (System i)

### System i

Start the Capture program to begin capturing data from the journal.

#### Prima di iniziare

Before you start the Capture program, ensure that the following prerequisites are met:

- You have the proper authorization.
- The control tables are created for the appropriate Capture schema, and registrations are defined.
- The replication programs are configured if the Capture program is reading a remote journal.

#### Informazioni su questa attività

After you start the Capture program, the Capture program might not start capturing data right away. It will start capturing data only after the Apply program signals the Capture program to start capturing changes from the log for a given source table.

#### Procedure

To start the Capture program on System i, use one of the following methods:

Method	Description
<b>STRDPRCAP system command (System i)</b>	Use the Start DPR Capture (STRDPRCAP) command to start capturing changes.
<b>Replication Center</b>	Use the Start Capture window. To open the window, click the <b>Capture Control Servers</b> folder in the <b>Operations</b> branch of the object tree, and in the contents pane right-click the Capture control server on which the Capture program that you want to start is located. Select <b>Start Capture</b> .

---

## Parametri di funzionamento predefiniti per il programma Capture

Quando si creano tabelle di controllo Capture, i valori predefiniti per i parametri di funzionamento del programma Capture vengono salvati nella tabella IBMSNAP\_CAPPARMS.

I valori predefiniti sono illustrati in Tabella 7 a pagina 113 e Tabella 8 a pagina 113.

Tabella 7. Impostazioni predefinite per i parametri operativi Capture (Linux, UNIX, Windows, z/OS)

Parametro operativo	Valore predefinito	Nome colonna nella tabella IBMSNAP_CAPPARMS
capture_server	DB2DBDFT <sup>1</sup>	non applicabile
capture_schema	ASN <sup>2</sup>	non applicabile
add_partition	n <sup>4</sup>	non applicabile
asynchlogrd	n <sup>4</sup>	non applicabile
retention_limit	10080 minuti	RETENTION_LIMIT
lag_limit	10080 minuti	LAG_LIMIT
commit_interval	30 secondi	COMMIT_INTERVAL
prune_interval	300 secondi	PRUNE_INTERVAL
trace_limit	10080 minuti	TRACE_LIMIT
monitor_limit	10080 minuti	MONITOR_LIMIT
monitor_interval	300 secondi	MONITOR_INTERVAL
memory_limit	32 MB	MEMORY_LIMIT
autoprun	y <sup>3</sup>	AUTOPRUNE
term	y <sup>3</sup>	TERM
autostop	n <sup>4</sup>	AUTOSTOP
caf	n <sup>4</sup>	non applicabile
logreuse	n <sup>4</sup>	LOGREUSE
logstdout	n <sup>4</sup>	LOGSTDOUT
sleep_interval	5 secondi	SLEEP
capture_path	Directory in cui è stato avviato Capture <sup>5</sup>	CAPTURE_PATH
startmode	warmsi <sup>6</sup>	STARTMODE

**Nota:**

1. Il server di controllo Capture è il valore della variabile di ambiente DB2DBDFT per Windows, Linux e UNIX, se è specificata tale variabile. Non esiste alcun valore predefinito per z/OS.
2. Non è possibile modificare il valore predefinito per lo schema Capture. Per utilizzare un altro schema Capture, utilizzare il parametro di avvio **capture\_schema**.
3. Sì
4. No
5. Se Capture viene avviato come un servizio Windows, il relativo percorso Capture è \sqllib\bin.
6. Il programma Capture si avvia a caldo. Passa all'avvio a freddo solo se questa è la prima volta in cui il programma viene avviato.

Tabella 8. Impostazioni predefinite per i parametri facoltativi Capture (System i)

Parametro operativo	Valore predefinito	Nome colonna nella tabella IBMSNAP_CAPPARMS
CAPCTLLIB	ASN <sup>1</sup>	non applicabile

Tabella 8. Impostazioni predefinite per i parametri facoltativi Capture (System i) (Continua)

Parametro operativo	Valore predefinito	Nome colonna nella tabella IBMSNAP_CAPPARMS
JOB	*LIBL/QZSNDPR	non applicabile
JRN	*ALL	non applicabile
RETAIN	10080 minuti	RETENTION_LIMIT
LAG	10080 minuti	LAG_LIMIT
FRCFRQ	30 secondi	COMMIT_INTERVAL
CLNUPITV	*IMMED <sup>2</sup>	non applicabile
CLNUPITV	86400 secondi <sup>2</sup>	PRUNE_INTERVAL
CLNUPITV	*IMMED <sup>2</sup>	non applicabile
TRCLMT	10080 minuti	TRACE_LIMIT
MONLMT	10080 minuti	MONITOR_LIMIT
MONITV	300 secondi	MONITOR_INTERVAL
MEMLMT	32 MB	MEMORY_LIMIT
WAIT	120 secondi	non applicabile
RESTART	*YES <sup>3</sup>	non applicabile

**Nota:**

1. Non è possibile modificare il valore predefinito per lo schema Capture. Per utilizzare un altro schema Capture, specificare il parametro **CAPCTLLIB** quando si avvia il programma Capture. I valori predefiniti per la maggior parte degli altri parametri operativi sono memorizzati nella tabella IBMSNAP\_CAPPARMS.
2. **CLNUPITV** ha due parametri secondari. Per impostazione predefinita, il programma Capture viene eliminato subito dopo che avvia l'esecuzione e ogni volta che viene raggiunto l'intervallo di eliminazione (che, per impostazione predefinita, è ogni 24 ore).
3. Per impostazione predefinita, il programma Capture viene avviato a caldo.

## Descrizioni dei parametri di funzionamento di Capture

Quando si avvia il programma Capture, è possibile selezionare parametri di avvio. Di seguito sono riportati i parametri di avvio e dei consigli sulla scelta dei valori per ciascun parametro.

Tutti i parametri si applicano a z/OS, Linux, UNIX e Windows, a meno che non sia specificato diversamente.

- “add\_partition (Linux, UNIX, Windows)” a pagina 115
- “asynchlogrd” a pagina 115
- “autoprune” a pagina 115
- “autostop” a pagina 116
- caf
- “capture\_path” a pagina 116
- “capture\_schema” a pagina 117
- “capture\_server” a pagina 118
- “commit\_interval” a pagina 118
- “lag\_limit” a pagina 118
- “logreuse” a pagina 119
- “logstdout” a pagina 119

- “memory\_limit” a pagina 119
- “monitor\_interval” a pagina 120
- “prune\_interval” a pagina 120
- “retention\_limit” a pagina 121
- “sleep\_interval” a pagina 122
- “startmode” a pagina 122
- “term” a pagina 123
- “trace\_limit” a pagina 123

Linux UNIX Windows

## add\_partition (Linux, UNIX, Windows)

Valore predefinito: **add\_partition=n**

Il parametro **add\_partition** specifica se il programma Capture viene avviato leggendo il file di log per le partizioni aggiunte di nuovo da quando il programma Capture è stato riavviato l’ultima volta.

Impostare **add\_partition=y** per far leggere al programma Capture i file di log. Su ciascuna nuova partizione, quando il programma Capture viene avviato in modalità di avvio a caldo, Capture legge il file di log a partire dal primo LSN (log sequence number) utilizzato da DB2 dopo che viene immessa la prima istruzione database CONNECT per l’istanza DB2.

## asynchlogrd

Valore predefinito: **asynchlogrd=n**

Il parametro **asynchlogrd** specifica che si desidera che il programma Capture utilizzi un thread dedicato per l’acquisizione delle transazioni dal log di recupero di DB2. Il thread del programma di lettura delle transazioni effettua una prelettura delle transazioni di cui si è eseguito il commit in un buffer di memoria, da un altro thread riceve le transazioni e le elabora in istruzioni SQL per inserirle nella tabella CD. Questa modalità asincrona può migliorare la produttività di Capture in tutti gli ambienti, con particolari benefici per i database partizionati e la condivisione di dati z/OS.

In sistemi con livelli di attività molto elevati, questa prelettura potrebbe portare un maggiore utilizzo di memoria. Regolare il parametro **memory\_limit** di conseguenza. Se si dispone di un basso volume di modifiche, si può optare per il valore predefinito di N, per ridurre il consumo della CPU.

## autoprune

Valore predefinito: **autoprune=y**

Il parametro **autoprune** specifica se il programma Capture elimina automaticamente alcune delle relative tabelle di controllo. Per impostazione predefinita, con **autoprune=y**, il programma Capture elimina automaticamente le righe nelle tabelle CD e UOW così come le tabelle IBMSNAP\_CAPTRACE, IBMSNAP\_CAPMON e IBMSNAP\_SIGNAL. Se si imposta **autoprune=n**, occorre utilizzare il comando prune per eliminare tali tabelle.

Se si avvia Capture con il comando `autoprune` attivo, impostare l'intervallo di eliminazione per ottimizzarne la frequenza per l'ambiente di replica. Il programma Capture utilizza i seguenti parametri per determinare le righe che risultano sufficientemente obsolete per essere eliminate:

- **retention\_limit** per le tabelle di segnali, CD e UOW
- **monitor\_limit** per le tabelle di controllo
- **trace\_limit** per la tabella di traccia Capture

## autostop

**Valore predefinito:** `autostop=n`

Il parametro **autostop** controlla se il programma Capture rimane attivo o viene terminato quando raggiunge la fine della registrazione.

Per impostazione predefinita (`autostop=n`) il programma Capture non viene terminato dopo il richiamo delle transazioni.

Utilizzare l'opzione **autostop=y** se si sta eseguendo la replica in un ambiente mobile o collegato occasionalmente. L'opzione **autostop** garantisce che il programma Capture richiama tutte le transazioni idonee e si arresta quando raggiunge la fine della registrazione. È necessario avviare nuovamente il programma Capture per richiamare ulteriori transazioni. Si potrebbe voler utilizzare l'opzione **autostop=y** anche in un ambiente di prova.

**Suggerimento:** nella maggior parte dei casi, non si deve utilizzare **autostop=y** in quanto ciò aggiunge un sovraccarico alla gestione della replica (ad esempio, è necessario riavviare il programma Capture).

z/OS

## caf

**Valore predefinito:** `n`

L'opzione predefinita è **caf =n**. È possibile sovrascrivere questo valore predefinito e indicare al programma Capture di utilizzare CAF (Call Attach Facility) specificando l'opzione **caf =y**. L'opzione **caf =y** specifica che il programma di replica sovrascriverà l'RRS (Recoverable Resource Manager Services) di connessione predefinito e verrà eseguito con CAF di connessione.

Se l'RRS non è disponibile, l'utente riceverà un messaggio e il programma di replica passerà a CAF. Il messaggio avvisa che il programma non è stato in grado di inizializzare una connessione poiché l'RRS non è stato avviato. Il programma eseguirà pertanto un tentativo di utilizzo di CAF. Il programma verrà eseguito correttamente con CAF di connessione.

## capture\_path

Il percorso di Capture è la directory in cui il programma Capture memorizza i propri file di lavoro e il file di log. Per impostazione predefinita, il percorso di Capture è la directory in cui viene avviato il programma.

z/OS

Poiché il programma Capture è un'applicazione POSIX, il percorso predefinito di Capture dipende dalla modalità di avvio del programma:

### Richiesta della riga comandi USS

La directory in cui si è avviato il programma.

### Attività o JCL avviati

La directory principale nel file system USS dell'ID utente associato all'attività o al lavoro avviati.

È possibile specificare un nome di percorso o un HLQ (High Level Qualifier), come ad esempio //CAPV9. Quando si utilizza un HLQ, vengono creati dei file sequenziali conformi alle convenzioni di denominazione dei file per i nomi dei file del set di dati sequenziale z/OS. I set di dati sequenziali sono relativi all'ID utente che esegue il programma. Altrimenti, tali nomi di file sono simili a quelli memorizzati in un percorso di directory denominato esplicitamente, con l'HLQ concatenato come prima parte del nome di file. Ad esempio, *sysadm.CAPV8.nomefile*. Utilizzare un HLQ potrebbe essere conveniente se si desidera che il log Capture ed i file LOADMSG vengano gestiti dal sistema (SMS).

#### Linux UNIX Windows

È possibile modificare il percorso di Capture per specificare dove si desidera che tale programma memorizzi i relativi file. È possibile specificare un nome di percorso, ad esempio: */home/db2inst/capture\_files*. Se si avvia il programma Capture come servizio di Windows, per impostazione predefinita il programma Capture viene avviato nella directory *\sqlib\bin*.

## capture\_schema

**Valore predefinito:** `capture_schema=ASN`

Il parametro **capture\_schema** identifica il programma Capture da avviare. Per impostazione predefinita, lo schema Capture è ASN.

Se è già stato impostato un altro schema, è possibile avviare il programma Capture specificando tale schema con il parametro **capture\_schema**.

Si potrebbero utilizzare più schemi Capture nelle seguenti situazioni:

### Raggiungimento indipendenza dell'applicazione

Creare più schemi Capture in modo da avere un programma Capture per l'applicazione A e un altro programma Capture per l'applicazione B. Ciascun programma Capture utilizza le relative tabelle di controllo. Se uno dei programmi Capture non è attivo, viene interessata solo un'applicazione. L'altra applicazione non è interessata in quanto viene servita da un altro programma Capture.

### Incontro di diversi requisiti di applicazione

Creare più schemi Capture se si hanno diverse applicazioni che utilizzano le stesse tabelle di origine, ma hanno differenti requisiti di dati. Ad esempio, un'applicazione libro paga necessita di dati sensibili degli impiegati mentre ciò non è necessario ad un registro di impiegati interni. È possibile registrare le informazioni confidenziali in uno schema Capture, ma non nell'altro schema Capture. In modo simile, è possibile registrare una tabella più di una volta se alcune applicazioni necessitano un funzionamento diverso del programma Capture. Ad esempio, forse alcune applicazioni richiedono che il programma Capture salvi gli aggiornamenti come coppie di eliminazione e inserimento.

## Individuazione dei problemi con registrazioni

Se si ha un problema con una registrazione, è possibile creare un altro schema Capture e spostarvi le registrazioni in funzione. In tale modo, è possibile eseguire il debug della registrazione causa del problema nello schema di origine ed eseguire le registrazioni non coinvolte utilizzando l'altro schema.

## capture\_server

**z/OS**

**Valore predefinito:** `capture_server=Nessuno`

**Linux UNIX Windows**

**Valore predefinito:** `capture_server=`valore della variabile di ambiente `DB2DBDFT`, se impostata

Il parametro `capture_server` specifica il server di controllo Capture.

**z/OS**

È necessario specificare il parametro `capture_server`. Le tabelle di controllo Capture si trovano nel nome del sottosistema DB2. Poiché il programma Capture legge la registrazione DB2, tale programma deve essere eseguito sullo stesso server del database di origine.

**Linux UNIX Windows**

Le tabelle di controllo Capture (come la tabella di registro) contengono le informazioni di registrazione per le tabelle origine e si trovano nel server di controllo Capture.

## commit\_interval

**Valore predefinito:** `commit_interval=30`

Il parametro `commit_interval` specifica la frequenza, in secondi, con cui il programma Capture esegue il commit dei dati sulle tabelle di controllo Capture, comprese le tabelle UOW e CD. Per impostazione predefinita, il programma Capture attende 30 secondi prima di eseguire il commit dei dati sulle tabelle CD e UOW. Vengono eseguiti dei blocchi sulle tabelle aggiornate nell'ambito dell'intervallo di commit. Valori più elevati del parametro `commit_interval` riducono l'uso della CPU per il programma Capture, ma potrebbero anche far aumentare la latenza per le serie di richieste spesso in esecuzione poiché il programma Apply può estrarre solo dati di cui è stato eseguito il commit.

## lag\_limit

**Valore predefinito:** `lag_limit=10 080`

Il parametro `lag_limit` rappresenta il numero di minuti di ritardo che il programma Capture può accumulare nell'elaborazione dei record dalla registrazione DB2.

Per impostazione predefinita, se i record di registrazione sono più obsoleti di 10 080 minuti (sette giorni), il programma Capture non verrà avviato a meno che non si specifichi un valore per il parametro `startmode` che consente al programma Capture di passare ad un avvio a freddo.

Se il programma Capture non viene avviato in quanto stato raggiunto il limite di ritardo, è necessario determinare perché il programma Capture è in ritardo nella lettura della registrazione. Se si è in un ambiente di prova, dove non si ha un uso



pratico del parametro del limite di ritardo, si potrebbe voler impostare tale limite su un valore più elevato e tentare di avviare nuovamente il programma Capture. In alternativa, se la tabella di origine nell'ambiente di prova contiene pochi dati, si potrebbe voler utilizzare un avvio a freddo e aggiornare completamente i dati in tutte le tabelle di destinazione.

## logreuse

**Valore predefinito:** `logreuse=n`

Il programma Capture memorizza informazioni operative in un file di log.

**z/OS** Il nome del file di log non contiene il nome di un'istanza DB2. Ad esempio, SRCDB1.ASN.CAP.log. Tale file è memorizzato nella directory specificata dal parametro `capture_path`. Se il parametro `capture_path` è specificato come un HLQ (High Level Qualifier), vengono applicate le convenzioni di denominazione dei file del set di dati sequenziale di z/OS; il nome `capture_schema` utilizzato per il file di log viene quindi troncato a otto caratteri.

**Linux UNIX Windows** Il nome del file di log è `edb2instance.capture_server.capture_schema.CAP.log`. Ad esempio, DB2INST.SRCDB1.ASN.CAP.log.

Per impostazione predefinita (`logreuse=n`), il programma Capture aggiunge i messaggi al file di log, anche dopo il riavvio del programma Capture. Mantenere l'impostazione predefinita se si desidera la cronologia dei messaggi. Nelle situazioni seguenti si potrebbe desiderare che il programma Capture elimini la registrazione e la crei nuovamente al riavvio (`logreuse=y`):

- La registrazione sta assumendo dimensioni elevate e si desidera eseguirne la pulizia.
- Non è necessaria la cronologia memorizzata nella registrazione.
- Si desidera risparmiare spazio.

## logstdout

**Valore predefinito:** `logstdout=n`

Il parametro `logstdout` è disponibile solo se si utilizza il comando `asncap`, non è disponibile nel Centro di replica.

Per impostazione predefinita, il programma Capture invia alcuni messaggi informativi e di avvertenza solo al file di log. Si potrebbe scegliere di inviare i messaggi all'output standard (`logstdout=y`) se si sta eseguendo la risoluzione dei problemi o si sta controllando il funzionamento del programma Capture in un ambiente di prova.

## memory\_limit

**Valore predefinito:** `memory_limit=32`

Il parametro `memory_limit` specifica la quantità di memoria, in megabyte, che il programma Capture può utilizzare.

Per impostazione predefinita, il programma Capture utilizza 32 megabyte di memoria per memorizzare le informazioni di transazione prima di trasferirle in un

file ubicato nella directory **capture\_path**. È possibile modificare il limite di memoria in base alle necessità delle proprie prestazioni. L'impostazione di un limite di memoria più elevato può migliorare le prestazioni di Capture, ma diminuire la memoria disponibile per altri usi sul sistema in uso. L'impostazione di un limite di memoria inferiore libera memoria per altri usi. Se il limite di memoria impostato è troppo basso e il programma Capture esegue il trasferimento su un file, verrà utilizzato più spazio sul sistema in uso e l'I/O rallenterà tale sistema.

È possibile controllare il limite di memoria utilizzando Replication Alert Monitor. È possibile inoltre utilizzare i dati nella tabella CAPMON per determinare il numero di transazioni del sistema di origine trasferite su disco a causa di limitazioni di memoria. Sommare i valori contenuti nella colonna TRANS\_SPILLED della tabella CAPMON.

### **monitor\_interval**

**Valore predefinito:** `monitor_interval=300`

Il parametro **monitor\_interval** specifica la frequenza con cui il programma Capture scrive le informazioni sulla tabella IBMSNAP\_CAPMON.

Per impostazione predefinita, il programma Capture inserisce righe nella tabella di controllo Capture ogni 300 secondi (5 minuti). Questo parametro operativo opera insieme all'intervallo di commit. Se si è interessati ad eseguire il controllo dei dati in modo dettagliato, utilizzare un intervallo di controllo più vicino all'intervallo di commit.

### **monitor\_limit**

**Valore predefinito:** `monitor_limit=10080`

Il parametro **monitor\_limit** specifica l'intervallo di tempo in cui le righe devono essere contenute nella tabella di controllo prima di poter essere eliminate.

Per impostazione predefinita, vengono eliminate le righe presenti nella tabella IBMSNAP\_CAPMON da più di 10 080 minuti (sette giorni). La tabella IBMSNAP\_CAPMON contiene statistiche operative per il programma Capture. Utilizzare il limite di controllo predefinito se si necessita di meno di una settimana di statistiche. Se le statistiche vengono controllate di frequente, probabilmente non è necessario conservare quelle di una settimana ed è possibile impostare un limite di controllo inferiore, cosicché la tabella di controllo Capture viene eliminata più di frequente e le vecchie statistiche vengono eliminate. Se si desidera utilizzare le statistiche per analisi cronologiche e si necessita di statistiche di più di una settimana, aumentare il limite di controllo.

### **prune\_interval**

**Valore predefinito:** `prune_interval=300`

Il parametro **prune\_interval** specifica la frequenza con cui il programma Capture tenta di eliminare vecchie righe da alcune tabelle di controllo. Questo parametro è valido solo se **autoprun=y**.

Per impostazione predefinita, il programma Capture elimina i dati delle tabelle CD e UOW ogni 300 secondi (cinque minuti). Se i dati non vengono eliminati

abbastanza spesso da queste tabelle, è possibile che queste esauriscano lo spazio e quindi che il programma Capture venga arrestato. Se i dati delle tabelle vengono eliminati troppo spesso o durante i periodi di picco, l'eliminazione può interferire con i programmi applicativi in esecuzione sullo stesso sistema. È possibile impostare la frequenza di eliminazione ottimale per l'ambiente di replica in uso. Le prestazioni risultano in genere ottimali quando le tabelle sono di piccole dimensioni.

Prima di ridurre l'intervallo di eliminazione, accertarsi che i dati vengano applicati di frequente in modo che possa verificarsi l'eliminazione. Se il programma Apply non applica i dati di frequente, è inutile ridurre l'intervallo di eliminazione in quanto il programma Apply deve eseguire la replica dei dati su tutte le destinazioni prima che i dati delle tabelle CD e UOW possano essere eliminati.

L'intervallo di eliminazione determina la frequenza con cui il programma Capture tenta di eliminare le tabelle. Esso opera insieme ai seguenti parametri, che determinano quando i dati sono sufficientemente obsoleti per essere eliminati: **trace\_limit**, **monitor\_limit**, **retention\_limit**. Ad esempio, se il parametro **prune\_interval** è 300 secondi ed il parametro **trace\_limit** è 10080 secondi, il programma Capture proverà ad eliminare i dati ogni 300 secondi. Se questo rileva nella tabella di traccia delle righe più obsolete di 10080 minuti (7 giorni), le elimina.

## **retention\_limit**

**Valore predefinito:** **retention\_limit**=10 080

Il parametro **retention\_limit** determina la durata di permanenza dei vecchi dati nelle tabelle CD, UOW e IBMSNAP\_SIGNAL prima che divengano idonei per l'eliminazione del limite di conservazione.

Se il processo di eliminazione normale è inibito a causa di serie di richieste disattivate o eseguite raramente, i dati possono rimanere nelle tabelle CD e UOW per lunghi periodi di tempo. Se tali dati divengono più obsoleti della data/ora attuale DB2 meno il valore limite di conservazione, il processo di eliminazione del limite di conservazione elimina tale dati dalle tabelle. Se le serie di richieste vengono eseguite molto raramente o si arrestano i programmi Apply, le tabelle CD e UOW possono assumere grandi dimensioni e diventare idonee per l'eliminazione del limite di conservazione.

Le tabelle di destinazione in uso devono essere aggiornate per sincronizzarle con l'origine se alcune delle righe eliminate sono candidate per la replica, ma per qualche motivo non sono state ancora applicate alla tabella di destinazione. È possibile evitare un aggiornamento completo utilizzando limiti di conservazione superiori; tuttavia, le tabelle CD e UOW aumenteranno di dimensione e utilizzeranno spazio sul sistema in uso.

Se si sta eseguendo la replica di aggiornamenti, l'eliminazione del limite di conservazione garantisce che le transazioni rifiutate vengono eliminate. Il risultato delle transazioni rifiutate se si utilizza il rilevamento del conflitto con tabelle di destinazione replica e le transazioni in conflitto vengono eliminati. Le righe nelle tabelle CD e UOW di pertinenza di tali transazioni rifiutate non vengono replicate e vengono eliminate quando il limite di conservazione viene raggiunto. Un aggiornamento completo non è richiesto se tutte le vecchie righe eliminate erano di pertinenza delle transazioni rifiutate.

L'eliminazione della conservazione assicura inoltre che le informazioni di segnali non più richieste vengano eliminate dalla tabella IBMSNAP\_SIGNAL.

## **sleep\_interval**

**Valore predefinito:** `sleep_interval=5`

L'intervallo di attesa è il numero di secondi che il programma Capture attende prima di leggere nuovamente la registrazione dopo che raggiunge la fine di questa e il buffer è vuoto. Per la condivisione dei dati sul sistema operativo z/OS, l'intervallo di attesa rappresenta il numero di secondi che il programma Capture attende dopo che il buffer ritorna ad essere pieno meno della metà.

Per impostazione predefinita, il programma Capture attende 5 secondi. Cambiare l'intervallo di attesa se si desidera ridurre il carico di lavoro del programma Capture leggendo la registrazione. Un intervallo di attesa inferiore indica una minore possibilità di ritardo. Un intervallo di attesa maggiore fornisce salvataggi della CPU potenziali in un sistema che non viene aggiornato di frequente.

## **startmode**

**Valore predefinito:** `startmode=warmsi`

È possibile avviare Capture utilizzando una delle seguenti modalità di avvio:

### **warmsi (avvio a caldo, passa inizialmente all'avvio a freddo)**

Il programma Capture viene avviato a caldo; se invece è la prima volta che si avvia il programma Capture passa all'avvio a freddo. Utilizzare questa modalità di avvio se si desidera assicurare che l'avvio a freddo si verifichi solo all'avvio iniziale del programma Capture.

### **warmns (avvio a caldo, non passa mai all'avvio a freddo)**

Il programma Capture si avvia a caldo. Se non può essere avviato a caldo, esso non passa all'avvio a freddo. Quando si utilizza **warmns** nell'ambiente di replica quotidiano, si ha l'opportunità di risolvere qualsiasi problema (come database o tablespace non disponibili) che impediscono un avvio a caldo. Utilizzare questa modalità di avvio per impedire che si verifichi un imprevisto avvio a freddo. Quando il programma Capture viene avviato a caldo, riprende l'elaborazione dal punto in cui era terminata. In caso di errori dopo l'avvio del programma Capture, tale programma termina e lascia tutte le tabelle intatte.

**Suggerimento:** non è possibile utilizzare **warmns** per avviare il programma Capture per la prima volta perché non è presente alcuna informazione di avvio a caldo quando si avvia all'inizio il programma Capture. La prima volta che si avvia il programma Capture, utilizzare il parametro **startmode cold** quindi utilizzare il parametro **startmode warmns**. Se non si desidera passare da un parametro **startmode** all'altro, è possibile utilizzare invece **warmsi**.

### **a freddo**

Durante l'avvio a freddo, il programma Capture elimina tutte le righe nelle relative tabelle CD e UOW durante l'inizializzazione. Tutte le serie di richieste a queste origini di replica vengono completamente aggiornate durante il successivo ciclo di elaborazione Apply (ovvero, tutti i dati vengono copiati dalle tabelle di origine a quelle di destinazione). Se il programma Capture tenta di eseguire l'avvio a freddo, ma è stato

disabilitato l'aggiornamento completo, il programma Capture verrà avviato, ma il programma Apply non avrà esito positivo e verrà emesso un messaggio di errore.

Raramente si richiede esplicitamente che il programma Capture esegua un avvio a freddo. L'avvio a freddo è necessario solo la prima volta in cui viene avviato il programma Capture e **warmsi** è la modalità di avvio consigliata.

**Importante:** non eseguire l'avvio a freddo del programma Capture se si desidera conservare cronologie accurate dei dati modificati. Potrebbe verificarsi un intervallo se il programma Apply non può eseguire la replica delle modifiche prima che il programma Capture termini. Inoltre, poiché si desidera evitare avvii a freddo, non inserire l'avvio a freddo come valore predefinito per STARTMODE nella tabella IBMSNAP\_CAPPARMS.

## **term**

**Valore predefinito:** `term=y`

Il parametro **term** determina la modalità secondo cui lo stato di DB2 interessa il funzionamento del programma Capture.

Per impostazione predefinita, il programma Capture termina se DB2 termina.

Utilizzare **term=n** se si desidera che il programma Capture attenda l'avvio di DB2 se DB2 non è attivo. Se DB2 è in stato di sospensione, Capture non termina; rimane attivo ma non utilizza il database.

## **trace\_limit**

**Valore predefinito:** `trace_limit=10080`

**trace\_limit** specifica l'intervallo di tempo in cui le righe devono essere contenute nella tabella IBMSNAP\_CAPTRACE prima di essere eliminate.

Quando il programma Capture esegue l'eliminazione, per impostazione predefinita, le righe nella tabella IBMSNAP\_CAPTRACE possono essere eliminate ogni 10080 minuti (sette giorni). La tabella CAPTRACE contiene le informazioni traccia di controllo per il programma Capture. Ogni azione svolta dal programma Capture viene registrata in questa tabella; questa può quindi aumentare di dimensioni molto rapidamente se il programma Capture è molto attivo. Modificare il limite di traccia in base alle proprie necessità per le informazioni di controllo.

---

## **Metodi di modifica dei parametri di Capture**

È possibile modificare i valori salvati dei parametri di funzionamento di Capture ed è possibile inoltre sovrascrivere temporaneamente questi valori quando si avvia il programma o mentre il programma è in esecuzione.

### **Impostazione di nuovi valori predefiniti nella tabella IBMSNAP\_CAPPARMS**

La tabella IBMSNAP\_CAPPARMS contiene i parametri che è possibile modificare per controllare il funzionamento del programma Capture. Il nome dello schema della tabella è lo schema Capture. Una volta creata la tabella, questa contiene i valori predefiniti che vengono forniti per il programma Capture. Se non è impostato il valore della colonna nella tabella IBMSNAP\_CAPPARMS, vengono utilizzati i valori predefiniti.

### Specifica dei valori per i parametri all'avvio del programma Capture

È possibile specificare dei valori per il programma Capture quando lo si avvia. I valori impostati durante l'avvio controllano il funzionamento di Capture per la sessione corrente, essi sovrascrivono i valori dei parametri di funzionamento predefiniti e qualsiasi valore che potrebbe esistere nella tabella di parametri Capture. Essi non aggiornano i valori nella tabella di parametri Capture. Se non si modifica tale tabella prima di avviare il programma Capture e non si specifica alcun parametro quando si avvia tale programma, i valori predefiniti vengono utilizzati per i parametri di funzionamento.

### Modifica dei valori dei parametri mentre il programma Capture è in esecuzione

Mentre Capture è in esecuzione, è possibile modificare temporaneamente i relativi parametri di funzionamento. Il programma Capture utilizzerà i nuovi valori finché i valori non vengono modificati nuovamente o finché il programma Capture non viene arrestato e riavviato. Durante la sessione, è possibile modificare i parametri Capture ogni volta che si desidera.

Linux UNIX Windows

### Esempio 1

Presupporre che non si desidera utilizzare le impostazioni predefinite per l'intervallo di commit Capture per lo schema ASNPROD di Capture.

1. Aggiornare la tabella di parametri Capture per lo schema ASNPROD di Capture. Impostare l'intervallo di commit a 60 secondi; quindi, quando in seguito si avvia il programma Capture, l'impostazione predefinita per l'intervallo di commit sarà 60 secondi.

```
update asnprod.ibmsnap_capparms set commit_interval=60;
```

2. Infine, si potrebbero voler apportare delle migliorie alle prestazioni in modo da tentare di avviare il programma Capture utilizzando un intervallo di commit inferiore. Anziché modificare il valore nella tabella di parametri Capture, avviare semplicemente il programma Capture con l'impostazione del parametro di intervallo di commit a 20 secondi. Mentre il programma Capture è in esecuzione utilizzando un intervallo di commit di 20 secondi, controllarne le prestazioni.

```
asncap capture_server=srcdb1 capture_schema=asnprod commit_interval=20
```

3. Si desidera provare un intervallo di commit inferiore. Anziché arrestare il programma Capture, si inoltra una richiesta di modifica dei parametri che imposta l'intervallo di commit a 15 secondi. Il programma Capture continua l'esecuzione e solo ora esegue il commit dei dati ogni 15 secondi.

```
asnccmd capture_server=srcdb1 capture_schema=asnprod chgparms  
commit_interval=15
```

**Importante:** il parametro che viene modificato deve seguire immediatamente **chgparms**.

4. È possibile continuare a controllare le prestazioni e modificare il parametro di intervallo di commit senza arrestare il programma Capture. Eventualmente, una volta individuato l'intervallo di commit che soddisfa le proprie esigenze, è possibile aggiornare le tabelle di parametri di Capture (come descritto nel passo 1) in modo che al successivo avvio, il programma Capture utilizza il nuovo valore come intervallo di commit predefinito.

System i

## Esempio 2

Presupporre che non si desidera utilizzare le impostazioni predefinite per l'intervallo di commit Capture per lo schema ASNPROD di Capture.

1. Aggiornare la tabella di parametri Capture per lo schema ASNPROD di Capture. Impostare l'intervallo di commit a 90 secondi; quindi, quando in seguito si avvia il programma Capture, l'impostazione predefinita per l'intervallo di commit sarà 90 secondi.  
`CHGDPRCAP CAPCTLLIB(ASNPROD) FRCFRQ(90)`
2. Infine, si potrebbero voler apportare delle migliorie alle prestazioni in modo da tentare di avviare il programma Capture utilizzando un intervallo di commit inferiore. Anziché modificare il valore nella tabella di parametri Capture, avviare semplicemente il programma Capture con l'impostazione del parametro di intervallo di commit a 45 secondi. Mentre il programma Capture è in esecuzione utilizzando un intervallo di commit di 45 secondi, controllarne le prestazioni.  
`STRDPRCAP CAPCTLLIB(ASNPROD) FRCFRQ(45)`
3. Si desidera provare un intervallo di commit inferiore. Anziché arrestare il programma Capture, si inoltra una richiesta di modifica dei parametri che imposta l'intervallo di commit a 30 secondi. Il programma Capture continua l'esecuzione e solo ora esegue il commit dei dati ogni 30 secondi. (Nota: su System i, non è possibile impostare l'intervallo di commit ad un valore inferiore a 30 secondi).  
`OVRDPRCAP CAPCTLLIB(ASNPROD) FRCFRQ(30)`
4. Eventualmente, una volta individuato l'intervallo di commit che soddisfa le proprie esigenze, è possibile aggiornare le tabelle di parametri di Capture (come descritto nel passo 1) in modo che al successivo avvio, il programma Capture utilizza il nuovo valore come intervallo di commit predefinito.

---

## Modifica del funzionamento di un programma Capture in esecuzione

È possibile modificare dinamicamente il valore di uno o più parametri di funzionamento di Capture. Le modifiche non vengono salvate nella tabella IBMSNAP\_CAPPARMS, ma vengono utilizzate finché non si arresta il programma Capture o non vengono forniti nuovi valori.

### Informazioni su questa attività

**z/OS** **Linux UNIX Windows** È possibile modificare i seguenti parametri Capture mentre il programma Capture è in esecuzione:

- **autoprune**
- **autostop**
- **commit\_interval**
- **lag\_limit**
- **logreuse**
- **logstdout**
- **memory\_limit**
- **monitor\_interval**
- **monitor\_limit**
- **prune\_interval**
- **retention\_limit**

- `sleep_interval`
- `term`
- `trace_limit`

**Limitazione:** `z/OS` La quantità di memoria che il programma Capture può utilizzare per creare messaggi viene determinata all'avvio del programma Capture, in base al valore del parametro `memory_limit` e alla dimensione REGION specificata nel JCL. Il valore di `memory_limit` non può essere alterato con il programma Capture in esecuzione. Per modificare il valore, per prima cosa è necessario arrestare il programma Capture.

`System i` È possibile sovrascrivere i valori per i seguenti parametri facoltativi per un dato schema Capture:

- `CLNUPITV`
- `FRCFRQ`
- `MEMLMT`
- `MONLMT`
- `MONITV`
- `PRUNE`
- `RETAIN`
- `TRCLMT`

Quando si modificano i valori, gli effetti potrebbero non essere immediati per tutti i parametri.

### Procedure

Per modificare il funzionamento di un programma Capture in esecuzione, utilizzare uno dei metodi riportati di seguito:

Metodo	Descrizione
<b>Centro di replica</b>	Utilizzare la finestra di modifica parametri per l'esecuzione del programma Capture. Tale metodo consente di visualizzare i valori correnti dei parametri prima di modificarli. Per visualizzare la finestra, aprire il ramo <b>Operazioni</b> della struttura ad albero dell'oggetto, fare clic su <b>Server di controllo Capture</b> , fare clic con il tasto destro del mouse su un server di controllo Capture nel pannello dei contenuti e fare clic su <b>Modifica parametri</b> → <b>Esecuzione del programma Capture</b> .
<code>z/OS</code> <code>Linux UNIX Windows</code> Comando di sistema <code>asnccmd chgparms</code>	Questo metodo non visualizza i valori correnti dei parametri.
<code>System i</code> Comando di sistema <code>OVRDPRCAPA</code>	Utilizzare il comando <code>OVRDPRCAPA</code> (Sovrascrittura attributi DPR Capture) per modificare il funzionamento di un programma Capture in esecuzione.



---

## Modifica dei parametri di funzionamento salvati nella tabella IBMSNAP\_CAPPARMS

La tabella IBMSNAP\_CAPPARMS contiene i parametri di funzionamento salvati per il programma Capture. Quando si avvia il programma Capture, esso utilizza i valori di questa tabella a meno che questi non vengano sovrascritti temporaneamente utilizzando i parametri di avvio o mentre il programma è in esecuzione.


### Informazioni su questa attività

È consentita una sola riga nella tabella IBMSNAP\_CAPPARMS. Se si desidera modificare uno o più valori predefiniti, è possibile aggiornare le colonne anziché inserire le righe. Se si elimina la riga, il programma Capture verrà ancora avviato utilizzando i valori predefiniti, a meno che tali valori non vengano sovrascritti dai parametri di avvio.

Il programma Capture legge questa tabella solo durante l'avvio. Se si modifica la tabella di parametri di Capture quando il programma Capture è in esecuzione e si reinizializza il programma Capture ciò non influirà sul funzionamento del programma Capture.

### Procedure

Per modificare i parametri salvati nella tabella IBMSNAP\_CAPPARMS, utilizzare uno dei seguenti metodi:

Metodo	Descrizione
Centro di replica	Utilizzare la finestra Modifica parametri - Salvati. Per visualizzare la finestra, aprire il ramo <b>Operazioni</b> della struttura ad albero dell'oggetto, fare clic su <b>Server di controllo Capture</b> , fare clic con il tasto destro del mouse su un server di controllo Capture nel pannello dei contenuti e fare clic su <b>Modifica parametri</b> → <b>Salvati</b> .
 Comando di sistema CHGDPRCAPA	Utilizzare il comando CHGDPRCAPA (Modifica attributi DPR Capture) per modificare i parametri di funzionamento globali che vengono utilizzati dal programma Capture e sono memorizzati nella tabella IBMSNAP_CAPPARMS.


Le modifiche di parametro hanno effetto solo dopo che il programma Capture viene arrestato e riavviato.

---

## Arresto del programma Capture

È possibile arrestare il programma Capture per un determinato schema Capture. Quando si arresta il programma Capture, questo non cattura più i dati dall'origine.

### Informazioni su questa attività

 Se si sceglie di riorganizzare la tabella UOW e tutte le tabelle CD aperte quando il programma Capture è stato arrestato, il programma Capture necessita di tempo per la chiusura (non si chiude immediatamente).

### Procedure

Per arrestare il programma Capture, utilizzare uno dei metodi riportati di seguito:

Metodo	Descrizione
<b>Centro di replica</b>	Utilizzare la finestra Arresta Capture. Per visualizzare la finestra, aprire il ramo <b>Operazioni</b> della struttura ad albero dell'oggetto, fare clic su <b>Server di controllo Capture</b> , fare clic con il tasto destro del mouse su un server di controllo Capture nel pannello dei contenuti e fare clic su <b>Arresta Capture</b> .
<div style="background-color: #800000; color: white; padding: 2px; text-align: center; margin-bottom: 5px;">z/OS</div> <div style="background-color: #800000; color: white; padding: 2px; text-align: center; margin-bottom: 5px;">Linux UNIX Windows</div> Comando di sistema asncmd stop	Utilizzare questo comando per arrestare Capture.
<div style="background-color: #800000; color: white; padding: 2px; text-align: center; margin-bottom: 5px;">System i</div> Comando di sistema ENDDPRCAP	Utilizzare il comando ENDDPRCAP (Termine DPR Capture) per arrestare il programma Capture.

Se, durante l'eliminazione, si arresta o sospende il programma Capture, anche l'eliminazione viene sospesa. Quando si riprende o riavvia il programma Capture, l'eliminazione riprende in base al parametro **autoprun**.

Non è necessario arrestare il programma Capture per eliminare una registrazione. Disattivare sempre la registrazione prima di eliminarla.

---

## Reinizializzazione di Capture

Reinizializzare il programma Capture se si modifica qualche attributo degli oggetti registrati esistenti mentre tale programma è in esecuzione.

### Informazioni su questa attività

Ad esempio, è necessario reinizializzare il programma Capture se si modificano i valori CONFLICT\_LEVEL, CHGONLY, RECAPTURE, CHG\_UPD\_TO\_DEL\_INS nella tabella IBMSNAP\_REGISTER.

Per quanto riguarda Capture su System i, la reinizializzazione è inoltre necessaria per avviare l'acquisizione dei dati di un giornale non effettuata in precedenza.

### Procedure

Per reinizializzare il programma Capture, utilizzare uno dei metodi riportati di seguito:

Metodo	Descrizione
<b>Centro di replica</b>	Utilizzare la finestra per la nuova inizializzazione del programma Capture. Per visualizzare la finestra, aprire il ramo <b>Operazioni</b> della struttura ad albero dell'oggetto, fare clic su <b>Server di controllo Capture</b> , fare clic con il tasto destro del mouse su un server di controllo Capture nel pannello dei contenuti e fare clic su <b>Reinizializza Capture</b> .

Metodo	Descrizione
<p><b>z/OS</b></p> <p><b>Linux UNIX Windows</b></p> <p>Comando di sistema asnccmd reinit</p>	Utilizzare questo comando per reinizializzare Capture.
<p><b>System i</b></p> <p>Comando di sistema INZDPRCAP</p>	Utilizzare il comando INZDPRCAP (Inizializzazione DPR Capture) per inizializzare il programma Capture.

## Sospensione del programma Capture (Linux, UNIX, Windows, z/OS)

**z/OS** **Linux UNIX Windows**

È possibile sospendere il programma Capture per liberare risorse del sistema operativo durante periodi di picco senza distruggere l'ambiente di tale programma.

### Prima di iniziare

Deve essere avviato il programma Capture con lo specifico schema Capture.

### Informazioni su questa attività

È possibile inoltre sospendere il programma Capture anziché arrestarlo se non si desidera che tale programma si chiuda una volta terminato il lavoro in corso. Quando si indica la ripresa del programma Capture, non si richiede nuovamente l'avvio del sovraccarico di Capture.

**Importante:** non sospendere il programma Capture prima di eliminare un'origine replica. Disattivare, invece, quindi eliminare l'origine replica.

### Procedure

Per sospendere il programma Capture, utilizzare uno dei metodi riportati di seguito:

Metodo	Descrizione
<p><b>Centro di replica</b></p>	Utilizzare la finestra per la sospensione del programma Capture. Per visualizzare la finestra, aprire il ramo <b>Operazioni</b> della struttura ad albero dell'oggetto, fare clic su <b>Server di controllo Capture</b> , fare clic con il tasto destro del mouse su un server di controllo Capture nel pannello dei contenuti e fare clic su <b>Sospendi Capture</b> .
<p><b>z/OS</b></p> <p><b>Linux UNIX Windows</b></p> <p>Comando di sistema asnccmd suspend</p>	Utilizzare questo comando per sospendere Capture.

Se, durante l'eliminazione, si arresta o sospende il programma Capture, anche l'eliminazione viene sospesa. Quando si riprende o riavvia il programma Capture, l'eliminazione riprende in base al parametro **autoprune**.

---

## Ripresa di Capture (Linux, UNIX, Windows, z/OS)

z/OS

Linux UNIX Windows

Se si desidera avviare nuovamente la cattura dei dati, è necessario riprendere un programma Capture che è stato sospeso.

### Procedure

Per riprendere un programma Capture che è stato sospeso, utilizzare uno dei metodi riportati di seguito:

Metodo	Descrizione
<b>Centro di replica</b>	Utilizzare la finestra per la ripresa del programma Capture. Per visualizzare la finestra, aprire il ramo <b>Operazioni</b> della struttura ad albero dell'oggetto, fare clic su <b>Server di controllo Capture</b> , fare clic con il tasto destro del mouse su un server di controllo Capture nel pannello dei contenuti e fare clic su <b>Riprendi Capture</b> .
<p>z/OS</p> <p>Linux UNIX Windows</p> <p>Comando di sistema asnccmd resume</p>	Utilizzare questo comando per riprendere Capture.

Se, durante l'eliminazione, si arresta o sospende il programma Capture, anche l'eliminazione viene sospesa. Quando si riprende o riavvia il programma Capture, l'eliminazione riprende in base al parametro **autoprun**.

---

## Capitolo 10. Funzionamento del programma Apply per la replica SQL

Il funzionamento del programma Apply comprende attività quali l'avvio, l'arresto e l'utilizzo delle routine di chiusura ASNDONE e ASNLOAD.

---

### Avvio del programma Apply (Linux, UNIX, Windows, z/OS)

z/OS

Linux UNIX Windows

È possibile avviare un'istanza del programma Apply per iniziare ad applicare i dati alle destinazioni.

#### Prima di iniziare

Verificare quanto segue:

- Le connessioni sono configurate su tutti i server di replica necessari.
- Si dispone delle autorizzazioni appropriate.
- Le tabelle di controllo contenenti i dati di controllo e di origine per il qualificatore Apply desiderato sono state create.
- I programmi di replica sono stati configurati.
- **z/OS** L'utente collega manualmente il programma Apply a tutti i server necessari.
- **Linux UNIX Windows** Esiste un file della password per l'autenticazione dell'utente finale per i server remoti.

Inoltre, verificare che siano soddisfatte le seguenti condizioni:

- Esiste almeno una serie di richieste attiva per il qualificatore Apply e la serie di richieste contiene uno o più dei seguenti elementi:
  - Membro della serie di richieste
  - istruzione SQL
  - Procedure
- Tutte le tabelle di destinazione concentrate devono presentare una chiave di destinazione, ovvero una serie di colonne univoche, una chiave principale o un indice univoco, che il programma Apply utilizza per tracciare le modifiche replicate durante ciascun ciclo di Apply. Le tabelle CCD non concentrate non presentano chiavi principali o indici univoci.

#### Informazioni su questa attività

Quando si avvia il programma Apply, è anche possibile specificare i parametri di avvio.

#### Procedure

Per avviare il programma Apply:

Utilizzare uno dei metodi seguenti:

Opzione	Descrizione
Centro di replica	Utilizzare la finestra Avvia Apply. Per visualizzare la finestra, accedere alla cartella <b>Server di controllo Apply</b> nel ramo <b>Operazioni</b> della struttura ad albero degli oggetti e fare clic sulla cartella <b>Qualificatori Apply</b> . Nel pannello del contenuto, fare clic con il tasto destro del mouse sul qualificatore Apply che rappresenta il programma Apply che si desidera avviare e fare clic su <b>Avvia Apply</b> .
<div style="background-color: #800000; color: white; padding: 2px; text-align: center; margin-bottom: 5px;">z/OS</div> <div style="background-color: #800000; color: white; padding: 2px; text-align: center; margin-bottom: 5px;">Linux UNIX Windows</div> Comando di sistema asnapply	Utilizzare questo comando per avviare Apply.
<div style="background-color: #800000; color: white; padding: 2px; text-align: center; margin-bottom: 5px;">z/OS</div> Console z/OS o TSO	<p>In z/OS, è possibile avviare il programma Apply utilizzando JCL oppure come un'attività avviata dal sistema. È possibile specificare nuovi valori del parametro di richiamo quando si avvia un programma Apply con JCL.</p> <p>In z/OS, la lunghezza totale dei parametri che può essere specificata nel campo PARM= ha un limite di 100 byte. Per superare questo limite, adesso programmi di replica consentono di specificare tutti parametri aggiuntivi necessari nel set di dati SYSIN.</p> <p>Quando l'istruzione SYSIN DD è inclusa nel JCL di chiamata, il programma Apply concatena automaticamente ciò che viene specificato nel set di dati SYSIN con i parametri PARM=. È possibile specificare i parametri Apply solo nel set di dati SYSIN. Qualsiasi parametro LE deve essere specificato nel campo PARM= o in LE _CEE_ENVFILE=DD, seguito da una barra (/).</p> <p><b>Esempio:</b></p> <pre>L'asterisco /* indica una riga di commento // APP EXEC PGM=ASNAPP,PARM='LE/Parametri Apply' /* I parametri possono essere rappresentati da uno qualsiasi /* o nessuno dei parametri LE o Apply //SYSIN DD * /* parametri Apply aggiuntivi, uno o più /* parametri su ciascuna riga   APPLY_SERVER=DSN!! APPLY_SCHEMA=APPCAT   DEBUG=Y LOGSTDOUT=N</pre>
<div style="background-color: #800000; color: white; padding: 2px; text-align: center; margin-bottom: 5px;">Windows</div> Servizi Windows	È possibile creare un servizio di replica del DB2 in Windows per avviare automaticamente il programma Q Apply all'avvio del sistema.

Dopo avere avviato il programma Apply, quest'ultimo rimane in esecuzione (a meno che non venga utilizzato il parametro di avvio **copyonce**) finché non si verifica uno dei seguenti eventi:

- L'utente arresta il programma Apply utilizzando il Centro di replica o un comando.
- Il programma Apply non può connettersi al server di controllo Apply.
- Il programma Apply non può assegnare memoria per l'elaborazione.

Per verificare se il programma Apply è stato avviato, utilizzare uno dei seguenti metodi:

- **z/OS** Se si sta eseguendo in modalità batch, ricercare nella console z/OS o nella registrazione lavoro z/OS dei messaggi che indicano che il programma è stato avviato.
- Esaminare il file di log della diagnostica di Apply (*apply\_server.apply\_qualifier.APP.log* in z/OS e *db2instance.apply\_server.apply\_qualifier.APP.log* in Linux, UNIX e Windows) per individuare un messaggio che indica che il programma sta catturando le modifiche.
- Controllare la tabella IBMSNAP\_APPLYTRACE per individuare un messaggio che indica che il programma sta applicando le modifiche.
- Utilizzare la finestra Messaggi Apply nel Centro di replica per visualizzare un messaggio che indica che il programma è stato avviato. Per visualizzare la finestra, fare clic con il tasto destro del mouse sul qualificatore Apply nel pannello del contenuto che identifica il programma Apply di cui si desidera visualizzare i messaggi e selezionare **Prospetti** → **Messaggi Apply**.
- Utilizzare la finestra Controlla stato nel Centro di replica oppure il comando `asnacmd` per visualizzare lo stato dei thread di Apply. Per visualizzare la finestra, fare clic con il tasto destro del mouse sul qualificatore Apply nel pannello del contenuto che identifica il programma Apply che si desidera controllare e selezionare **Controlla stato**.

---

## Starting an Apply program (System i)

### System i

You can start an instance of the Apply program to begin applying data to your targets.

#### Prima di iniziare

Ensure that your system is set up correctly:

- Connections are configured to all replication servers.
- You have the proper authorization.
- The control tables are created.
- The replication programs are configured.

Also, make sure that the following conditions are met:

- At least one active subscription set exists for the Apply qualifier and that subscription set contains one or more of the following items:
  - Subscription-set member
  - SQL statement
  - Procedure
- All condensed target tables must have a target key, which is a set of unique columns, either a primary key or unique index, that the Apply program uses to track which changes it replicates during each Apply cycle. (Non-condensed CCD tables do not have primary keys or unique indexes.)

#### Procedure

To start an Apply program, use one of the following methods:

Method	Description
<b>STRDPRAPY system command</b>	Use the Start DPR Apply (STRDPRAPY) command to start an Apply program on your local system.
<b>Replication Center</b>	Use the Start Apply window. To open the window, open the <b>Apply Control Servers</b> folder in the <b>Operations</b> branch of the object tree and click the <b>Apply Qualifiers</b> folder. In the contents pane, right-click the Apply qualifier that represents the Apply program that you want to start and click <b>Start Apply</b> .

After you start the Apply program, it runs continuously unless one of the following conditions are true:

- You started the program with the COPYONCE(\*YES) startup parameter.
- You specified ALWINACT(\*NO) and there is no data to be processed.
- You stop the Apply program by using the Replication Center or a command.
- The Apply program cannot connect to the Apply control server.
- The Apply program cannot allocate memory for processing.

## Parametri di funzionamento predefiniti per il programma Apply

Quando si creano tabelle di controllo Apply, i valori predefiniti per i parametri di funzionamento del programma Apply vengono salvati nella tabella IBMSNAP\_APPPARMS.

I valori predefiniti sono illustrati in Tabella 9 e Tabella 10 a pagina 135.

z/OS

Linux UNIX Windows

Tabella 9. Impostazioni predefinite per i parametri operativi Apply (z/OS, Linux, UNIX, Windows)

Parametro operativo	Valore predefinito	Nome colonna nella tabella IBMSNAP_APPPARMS
apply_qual	Nessun valore predefinito	APPLY_QUAL
apply_path	Directory in cui è stato avviato Apply <sup>1</sup>	APPLY_PATH
caf	y <sup>5</sup>	non applicabile
control_server	DB2DBDFT <sup>2</sup>	non applicabile
copyonce	n <sup>3</sup>	COPYONCE
db2_subsystem	Nessun valore predefinito <sup>4</sup>	non applicabile
delay	6 secondi	DELAY
errwait	300 secondi	ERRWAIT
inamsg	y <sup>5</sup>	INAMSG
loadxit	n <sup>3</sup>	LOADXIT
logreuse	n <sup>3</sup>	LOGREUSE
logstdout	n <sup>3</sup>	LOGSTDOUT
notify	n <sup>3</sup>	NOTIFY
opt4one	n <sup>3</sup>	OPT4ONE
pwdfile	asnpwd.aut	non applicabile



Tabella 9. Impostazioni predefinite per i parametri operativi Apply (z/OS, Linux, UNIX, Windows) (Continua)

Parametro operativo	Valore predefinito	Nome colonna nella tabella IBMSNAP_APPPARMS
spillfile	disk <sup>6</sup>	SPILLFILE
sleep	y <sup>5</sup>	SLEEP
sqlerrcontinue	n <sup>3</sup>	SQLERRCONTINUE
term	y <sup>5</sup>	TERM
trlreuse	n <sup>3</sup>	TRLREUSE

**Nota:**

1. Se Apply viene avviato come un servizio Windows, il relativo percorso è sqllib\bin
2. Il server di controllo Apply è il valore della variabile di ambiente DB2DBDFT, se specificato. Solo per sistemi operativi Linux, UNIX e Windows.
3. no
4. Il nome del sottosistema DB2 può avere un massimo di quattro caratteri. Questo parametro è necessario. Il nome del sottosistema DB2 è applicabile solo a sistemi operativi z/OS.
5. yes
6. Sui sistemi operativi z/OS, il valore predefinito è MEM.

**System i**

Tabella 10. Impostazioni predefinite per i parametri facoltativi Apply (System i)

Parametro operativo	Descrizione di (*valore)
USER (*CURRENT)	L'utente che ha eseguito il collegamento al sistema.
JOBID (*LIBL/QZSNDPR)	Nome libreria prodotto/descrizione lavoro.
APYQUAL (*USER)	Nome utente attuale (da sopra).
CTLSVR (*LOCAL)	Nome server RDB locale.
TRACE (*NONE)	Non genera una traccia.
FULLREFPGM (*NONE)	Non esegue la routine di chiusura ASNLOAD.
SUBNFYPGM (*NONE)	Non esegue la routine di chiusura ASNDONE.
INACTMSG (*YES)	Quando il programma Apply inizia un periodo di inattività, genera il messaggio ASN1044, che descrive la durata di inattività del programma.
ALWINACT (*YES)	In attesa se non c'è nulla da elaborare.
DELAY (6)	Attende 6 secondi successivi a un ciclo Apply prima di eseguire nuovamente l'elaborazione.
RTYWAIT (300)	Attende 300 secondi prima di rieseguire un'operazione non riuscita.
COPYONCE (*NO)	Non termina dopo il completamento di un ciclo di copia, continua l'elaborazione.
TRLREUSE (*NO)	Non vuota la tabella IBMSNAP_APPLYTRAIL quando il programma Apply viene avviato.
OPTSNGSET (*NO)	Non ottimizza le prestazioni del programma Apply per l'elaborazione di una singola serie di richieste.

---

## Descrizioni dei parametri di funzionamento di Apply

Quando si avvia il programma Apply, è possibile selezionare parametri di avvio. Di seguito sono riportati i parametri di avvio e dei consigli sulla scelta dei valori per ciascun parametro.

Questi parametri si applicano a z/OS, Linux, UNIX e Windows a meno che sia specificato diversamente.

- “apply\_path”
- “apply\_qual” a pagina 137
- caf
- “control\_server” a pagina 137
- “copyonce” a pagina 138
- “db2\_subsystem (z/OS)” a pagina 139
- “delay” a pagina 139
- “errwait” a pagina 139
- “inamsg” a pagina 140
- “loadxit” a pagina 140
- “logreuse” a pagina 140
- “logstdout” a pagina 141
- “notify” a pagina 141
- “opt4one” a pagina 141
- “pwdfile” a pagina 141
- “sleep” a pagina 142
- “spillfile” a pagina 142
- “sqlerrcontinue” a pagina 143
- “term” a pagina 143
- “trlreuse” a pagina 144

### apply\_path

**Linux UNIX Windows** Valore predefinito: `apply_path=current_directory`

**Windows** Valore predefinito (servizio su Windows): `apply_path=sqllib\bin`

Il percorso di Apply è la directory in cui il programma Apply memorizza i propri file di lavoro e di registrazione. Per impostazione predefinita, il percorso Apply è la directory in cui viene avviato il programma. È possibile cambiare il percorso Apply per memorizzare i file di lavoro e registrazione in qualsiasi altra ubicazione (ad esempio, /home/db2inst/apply\_files su un sistema AIX). Tenere traccia della directory scelta in quanto potrebbe essere necessario passare a questa directory per accedere al file di log Apply.

È possibile specificare un nome di percorso o un HLQ (High Level Qualifier), come ad esempio //APPV9. Quando si utilizza un HLQ, vengono creati dei file sequenziali conformi alle convenzioni di denominazione dei file relative ai nomi dei file del set di dati sequenziali per z/OS. I set di dati sequenziali sono relativi all>ID utente che esegue il programma. Altrimenti, tali nomi di file sono simili a quelli memorizzati in un percorso di directory denominato esplicitamente, con

L'HLQ concatenato come prima parte del nome file. Ad esempio, sysadm.APPV9.filename. L'utilizzo di un HLQ potrebbe essere conveniente se si desidera che il log Apply ed i file LOADMSG vengano gestiti dal sistema (SMS).

**z/OS** Fare riferimento al lavoro SASNSAMP(ASNSTRA) per informazioni su come è possibile cambiare il percorso Apply.

**Importante:** Accertarsi che la directory scelta abbia spazio sufficiente per i file temporanei utilizzati dal programma Apply.

**Windows** **Avvio di istanze di Apply su un sistema Windows:** quando si avvia il programma Apply utilizzando il Centro di replica o il comando asnapply è necessario specificare il percorso Apply se si hanno due o più qualificatori Apply identici ad eccezione delle relative maiuscole. I nomi di file sui sistemi Windows non sono sensibili al maiuscolo/minuscolo. Ad esempio, si assuma di avere tre qualificatori Apply: APPLYQUAL1, ApplyQual1, applyqual1. Ognuna di queste istanze Apply deve essere avviata con un diverso **apply\_path** per impedire conflitti dei nomi dei file di log per ogni istanza del programma Apply.

## apply\_qual

È necessario specificare il qualificatore Apply per le serie di richieste che si desidera elaborare. (Il qualificatore Apply è stato definito quando è stata creata la serie di richieste). È possibile specificare un solo qualificatore per comando start.

**Importante:** il qualificatore Apply è sensibile alle maiuscole/minuscole e il valore che si immette deve corrispondere a quello della colonna APPLY\_QUAL nella tabella IBMSNAP\_SUBS\_SET.

Se si ha più di un qualificatore definito, è possibile avviare un'altra istanza del programma Apply. Ogni istanza del programma Apply che viene avviata, elaborerà diverse serie di richieste che sono rappresentate nello stesso server di controllo Apply. Ad esempio, si assuma di avere due serie di richieste definite e ciascuna serie ha un qualificatore Apply univoco: APPLY1 e APPLY2. È possibile avviare due istanze del programma Apply (una per ogni qualificatore Apply), e ogni istanza utilizza le tabelle di controllo sul server di controllo Apply denominato CNTRLSVR. Ogni istanza di Apply elabora le relative serie di richieste in modo indipendente, fornendo prestazioni migliori di quelle di una singola istanza di Apply che elabori tutte le serie.

**z/OS**

## caf

**Valore predefinito:** y

Il parametro di runtime **caf =y** specifica se il programma Apply sovrascriverà l'RRS (Recoverable Resource Manager Services) di connessione e verrà eseguito con CAF (Call Attach Facility) di connessione. L'opzione **caf =y** è il valore predefinito per il programma Apply.

## control\_server

**z/OS** **Valore predefinito:** Nessuno

**Linux UNIX Windows** **Valore predefinito:** il valore della variabile di ambiente DB2DBDFT, se disponibile

Il server di controllo Apply è il server su cui risiedono le definizioni di sottoscrizione e le tabelle di controllo Apply. Specificare solo un server di controllo per qualificatore Apply. Se non si specifica un valore, il programma Apply verrà avviato sul server predefinito. Il valore predefinito dipende dal sistema operativo.

**z/OS** In z/OS, è necessario specificare il parametro **control server**.

Se il programma Apply non può connettersi al server di controllo, segue l'azione impostata dal parametro **term**:

**term=y (valore predefinito)**

Il programma Apply termina.

**term=n**

Apply attende per la quantità di tempo impostata dal parametro **errwait**, quindi ritenta la connessione.

Il thread di lavoro Apply imposta il proprio stato su "in attesa del database" se non è in grado di connettersi al proprio server di controllo Apply e se il programma Apply è stato avviato con il parametro **term=n**. È possibile eseguire il comando di stato **asnacmd** o **MODIFY** su z/OS per controllare se il thread di lavoro Apply è in esecuzione ma non è in grado di connettersi al server di controllo.

Se il programma Apply non può connettersi ad altri server, emette un messaggio di errore e prosegue l'elaborazione.

## **copyonce**

**Valore predefinito:** **copyonce=n**

Il parametro **copyonce** determina il ciclo di copia per il programma Apply.

Quando si avvia il programma Apply utilizzando **copyonce=y**, questo elabora una sola volta ogni serie di richieste idonea e quindi termina. In tale caso, una serie di richieste è idonea per essere elaborata se si verifica una delle seguenti condizioni:

- La serie di richieste utilizza il relativo tempo, il tempo è trascorso e la serie di richieste viene attivata.
- La serie di richieste utilizza il tempo basato sull'evento, è attivo e l'evento è stato eseguito, ma il programma Apply non ha elaborato ancora la serie di richieste.

Di solito, si desidera avviare il programma Apply utilizzando **copyonce=n** in quanto si desidera che il programma Apply prosegua l'esecuzione e l'elaborazione delle richieste idonee.

Se si sta eseguendo il programma Apply da un ambiente di collegamento che è connesso occasionalmente alla rete, utilizzare **copyonce=y** anziché **copyonce=n**. Si potrebbe inoltre voler utilizzare **copyonce=y** se si sta eseguendo il programma Apply in un ambiente di prova.

**Suggerimento:** Utilizzare **sleep=n** anziché **copyonce=y** se si desidera che il programma Apply elabori più volte ogni serie di richieste, finché la serie è idonea

e i dati sono disponibili per la replica. **copyonce=y** elabora ogni serie solo una volta se sono presenti più dati da replicare.

## **db2\_subsystem (z/OS)**

Il parametro **db2\_subsystem** specifica il nome del sottosistema DB2 se si sta eseguendo Apply su z/OS. Il nome del sottosistema DB2 che viene immesso può avere un massimo di quattro caratteri. Non esiste alcun valore predefinito per questo parametro. Questo parametro è obbligatorio.

## **delay**

**Valore predefinito:** delay=6 secondi

Il parametro **delay** imposta una quantità di tempo in secondi in cui il programma Apply attende alla fine del ciclo Apply.

Per impostazione predefinita, durante la replica continua (ovvero, quando la serie di richieste usa **sleep=0** minuti), il programma Apply attende 6 secondi dopo che una serie di richieste è elaborata con esito positivo prima di rieseguirla. Utilizzare un valore delay diverso a zero per salvare i cicli CPU quando non è presente alcuna attività di database da replicare. Utilizzare un valore delay inferiore per bassa latenza.

**Nota:** Il parametro **delay** viene ignorato se è specificato **copyonce**.

## **errwait**

**Valore predefinito:** errwait=300 secondi (5 minuti)

Il parametro **errwait** specifica il numero di secondi che il programma Apply attende prima di rieseguire una serie di richieste dopo che un ciclo di richieste ha avuto esito negativo.

Per impostazione predefinita, il programma Apply attende 300 secondi prima di rieseguire una serie di richieste dopo che un ciclo di richieste ha avuto esito negativo. Si potrebbe voler utilizzare un valore inferiore in un ambiente di prova. Il valore minimo è 1 secondo. In un ambiente di produzione, considerare gli svantaggi prima di modificare il valore predefinito per questo parametro:

- Se si utilizza un valore inferiore, si potrebbero perdere cicli CPU se il programma Apply continua a rieseguire errori gravi. Ad esempio, verranno utilizzati cicli CPU non necessari se il programma Apply continua a provare a rieseguire una serie di richieste quando è presente un problema con una tabella di destinazione. Si potrebbe ricevere un elevato numero di messaggi nel file di log, se il programma Apply viene eseguito su z/OS, sulla console dell'operatore.
- Se si utilizza un valore maggiore, si potrebbe aumentare la latenza se il programma Apply deve attendere per rieseguire condizioni di errore transitorie. Ad esempio, si aumenterà la latenza se si utilizza un valore maggiore per il parametro **errwait** in quanto il programma Apply attende inutilmente dopo che rileva un errore di rete che potrebbe essere risolto rapidamente.

**Nota:** il parametro **errwait** viene ignorato se è specificato **copyonce**.

## inamsg

Valore predefinito: `inamsg=y`

Il parametro `inamsg` specifica se il programma Apply emette o meno un messaggio quando diviene inattivo.

Per impostazione predefinita, il programma Apply emette un messaggio quando diviene inattivo. Si potrebbe volere che il programma Apply non emetta un messaggio quando diventa inattivo in quanto i messaggi occupano molto spazio nel file di log Apply, specialmente se il programma Apply non attende molto tra un'elaborazione di serie di richieste e l'altra. Per disattivare questi messaggi, utilizzare `inamsg=n`.

## loadxit

Valore predefinito: `loadxit=n`

Il parametro `loadxit` specifica se il programma Apply aggiorna o meno le tabelle di destinazione utilizzando la routine di chiusura ASNLOAD.

Per impostazione predefinita, il programma Apply non utilizza la routine di chiusura ASNLOAD per aggiornare le tabelle di destinazione (`loadxit=n`). Utilizzare `loadxit=y` se si desidera che il programma Apply richiami la routine di chiusura ASNLOAD per aggiornare le tabelle di destinazione. Considerare di utilizzare la routine di chiusura ASNLOAD se è presente una grande quantità di dati da copiare sulle tabelle di destinazione durante un aggiornamento completo.

**z/OS** Su z/OS, la routine di chiusura ASNLOAD utilizza la procedura memorizzata DSNUTILS per la chiamata delle utilità DB2 necessarie per il caricamento della tabella di destinazione.

## logreuse

Valore predefinito: `logreuse=n`

Il programma Apply memorizza informazioni operative in un file di log. Il parametro specifica se eseguire l'aggiunta al file di log o sovrascriverlo.

**z/OS**  
Il nome del file di log è `control_server.apply_qualifier.APP.log`.

**Linux UNIX Windows**  
Il nome del file di log è `edb2instance.control_server.apply_qualifier.APP.log`.

Per impostazione predefinita, il programma Apply aggiunge messaggi al file di log (`logreuse=n`) ogni volta che si avvia tale programma. Mantenere l'impostazione predefinita se si desidera la cronologia dei messaggi emessi dal programma Apply. Nelle seguenti situazioni si potrebbe voler utilizzare `logreuse=y`, dove il programma Apply elimina la registrazione e la ricrea quando viene avviato:

- La registrazione sta assumendo dimensioni elevate e si desidera eseguirne la pulizia per salvare spazio.
- Non è necessaria la cronologia memorizzata nella registrazione.

## logstdout

**Valore predefinito:** logstdout=n

Il parametro **logstdout** è disponibile solo se si utilizza il comando `asnapply`; **logstdout** non è disponibile nel Centro di replica.

Il parametro **logstdout** specifica se il programma Apply invia messaggi di completamento (ASN10251) sia al file di log che all'output standard.

Per impostazione predefinita, il programma Apply non invia messaggi di completamento all'output standard (STDOUT). Se si specifica **logstdout=y**, il programma Apply invierà messaggi di completamento sia al file di log che all'output standard (STDOUT). Si potrebbe scegliere di inviare messaggi all'output standard se si sta eseguendo la risoluzione dei problemi o si sta controllando il funzionamento del programma Apply.

## notify

**Valore predefinito:** notify=n

Il parametro **notify** specifica se il programma Apply notifica la routine di chiusura ASNDONE dopo che elabora una sottoscrizione.

Per impostazione predefinita, il programma Apply non notifica la routine di chiusura ASNDONE dopo il completamento dell'elaborazione della sottoscrizione. Se si specifica **notify=y**, dopo che il programma Apply completa un ciclo di richieste, esso richiama ASNDONE per eseguire ulteriori elaborazioni, come esaminare le tabelle di controllo Apply o inviare messaggi e-mail.

## opt4one

**Valore predefinito:** opt4one=n

Il parametro **opt4one** specifica se l'elaborazione del programma Apply è ottimizzata per una serie di richieste.

**Nota:** il parametro **opt4one** viene ignorato se è specificato **copyonce**.

Per impostazione predefinita, il programma Apply è ottimizzato per diverse serie di richieste. Il programma Apply legge le informazioni dalle tabelle di controllo di replica all'inizio di ogni ciclo di copia. Se si dispone di una serie di richieste per il qualificatore Apply, avviare il programma Apply utilizzando **opt4one=y** in modo che tale programma memorizzi informazioni su colonne e membri della serie di richieste e le riutilizzi. Quando si ottimizza il programma Apply per una serie di richieste, tale programma utilizza meno CPU e si migliora la velocità di trasmissione.

**Importante:** quando si utilizza **opt4one=y** e si aggiunge un membro ad una serie o la si modifica, è necessario arrestare il programma Apply e avviarlo nuovamente in modo che esso applichi le modifiche nelle tabelle di controllo.

## pwdfile

**Valore predefinito:** pwdfile=asnpwd.aut

Se i dati vengono distribuiti ai server, è possibile memorizzare ID utente e password in un file di password codificate in modo che il programma Apply possa accedere ai dati sui server remoti.

## sleep

**Valore predefinito:** `sleep=y`

Il parametro **sleep** specifica se il programma Apply continua l'esecuzione in modalità di attesa o termina dopo che elabora le serie di richieste idonee.

Per impostazione predefinita, il programma Apply viene avviato con **sleep=y**. Esso ricerca le serie di richieste idonee. Se rileva una serie di richieste idonee, la elabora e procede ricercando un'altra serie idonea. Se ne individua altre, il programma Apply prosegue l'elaborazione. Quando non ne individua più, il programma Apply continua l'esecuzione in modalità di attesa ripristinando periodicamente tale ricerca. Di solito, si desidera avviare il programma Apply in questo modo in quanto si desidera che gli aggiornamenti vengano applicati nel tempo e ci si aspetta che il programma Apply sia attivo e in esecuzione.

**Nota:** il parametro **sleep** viene ignorato se è specificato **copyonce**.

Quando si avvia il programma Apply con **sleep=n**, tale programma ricerca le serie di richieste idonee e le elabora. Procede nell'elaborazione delle serie di richieste idonee finché non ne individua più e ripete l'elaborazione per le serie idonee finché non ci sono più dati da replicare; quindi il programma Apply termina. Di solito, si desidera utilizzare **sleep=n** in un ambiente mobile o di prova in cui si desidera che il programma Apply venga eseguito solo se rileva serie di richieste idonee e quindi si desidera che termini. Non si desidera che il programma Apply attenda in modalità di attesa e ripristini periodicamente la ricerca di ulteriori serie idonee. In tali ambienti, si desidera controllare quando il programma Apply viene eseguito piuttosto che doverlo eseguire senza fine.

**Suggerimento:** Utilizzare **copyonce=y** anziché **sleep=n** se si desidera elaborare ogni serie di richieste solo una volta.

## spillfile

**z/OS**

**Valore predefinito:** `spillfile=MEM`

**Linux UNIX Windows**

**Valore predefinito:** `spillfile=disk`

Il programma Apply richiama i dati dalle tabelle di origine e li inserisce in un file di trasferimento sul sistema in cui il programma è in esecuzione.

**z/OS**

Per impostazione predefinita, su z/OS il file di trasferimento è archiviato in memoria. Se si specifica di memorizzare il file di trasferimento su disco, il programma Apply utilizza le specifiche sull'istruzione ASNASPL DD per assegnare file di trasferimento. Se l'istruzione ASNASPL DD non è specificata, utilizza VIO.

**Linux UNIX Windows**

L'unica impostazione valida per **spillfile** è `disk` in quanto i file di trasferimento sono sempre su disco nell'ubicazione specificata dal parametro **apply\_path**.



## sqlerrcontinue

**Valore predefinito:** sqlerrcontinue=n

Il parametro **sqlerrcontinue** specifica l'azione intrapresa dal programma Apply in caso di determinati errori SQL.

Per impostazione predefinita, quando il programma Apply rileva errori SQL, arresta l'elaborazione di tale serie di richieste e genera un messaggio di errore. Di solito, si utilizza il valore predefinito nell'ambiente di produzione in uso.

Se si è in un ambiente di prova, è possibile che si verifichino alcuni errori SQL quando si inseriscono i dati nelle tabelle di destinazione. Talvolta tali errori sono accettabili, ma potrebbero provocare l'arresto del ciclo di richieste attuale. In tali situazioni, è possibile avviare il programma Apply utilizzando **sqlerrcontinue=y** in modo che ignori tali errori e non esegua il rollback dei dati replicati da tale ciclo. Se il programma Apply riceve un errore SQL nell'inserimento dei dati in una tabella di destinazione, verifica i valori nel file *apply\_qualifier.sqs*. Se rileva una corrispondenza, scrive i dettagli relativi all'errore su un file, *apply\_qualifier.err*, e procede l'elaborazione. Se il programma Apply rileva un errore SQL non presente nel file *apply\_qualifier.sqs*, arresta l'elaborazione della serie e passa a quella successiva.

Prima di avviare il programma Apply utilizzando l'opzione **sqlerrcontinue=y**, è necessario creare il file *apply\_qualifier.sqs* e memorizzarlo nella directory da cui si richiama tale programma. Nel file è possibile elencare fino ad un massimo di 20 valori di cinque byte, uno dopo l'altro. Se si modifica il contenuto di questo file quando il programma Apply è in esecuzione, arrestare il programma e riavviarlo in modo che riconosca i nuovi valori.

**Esempio:** si supponga che si desideri che il programma Apply continui l'elaborazione di una serie di richieste se una tabella di destinazione riceve il seguente errore (sqlstate/code):

**42704/-803**

Violazione di indice duplicato

Si crea un file di stato SQL che contiene il seguente stato SQL:

```
42704
```

Se lo stato SQL viene restituito quando si aggiorna la tabella di destinazione, il programma Apply applica le modifiche alle altre tabelle di destinazione all'interno della serie e crea un file di errore che indica sia l'errore che le righe rifiutate.

**Suggerimento:** verificare la colonna STATUS della tabella IBMSNAP\_APPLYTRAIL. Il valore 16 indica che il programma Apply ha elaborato la serie di richieste con esito positivo, ma si sono verificati alcuni degli errori consentiti, che sono stati definiti nel file *apply\_qualifier.sqs*.

## term

**Valore predefinito:** term=y

Il parametro **term** determina il comportamento del programma Apply qualora non sia in grado di connettersi al proprio server di controllo.

Per impostazione predefinita, il programma Apply termina se non riesce a connettersi.

Utilizzare **term=n** se si desidera che il programma Apply continui ad essere eseguito. Apply registra un errore, attende per la quantità di tempo impostata dal parametro **errwait**, quindi ritenta la connessione al proprio server di controllo.

il parametro **term** viene ignorato se è specificato **copyonce**.

## **trlreuse**

**Valore predefinito:** **trlreuse=n**

Il parametro **trlreuse** specifica se la tabella IBMSNAP\_APPLYTRAIL deve essere riutilizzata o meno (aggiunta a) o sovrascritta quando il programma Apply viene avviato.

Per impostazione predefinita, quando il programma Apply viene avviato, aggiunge voci alla tabella di traccia Apply. Questa tabella contiene la cronologia delle operazioni per tutte le istanze Apply sul server di controllo Apply. È un repository di statistiche di diagnostica e prestazioni. Mantenere l'impostazione predefinita se si desidera la cronologia degli aggiornamenti. Nelle seguenti situazioni si potrebbe volere che il programma Apply svuoti la tabella di traccia Apply quando viene avviato anziché eseguire aggiunte a questa (**trlreuse=y**):

- La tabella di traccia Apply sta assumendo dimensioni troppo elevate e si desidera eseguirne la pulizia per salvare spazio.
- Non è necessaria la cronologia memorizzata nella tabella.

**Suggerimento:** anziché utilizzare **trlreuse=y**, è possibile usare l'elaborazione SQL dopo che il programma Apply ha terminato con esito positivo una serie di richieste (dove **status=0**) per eliminare le righe dalla tabella di traccia Apply.

---

## **Metodi per modificare i parametri di funzionamento di Apply**

È possibile impostare i valori predefiniti dei parametri di funzionamento sui valori utilizzati nel proprio ambiente. È anche possibile ignorare questi valori predefiniti quando si avvia il programma Apply.

### **Impostazione di nuovi valori predefiniti nella tabella IBMSNAP\_APPPARMS**

La tabella IBMSNAP\_APPPARMS contiene i parametri che è possibile modificare per controllare il funzionamento del programma Apply. Dopo avere creato la tabella, essa conterrà i valori predefiniti del programma Apply.

### **Specifica dei valori per i parametri quando si avvia il programma Apply.**

È possibile specificare i valori per il programma Apply quando quest'ultimo viene avviato. I valori impostati durante l'avvio controllano il funzionamento di Apply per la sessione corrente, essi ignorano i valori dei parametri di funzionamento predefiniti ed eventuali valori contenuti nella tabella di parametri di Apply. Essi non aggiornano i valori nella tabella di parametri di Apply. Se non si modifica la tabella di parametri di Apply prima di avviare il programma Apply e non si specifica nessuno dei parametri facoltativi quando si avvia il programma Apply, per i parametri di funzionamento verranno utilizzati i valori predefiniti.

## Esempio

Presupporre che non si desidera utilizzare le impostazioni predefinite per **errwait** per il qualificatore ASNPROD di Apply. Aggiornare la tabella di parametri di Apply per il qualificatore Apply ASNPROD. Impostare l'intervallo **errwait** su 600 secondi.

```
update asn.ibmsnap_appparms set errwait=600 where apply_qual='ASNPROD'
```

---

## Modifica di parametri Apply salvati nella tabella IBMSNAP\_APPPARMS (z/OS, Linux, UNIX, Windows)

z/OS

Linux UNIX Windows

La tabella IBMSNAP\_APPPARMS contiene i parametri di funzionamento salvati per il programma Apply. Quando si avvia il programma Apply, esso utilizza i valori di questa tabella a meno che questi non vengano sovrascritti temporaneamente utilizzando parametri di avvio.

### Informazioni su questa attività

È consentita una sola riga per ogni qualificatore Apply. Se si desidera modificare uno o più valori predefiniti, è possibile aggiornare le colonne anziché inserire le righe. Se si elimina la riga, il programma Apply si avvia ancora utilizzando i valori predefiniti forniti, a meno che tali valori non vengano sovrascritti dai parametri di avvio.

Il programma Apply legge questa tabella solo durante l'avvio; quindi, è necessario arrestare e avviare il programma Apply se si desidera che venga eseguito con le nuove impostazioni. Se si modifica la tabella di parametri di Apply quando il programma Apply è in esecuzione ciò non influirà sul funzionamento di tale programma.

---

## Arresto del programma Apply

Quando si arresta il programma Apply, quest'ultimo smette di copiare i dati nelle tabelle di destinazione e aggiorna le tabelle di controllo. In tal modo, successivamente, potrà essere riavviato correttamente.

### Procedure

Per arrestare il programma Apply:

Utilizzare uno dei metodi seguenti:

Opzione	Descrizione
Centro di replica	Utilizzare la finestra Arresta Apply. Per visualizzare la finestra, accedere alla cartella <b>Server di controllo Apply</b> nel ramo <b>Operazioni</b> della struttura ad albero degli oggetti e fare clic sulla cartella <b>Qualificatori Apply</b> . Nel pannello del contenuto, fare clic con il tasto destro del mouse sul qualificatore Apply che rappresenta il programma Apply che si desidera arrestare e fare clic su <b>Arresta Apply</b> .

Opzione	Descrizione
<p><b>z/OS</b></p> <p>Linux UNIX Windows</p> <p>Comandi di sistema asnacmd stop</p>	Utilizzare questo comando per arrestare Apply.
<p><b>System i</b></p> <p>Comando di sistema ENDDPRAPY</p>	Utilizzare il comando ENDDPRAPY (Termine DPR Apply) per arrestare un programma Apply sul sistema locale.

## Modifica della routine di chiusura ASNDONE (z/OS, Linux, UNIX, Windows)

**z/OS**    **Linux UNIX Windows**

È possibile personalizzare la routine di chiusura ASNDONE sui sistemi operativi Linux, UNIX, Windows e z/OS per modificare il funzionamento del programma Apply una volta che questo ha finito l'elaborazione delle richieste.

### Informazioni su questa attività

Se si avvia il programma Apply con il parametro **notify=y**, tale programma richiama la routine di chiusura ASNDONE una volta che ha terminato l'elaborazione delle richieste, indipendentemente dall'esito positivo dell'elaborazione di queste. Di seguito sono riportati alcuni esempi su come modificare la routine di chiusura ASNDONE da utilizzare nel proprio ambiente di replica:

- Se viene rilevata una transazione rifiutata, utilizzare la routine di chiusura per esaminare la tabella UOW relativamente alle transazioni rifiutate e intraprendere altre azioni (ad esempio, inviare automaticamente e-mail all'operatore di replica, inoltrare un messaggio o generare un avviso).
- Utilizzare la routine di chiusura per disattivare una serie di richieste non riuscite in modo che il programma Apply eviti di rieseguire tale serie di richieste finché non viene risolto il problema. Per rilevare una serie di richieste non riuscite, modificare la routine di chiusura in modo da ricercare STATUS= -1 nella tabella IBMSNAP\_APPLYTRAIL. Per disattivare la serie di richieste, configurare la routine di chiusura in modo da impostare ACTIVATE=0 nella tabella IBMSNAP\_SUBS\_SET.
- Utilizzare la routine di chiusura per manipolare i dati una volta applicata ad ogni serie di richieste (in alternativa, è possibile definire istruzioni di elaborazione di run-time utilizzando istruzioni SQL o procedure memorizzate eseguite prima o dopo che il programma Apply elabora una determinata serie di richieste).

### Procedure

Per utilizzare una versione modificata della routine di chiusura ASNDONE di esempio:

1. Modificare la routine ASNDONE in modo da soddisfare ai propri requisiti.
  - **z/OS** Consultare la sezione PROLOG del programma di esempio SASNSAMP(ASNDONE).

- **Linux UNIX Windows** Consultare la sezione PROLOG del programma di esempio (\sqlib\samples\repl\asndone.smp) per informazioni su come modificare questa routine di chiusura.
2. Compilare, collegare ed eseguire il bind del programma e porre l'eseguibile nella directory appropriata.
  3. Avviare il programma Apply con il parametro **notify=y** per richiamare la routine di chiusura ASNDONE.

---

## Modifying the ASNDONE exit routine (System i)

### System i

You can customize the ASNDONE exit routine on System i operating systems to modify the behavior of the Apply program after it finishes processing subscriptions.

#### Informazioni su questa attività

If you start the Apply program with the SUBNFYPGM parameter set to the name of the ASNDONE exit routine, the Apply program calls the ASNDONE exit routine after it finishes processing subscriptions, regardless of whether the subscriptions were processed successfully. The following list describes some examples of how you might modify the ASNDONE exit routine to use it in your replication environment:

- Use the exit routine to examine the UOW table for rejected transactions and initiate further actions (for example, send e-mail automatically to the replication operator, issue a message, or generate an alert) if a rejected transaction is detected.
- Use the exit routine to deactivate a failed subscription set so that the Apply program avoids retrying that subscription set until it is fixed. To detect a failed subscription set, modify the exit routine to look for STATUS= -1 in the IBMSNAP\_APPLYTRAIL table. To deactivate the subscription set, configure the exit routine so that it sets ACTIVATE=0 in the IBMSNAP\_SUBS\_SET table.
- Use the exit routine to manipulate data after it is applied for each subscription set. (You can also define run-time processing statements by using SQL statements or stored procedures that run before or after the Apply program processes a specific subscription set.)

#### Procedure

To use a modified version of the ASNDONE sample exit routine:

1. Modify the ASNDONE exit routine to meet your requirements. Tabella 11 indicates where you can find the source code for this routine in C, COBOL, and RPG languages:

*Tabella 11. Source code for ASNDONE*

Compiler language	Library name	Source file name	Member name
C	QDP4	QCSRC	ASNDONE
COBOL	QDP4	QCBLESRC	ASNDONE
RPG	QDP4	QRPGLESRC	ASNDONE

When modifying the program, consider these activation group concerns:

**If the program is created to run with a new activation group**

The Apply program and the ASNLOAD program will not share SQL resources, such as relational database connections and open cursors. The activation handling code in the System i operating system frees any resources allocated by the ASNLOAD program before control is returned to the Apply program. Additional resource is used every time that the Apply program calls the ASNLOAD program.

**If the program is created to run in the caller's activation group**

It shares SQL resources with the Apply program. Design the program so that you minimize its impact on the Apply program. For example, the program might cause unexpected Apply program processing if it changes the current relational database connection.

**If the program is created to run in a named activation group**

It does not share resources with the Apply program. Use a named activation group to avoid the activation group overhead every time the ASNLOAD program is called. Run-time data structures and SQL resources can be shared between invocations. Application clean-up processing is not performed until the Apply program is ended, so design the subscription notify program to ensure that it does not cause lock contention with the Apply program by leaving source tables, target tables, or control tables locked when control is returned to the Apply program.

2. Compile, link, and bind the program, and place the executable in the appropriate directory.
3. Start the Apply program and specify the name of the ASNDONE program by using the parameter SUBNFYPGM on the STRDPRAPY command.

For example, if the program is named ASNDONE\_1 and resides in library APPLIB, use the following command:

```
SUBNFYPGM(APPLIB/ASNDONE_1)
```

---

## Aggiornamento delle tabelle di destinazione utilizzando la routine di chiusura ASNLOAD

È possibile utilizzare la routine di chiusura ASNLOAD per eseguire un aggiornamento completo delle tabelle di destinazione in modo più efficace del metodo normale di caricamento dei dati nelle destinazioni del programma Apply.

Per impostazione predefinita, il programma Apply non utilizza la routine di chiusura ASNLOAD quando esegue un aggiornamento completo per ciascuna tabella di destinazione in una serie di richieste. Esegue una selezione completa nella tabella di origine, trasferisce i dati in un file di trasferimento sul server dove è in esecuzione il programma Apply e utilizza le istruzioni INSERT per inserire dati nella tabella di destinazione. Se si dispone di tabelle di origine di grandi dimensioni, si potrebbe voler utilizzare invece la routine di chiusura ASNLOAD.

La routine di chiusura di esempio differisce su ciascuna piattaforma DB2 per sfruttare le opzioni del programma di utilità offerte su tale piattaforma:

**z/OS**

**Linux UNIX Windows**

La routine di chiusura ASNLOAD è fornita come una routine di chiusura di esempio sia in formato origine che in formato compilato.

ASNLOAD è distribuita solo in formato origine.

Se si verifica un errore quando il programma Apply richiama la routine di chiusura ASNLOAD, tale programma emette un messaggio, arresta l'elaborazione della serie di richieste attuale ed elabora la serie di richieste successiva.

## Aggiornamento di tabelle di destinazione con la routine di chiusura ASNLOAD (Linux, UNIX, Windows)

Linux UNIX Windows

È possibile utilizzare la routine di chiusura ASNLOAD per aggiornare le tabelle di destinazione in modo più efficace sui sistemi operativi Linux, UNIX e Windows. È possibile inoltre modificare la routine prima di utilizzarla.

### Prima di iniziare

- La tabella di destinazione deve contenere solo colonne che fanno parte dell'associazione di replica.
- L'ID utente che esegue Apply deve essere l'ID utente dell'istanza DB2 su cui viene eseguita la routine ASNLOAD. Ad esempio, su Linux e UNIX, accertarsi che sia l'ID utente Apply che l'istanza DB2 siano membri di un gruppo comune. Quindi, impostare i bit di autorizzazione per la directory di avvio Apply in modo da fornire accesso in scrittura per l'istanza DB2 utilizzando il comando `chmod 775`.

### Restrizioni

La routine di chiusura ASNLOAD opera con i programmi di utilità EXPORT, IMPORT e LOAD, compresa la funzione LOAD FROM CURSOR. LOAD FROM CURSOR è l'opzione predefinita utilizzata dalla routine di chiusura ASNLOAD se l'origine di un membro di una serie di richieste è un nickname o se il database di destinazione è lo stesso del database di origine. LOAD FROM CURSOR può essere inoltre utilizzata con le origini dati DB2 se sono state eseguite le seguenti azioni:

- Nel database di destinazione è stato creato un nickname per la tabella di origine.
- Le colonne nella tabella IBMSNAP\_SUBS\_MEMBR per il membro della serie di richieste sono state impostate ad indicare che deve essere utilizzata la funzione LOAD FROM CURSOR. Il valore di tali colonne può essere impostato utilizzando il Centro di replica:
  - La colonna LOADX\_TYPE deve essere impostata ad indicare che verrà utilizzata la funzione LOAD FROM CURSOR.
  - Le colonne LOADX\_SRC\_N\_OWNER e LOADX\_SRC\_N\_TABLE devono specificare le informazioni del nickname di origine per il membro della serie di richieste che include la tabella di origine.

### Informazioni su questa attività

Quando si richiama la routine di chiusura di esempio, questa per impostazione predefinita sceglie il programma di utilità da usare in base al server di origine, al server di destinazione e all'ambiente di runtime. La routine può utilizzare la funzione di utilità DB2 EXPORT con la funzione di utilità DB2 IMPORT o la funzione di utilità DB2 LOAD oppure può utilizzare la funzione di utilità LOAD FROM CURSOR.

È possibile utilizzare la routine di chiusura compilata, configurarne il funzionamento personalizzando la configurazione di replica oppure è possibile personalizzare il codice di chiusura stesso. È possibile personalizzare la configurazione di replica aggiornando le colonne nella tabella IBMSNAP\_SUBS\_MEMBR o aggiornando un file di configurazione di esempio (asnload.ini).

Per utilizzare la routine ASNLOAD come fornita, avviare il programma Apply utilizzando il parametro **loadxit=y**.

### Procedure

Per utilizzare una versione modificata della routine di chiusura ASNLOAD:

1. Modificare la routine ASNLOAD in modo da soddisfare i requisiti del sito. Per informazioni su come modificare questa routine di chiusura, fare riferimento alla sezione PROLOG del programma di esempio (\sqllib\samples\repl\asnload.smp).

**Importante:** L'origine di esempio utilizza combinazioni ID utente e password dal file asnload.ini. Se il file asnload.ini non ha un ID utente e una password per un determinato server oppure se il file asnload.ini non è disponibile, la routine di chiusura cercherà di eseguire il collegamento senza i parametri **user** o **using**.

2. Compilare, collegare ed eseguire il bind del programma e porre l'eseguibile nella directory appropriata.
3. Impostare LOADX\_TYPE su 2 per i membri che verranno popolati utilizzando il codice fornito.
4. Avviare il programma Apply con il parametro loadxit=y per richiamare la routine di chiusura ASNLOAD.

La routine di chiusura ASNLOAD genera i seguenti file nella directory **apply\_path** per l'istanza Apply che ha richiamato la routine di chiusura ASNLOAD:

#### **asnload** *apply\_qualifier.trc*

Questo file contiene informazioni di traccia se la traccia è attiva. La routine di chiusura ASNLOAD crea questo file. Se il file esiste, le informazioni vengono aggiunte al file.

#### **asnload** *apply\_qualifier.msg*

Questo file contiene generali malfunzionamenti della routine di chiusura, messaggi di avvertenza e informativi comprendenti statistiche di caricamento. La routine di chiusura ASNLOAD crea questo file. Se il file esiste, le informazioni vengono aggiunte al file.

#### **asnaEXPT** *apply\_qualifier.msg*

Questo file contiene messaggi di errore, di avvertenza o informativi emessi dal programma di utilità DB2 EXPORT. La routine di chiusura ASNLOAD crea questo file. Se il file esiste, le informazioni vengono aggiunte al file.

#### **asnaIMPT** *apply\_qualifier.msg*

Questo file contiene messaggi di errore, di avvertenza o informativi emessi dal programma di utilità DB2 IMPORT. La routine di chiusura ASNLOAD crea questo file. Se il file esiste, le informazioni vengono aggiunte al file.

#### **asnaLOAD** *apply\_qualifier.msg*

Questo file contiene messaggi di errore, di avvertenza o informativi emessi dal programma di utilità DB2 LOAD. La routine di chiusura ASNLOAD crea questo file. Se il file esiste, le informazioni vengono aggiunte al file.



## Aggiornamento delle tabelle di destinazione tramite la routine di chiusura ASNLOAD (z/OS)

z/OS

È possibile utilizzare la routine di chiusura ASNLOAD per aggiornare le tabelle di destinazione in modo più efficiente sui sistemi operativi z/OS. È anche possibile modificare la routine prima di utilizzarla.

### Prima di iniziare

La tabella di destinazione deve contenere solo colonne che fanno parte dell'associazione di replica.

### Informazioni su questa attività

La routine di chiusura ASNLOAD esegue la chiamata dell'utilità LOAD FROM CURSOR disponibile in Utilities Suite DB2 V7 (o successiva). L'utilità effettua trasferimenti basati sul cursore per l'ottenimento di dati dall'origine e carica i dati nella destinazione.

La routine di chiusura ASNLOAD utilizza LOAD con LOG NO e azzerava lo stato di COPYPEND nello spazio tabella. È possibile modificare il codice di origine ASNLOAD di esempio, per la modifica delle opzioni di caricamento. L'origine è composta da due file di intestazione e tre programmi C++.

Per utilizzare la routine ASNLOAD come fornita, avviare il programma Apply con il parametro **loadxit=y**.

### Procedure

Per utilizzare una versione modificata della routine di chiusura ASNLOAD:

1. Modificare la routine per soddisfare i requisiti del proprio sito. Per informazioni su come modificare questa routine di chiusura, consultare la sezione PROLOG del programma di esempio SASNSAMP (ASNLOAD).
2. Compilare, collegare ed eseguire il bind del programma e porre l'eseguibile nella directory appropriata.
  - a. Assicurarsi che vengano soddisfatte le seguenti condizioni:
    - DB2 Universal Database per z/OS e OS/390 Versione 7 o successiva, con supporto per le utilità, sia installato.
    - La procedura memorizzata DSNUTILS sia in esecuzione. DSNUTILS deve essere eseguito in un ambiente WLM. Per ulteriori informazioni sull'utilizzo di DSNUTILS, consultare il manuale *DB2 for z/OS V8 Utility Guide and Reference*.
  - b. Utilizzare il file zmak di esempio (SASNSAMP(ASNCMPLD)) per la compilazione e la modifica del collegamento del programma di chiusura ASNLOAD dell'utente in USS.
  - c. Effettuare il bind della routine di chiusura ASNLOAD con DSNUTILS e il pacchetto Apply. L'ASNLOAD di esempio esegue un caricamento con LOG NO, quindi ripara lo spazio tabella, impostando nocopypend. Non esegue il backup degli spazi tabella. Per impostazione predefinita, ASNLOAD crea due file temporanei nell'ID utente su cui è in esecuzione l'istanza del programma Apply, a meno che venga specificato il parametro **apply\_path** con l'opzione **APPLY\_PATH=//** per tale istanza di Apply. In tal caso,

verranno creati due file temporanei nel qualificatore di alto livello specificato in APPLY\_PATH. La routine crea, inoltre, un file contenente tutte le informazioni relative al caricamento.

3. Impostare loadx\_type = 2 per i membri che verranno popolati utilizzando il codice fornito.
4. Avviare il programma Apply con il parametro **loadxit=y** per richiamare la routine di chiusura ASNLOAD.

La routine di chiusura ASNLOAD genera i seguenti file nella directory **apply\_path** o qualificatore di alto livello per l'istanza di Apply che ha richiamato la routine di chiusura ASNLOAD:

*idutente.qual\_apply.LOADMSG*

Questo file contiene messaggi di operazioni non riuscite, avvisi e messaggi informativi, comprese le statistiche di caricamento. La routine di chiusura ASNLOAD crea questo file. Se il file esiste, le informazioni vengono aggiunte al file.

*idutente.qual\_apply.LOADTRC*

Questo file contiene informazioni di traccia se la traccia è attiva. La routine di chiusura ASNLOAD crea questo file. Se il file esiste, le informazioni vengono aggiunte al file.

## Personalizzazione del funzionamento della routine di chiusura ASNLOAD (z/OS, Linux, UNIX, Windows)

z/OS

Linux UNIX Windows

Oltre a personalizzare il codice di chiusura stesso, è possibile personalizzare il funzionamento della routine di chiusura ASNLOAD aggiornando le colonne nella tabella IBMSNAP\_SUBS\_MEMBR o aggiornando un file di configurazione.

### Uso della tabella IBMSNAP\_SUBS\_MEMBR per impostare le opzioni ASNLOAD

È possibile utilizzare colonne nella tabella IBMSNAP\_SUBS\_MEMBR per personalizzare il funzionamento della routine di chiusura ASNLOAD.

#### Informazioni su questa attività

Utilizzare la colonna LOADX\_TYPE per specificare un'opzione di caricamento. I valori validi per LOADX\_TYPE sono:

#### null (valore predefinito)

z/OS

Utilizzare il programma di utilità LOAD FROM CURSOR.

Linux UNIX Windows

La routine di chiusura ASNLOAD determina l'utilità più appropriata (opzione 3, 4 o 5).

- 1 Non richiamare la routine di chiusura ASNLOAD per questo membro.  
Impostare LOADX\_TYPE su 1 se non si desidera che la routine di chiusura ASNLOAD venga richiamata per tale membro.
- 2 Fornire la propria logica di chiusura.  
Se si desidera fornire la propria logica nella routine di chiusura ASNLOAD, impostare LOADX\_TYPE su 2 per quei membri di serie di

richieste che si desidera vengano popolati dalla routine di chiusura ASNLOAD. Se si imposta LOADX\_TYPE su 2, ma non si fornisce una logica di chiusura, la chiusura avrà esito negativo.

### 3 Utilizzare il programma di utilità LOAD FROM CURSOR.

**Linux UNIX Windows** La funzione LOAD FROM CURSOR richiede un'istruzione SELECT per estrarre i dati che devono essere caricati sulla tabella di destinazione (la tabella di destinazione deve risiedere in un database locale). Questa istruzione può fare riferimento ad una tabella DB2 o ad un nickname e l'impostazione deve essere come segue:

Se si sta eseguendo la replica da un'origine non IBM su una tabella DB2 dove il nickname di origine registrato si trova su un database diverso da quello di destinazione oppure se si sta eseguendo la replica da una tabella DB2 su un'altra tabella DB2 e il database di origine è diverso da quello di destinazione, è necessario attenersi alla seguente procedura:

1. Creare un nickname per la tabella(e) di origine nel database del server di destinazione.
2. Aggiornare le colonne del proprietario del nickname e del nome di tabella (LOADX\_SRC\_N\_OWNER e LOADX\_SRC\_N\_TABLE) della tabella IBMSNAP\_SUBS\_MEMBR.

Se si sta eseguendo la replica da una tabella DB2 su un'altra tabella DB2 e il database di origine e destinazione è lo stesso oppure se si sta eseguendo la replica da un'origine non IBM su una tabella DB2 dove il nickname di origine registrata si trova sullo stesso database di quello di destinazione, non è necessario eseguire ulteriori azioni per utilizzare il programma di utilità LOAD FROM CURSOR.

#### **Linux UNIX Windows** 4

Utilizzare una combinazione dei programmi di utilità EXPORT e LOAD.

#### **Linux UNIX Windows** 5

Utilizzare una combinazione dei programmi di utilità EXPORT e IMPORT.

## **Uso del file di configurazione per ASNLOAD (Linux, UNIX, Windows)**

#### **Linux UNIX Windows**

È possibile utilizzare un file di configurazione facoltativo per configurare l'input sulla routine di chiusura ASNLOAD. Questo file non è richiesto per l'esecuzione di ASNLOAD.

### **Informazioni su questa attività**

Il file di configurazione deve avere il nome `asnload.ini`. La routine di chiusura ASNLOAD ricerca questo file di configurazione facoltativo nella directory specificata dal parametro **apply\_path**.

### **Procedure**

Per utilizzare il file di configurazione ASNLOAD:

1. Modificare il file di esempio `sqlib/samples/repl/asnload.ini`.
2. Memorizzare il file nella directory specificata dal parametro **apply\_path** per l'istanza Apply che ha richiamato la routine di chiusura ASNLOAD.

# Refreshing target tables with the ASNLOAD exit routine (System i)

## System i

You can use the ASNLOAD exit routine to refresh target tables more efficiently on System i. You can also modify the routine before using it.

### Prima di iniziare

- The target-table columns must match both the order and data type of the source tables.
- The target table can only contain columns that are part of the replication mapping.

### Informazioni su questa attività

For example, if you are copying every row and every column from a source table to a target table, you can design a full-refresh exit routine that uses a Distributed Data Management (DDM) file and the Copy File (CPYF) CL command to copy the entire file from the source table to the target table.

To use the ASNLOAD exit routine as provided, start the Apply program and specify the FULLREFPGM parameter.

### Procedure

To use a modified version of the ASNLOAD exit routine:

1. Modify the ASNLOAD exit routine to meet your site's requirements. See the PROLOG section of the sample program for information about how to modify this exit routine. The source is available in C, COBOL, and RPG languages, as shown in Tabella 12.

Tabella 12. Source code for ASNLOAD

Compiler language	Library name	Source file name	Member name
C	QDP4	QCSRC	ASNLOAD
COBOL	QDP4	QCBLESRC	ASNLOAD
RPG	QDP4	QRPGLESRC	ASNLOAD

2. Compile, link, and bind the program and place the executable in the appropriate directory. To avoid interference with the Apply program, compile the exit routine so that it uses a new activation group (not the activation group of the caller).

You can compile the exit routine with a named activation group or with a new activation group. To get better performance, use a named activation group. With a named activation group, the exit routine must commit or roll back changes as needed. The Apply program will not cause changes to be committed or rolled back (unless it ends). The exit routine should either explicitly commit changes, or it should be compiled to implicitly commit changes when it completes. Any uncommitted changes when the exit routine completes are not committed until either:

- The Apply program calls another exit routine with the same activation group.
- The job started for the Apply program ends.

3. Start the Apply program with the FULLREFPGM parameter set to the name of the ASNLOAD program. When you start the Apply program, it uses the ASNLOAD exit routine that you specified. If you want it to use another ASNLOAD exit routine, end the Apply program and start it again.

When you run the ASNLOAD exit routine, it refreshes all the target tables, table by table.



---

## Capitolo 11. Funzionamento dei programmi di replica (z/OS)

z/OS

Gli argomenti seguenti illustrano il funzionamento dei programmi di replica sul sistema operativo z/OS.

---

### Utilizzo di attività di sistema per il funzionamento dei programmi di replica

z/OS

È possibile utilizzare attività di sistema per il funzionamento dei programmi Capture e Apply, nonché per Replication Alert Monitor.

#### Procedure

Per utilizzare le attività avviate dal sistema per il funzionamento dei programmi di replica, utilizzare questo esempio del programma Capture:

1. Creare una procedura *nomeproc* in PROCLIB.
2. Creare una voce nella classe RACF STARTED per *nomeproc*. Questa voce associa *nomeproc* all'ID utente RACF da utilizzare per l'avvio del programma Capture. Assicurarsi che l'autorizzazione DB2 necessario sia stata concessa a questo ID utente prima di avviare il programma Capture.
3. Dalla console di sistema MVS, eseguire il comando start *nomeproc*.

La seguente procedura di esempio è relativa al programma Capture:

```
//CAPJAYC PROC
//ASNCAP EXEC PGM=ASNCAP,REGION=M,
//PARM='V71A autostop LOGSTDOUT startmode=COLD
//capture_schema=JAY logreuse'
//STEPLIB DD DISP=SHR,DSN=DPROPR.ASN81 .SASNLOAD
//DD DISP=SHR,DSN=SYS1.SCEERUN
//DD DISP=SHR,DSN=DSN7.SDSNLOAD
//CEEDUMP DD SYSOUT=
//SYSPRINT DD SYSOUT=
//SYSTEM DD DUMMY
//
```

---

### Utilizzo di JCL per il funzionamento dei programmi di replica

z/OS

Su z/OS, è possibile utilizzare JCL per avviare, arrestare e modificare i programmi di replica. Ciò permette all'utente di salvare gli script, nel caso in cui l'operazione venga eseguita ripetutamente.

#### Informazioni su questa attività

La libreria di esempi della replica SQL contiene JCL e script di esempio.

**suggerimento:** Copiare i lavori dalla libreria SASNSAMP in una libreria diversa prima di eseguire delle modifiche. Consultare la directory di programma per un elenco completo dei lavori di esempio rilevati nella libreria SASNSAMP.

## Procedure

Per il funzionamento dei programmi di replica con JCL:

1. Avvio dei programmi di replica.

Opzione	Descrizione
<b>Avviare il programma Capture con un lavoro batch.</b>	Preparare JCL per z/OS specificando i parametri di chiamata facoltativi appropriati nel campo PARM del lavoro batch ASNSTRC. Eseguire il lavoro da TSO o dalla console z/OS. È possibile trovare il lavoro nella libreria di esempi SASNSAMP.
<b>Avviare il programma Apply con un lavoro batch.</b>	Preparare JCL per z/OS specificando i parametri di chiamata facoltativi appropriati nel campo PARM del lavoro batch ASNSTRA. Eseguire il lavoro da TSO o dalla console z/OS. È possibile trovare il lavoro nella libreria di esempi SASNSAMP.
<b>Avviare Replication Alert Monitor con un lavoro batch.</b>	Preparare JCL per z/OS specificando i parametri di chiamata facoltativi appropriati nel campo PARM del lavoro batch ASNSTRM. Eseguire il lavoro da TSO o dalla console z/OS. È possibile trovare il lavoro nella libreria di esempi SASNSAMP.
<b>Avviare Replication Alert Monitor con JCL.</b>	Preparare il JCL per z/OS specificando i parametri di chiamata appropriati nel campo PARM del lavoro Replication Alert Monitor. Personalizzare il JCL in modo da soddisfare i requisiti del proprio sito. Un esempio di un JCL di chiamata nella libreria SASNSAMP(ASNMON#) è stato incluso in Replication Alert Monitor per z/OS.  Un esempio di questa riga nel JCL di chiamata è: <pre>//monasn EXEC PGM=ASNMON,PARM='monitor_server=DSN                                 monitor_qual=monqual'</pre> in cui DSN è il nome di un sottosistema e monqual è il qualificatore monitor.

2. Opzionale: Modificare i programmi di replica già avviati.

Dopo aver avviato il programma Capture, Apply o il programma Replication Alert Monitor, è possibile utilizzare il comando MODIFY per l'arresto del programma o per l'esecuzione delle attività correlate. È necessario eseguire il comando MODIFY da una console MVS. È possibile utilizzare l'abbreviazione F, come illustrato nell'esempio di sintassi seguente:

```
▶▶—F—nomelavoro—,—| Parametri |—————▶▶
```

F *nomelavoro* , sostituisce il nome effettivo del comando: asnacmd, asncmd, o asnmcmd. Ad esempio, per arrestare il programma Capture, utilizzare il comando seguente:

```
F capjfa,stop
```

Per informazioni su MODIFY, consultare i comandi di sistema MVS> z/OS.

## Avvio del programma Apply su z/OS con JCL

z/OS



È possibile avviare il programma Apply su z/OS modificando ed eseguendo uno script di esempio preparato. dalla directory degli esempi.

### Procedure

Per avviare il programma Apply su z/OS con JCL:

1. Preparare il JCL per z/OS specificando i parametri di chiamata appropriati nel campo PARM del lavoro Apply.
2. Personalizzare il JCL in modo da soddisfare i requisiti del proprio sito.

Per sistemi operativi z/OS, un esempio di questa riga nel JCL di chiamata è:

```
//apysn EXEC PGM=ASNAPPLY,PARM='control_server=CTLDB1
                                DB2_SUBSYSTEM=DSN
                                apply_qual=myqual spillfile=disk'
```

Per sistemi operativi UNIX e Windows, un esempio di questa riga nel JCL di chiamata è:

```
//apysn EXEC PGM=ASNAPPLY,PARM='control_server=CTLDB1
                                apply_qual=myqual spillfile=disk'
```

3. Inviare il JCL da TSO o dalla console MVS.

---

## Lavorare con i programmi di replica SQL in esecuzione mediante il comando MVS MODIFY

### z/OS

Dopo aver avviato il programma Capture, Apply o Replication Alert Monitor, è possibile utilizzare il comando MODIFY per l'arresto del programma, oppure per l'esecuzione di attività correlate.

### Procedure

Per lavorare con programmi in esecuzione su z/OS:

Eseguire il comando MODIFY dalla console z/OS. È possibile utilizzare l'abbreviazione *f*, come mostrato nell'esempio di sintassi seguente:

```
►► f—nomelavoro—, —| Parametri |—————►►
```

*f* *nomelavoro*, sostituisce il nome effettivo del comando: *asnccmd*, *asnacmd*, o *asnmcmd*. I parametri operativi che si applicano a ciascun comando possono essere utilizzati con la parola chiave *f*.

Ad esempio, per arrestare un programma Apply che utilizza il nome lavoro PLS, si utilizza il comando seguente:

```
F PLS,stop
```

Tabella 13 elenca i comandi Capture che è possibile eseguire con la parola chiave *f*. In tutti gli esempi, il nome lavoro è *mycap*.

Tabella 13. Comandi MODIFY di esempio per il programma Capture

Parametro	Comando di esempio che utilizza la parola chiave <i>f</i>
<b>prune</b>	f mycap,prune
<b>qryparms</b>	f mycap,qryparms
<b>reinit</b>	f mycap,reinit

Tabella 13. Comandi MODIFY di esempio per il programma Capture (Continua)

Parametro	Comando di esempio che utilizza la parola chiave f
<b>suspend</b>	f mycap,suspend
<b>resume</b>	f mycap,resume
<b>status</b>	f mycap,status
<b>stop</b>	f mycap,stop
<b>chgpargs</b>	f mycap,chgpargs autostop=y f mycap,chgpargs commit_interval=n f mycap,chgpargs logreuse=y f mycap,chgpargs logstdout=y f mycap,chgpargs memory_limit=n f mycap,chgpargs monitor_interval=n f mycap,chgpargs monitor_limit=n f mycap,chgpargs prune_interval=n f mycap,chgpargs retention_limit=n f mycap,chgpargs signal_limit=n f mycap,chgpargs sleep_interval=n f mycap,chgpargs term=y f mycap,chgpargs trace_limit=n

Tabella 14 elenca i comandi Apply che è possibile eseguire con la parola chiave f. In tutti gli esempi, il nome lavoro è myapp.

Tabella 14. Comandi MODIFY di esempio per il programma Apply

Parametro	Comando di esempio che utilizza la parola chiave f
<b>status</b>	f myapp,status
<b>stop</b>	f myapp,stop

Tabella 15 elenca i comandi del programma asntrc che è possibile eseguire con la parola chiave f. In tutti gli esempi, il nome lavoro è mycap.

Tabella 15. Comandi MODIFY di esempio per il programma asntrc

Attività	Comando di esempio che utilizza la parola chiave f
Avvia una traccia di programma con il comando asntrc	f mycap,asntrc on f mycap,asntrc statlong
Formatta un resoconto asntrc fmt e indirizza l'output verso un set di dati z/OS	F mycap, asntrc fmt -ofn // 'USRT001.TRCFMT'
Formatta un resoconto asntrc flw e indirizza l'output verso un set di dati z/OS	F mycap, asntrc flw -ofn // 'USRT001.TRCFLW'
Arresto di una traccia di programma	F mycap, asntrc off

**suggerimento:** Preallocare i file di output asntrc flw e fmt in modo che siano grandi abbastanza da contenere i resoconti asntrc. Usare gli attributi seguenti:

- **Nome set di dati:** USRT001.TRCFMT or USRT001.TRCFLW
- **Cilindri di allocazione primaria:** 2
- **Estensioni allocate normali:** 1
- **Classe di dati:** Nessuna (Utilizzo corrente)
- **Cilindri utilizzati:** 2

- **Formato di registrazione:** estensioni VB usate: 1
- **Lunghezza di registrazione:** 1028
- **Dimensione blocco:** 6144
- **Cilindri di prima estensione:** 2
- **Cilindri secondari:** 1
- **SMS comprimibile:** NO

Tabella 16 elenca i comandi Replication Alert che è possibile eseguire con la parola chiave f. In tutti gli esempi, il nome lavoro è mymon.

Tabella 16. Comandi MODIFY di esempio per il programma Replication Alert Monitor

Parametro	Comando di esempio che utilizza la parola chiave f
reinit	f mymon,reinit
status	f mymon,status
qryparms	f mymon,qryparms
suspend	f mymon,suspend
resume	f mymon,resume
stop	f mymon,stop
chgparms	f mymon,chgparms monitor_interval=n f mymon,chgparms autoprune=y f mymon,chgparms trace_limit=n f mymon,chgparms alert_prune_limit=n f mymon,chgparms max_notifications_per_alert=n f mymon,chgparms max_notifications_minutes=n

Per informazioni su MODIFY, consultare i comandi di sistema z/OS MVS .

## Avvio del programma Capture con JCL

z/OS

È possibile avviare il programma Capture su z/OS modificando ed eseguendo uno script di esempio preparato dalla directory di esempi.

### Procedure

Per avviare il programma Capture su z/OS con JCL:

1. Preparare JCL per z/OS.
  - a. Specificare i parametri di chiamata facoltativi appropriati nel campo PARM del lavoro Capture.
  - b. Se la variabile di ambiente TZ non è stata impostata nel file a livello di sistema /etc/profile o nel file .profile nella directory iniziale dell'utente che sta eseguendo il programma di replica, è necessario impostare le variabili di ambiente relative a fuso orario (TZ) e lunga nel JCL. Per ulteriori informazioni sull'impostazione della variabile TZ, consultare Replication installation and customization for z/OS.

L'esempio seguente di questa riga nel JCL di chiamata, comprende l'impostazione delle variabili TZ e LANG:

```
//CAPJFA EXEC PGM=ASNCAP, PARM='ENVAR('TZ=PST8PDT','LANG=en_US')/  
DSN6 cold capture_schema=JFA autostop'
```

2. Inviare il JCL da TSO o dalla console MVS.

---

## Utilizzare ARM (Automatic Restart Manager) per il riavvio automatico della replica e della pubblicazione (z/OS)

z/OS

È possibile utilizzare il sistema di recupero ARM (Automatic Restart Manager) su z/OS per il riavvio dei programmi Q Capture, Q Apply, Capture, Apply e Replication Alert Monitor.

### Prima di iniziare

Assicurarsi che ARM sia installato e che i programmi di replica siano stati configurati correttamente. Per utilizzare ARM con un programma di replica, assicurarsi che il programma dispone di autorizzazione APF. Ad esempio, per utilizzare ARM con Q Apply, Apply, o il programma Replication Alert Monitor, è necessario copiare il modulo di caricamento appropriato in una libreria che dispone di autorizzazione APF. (I programmi Q Capture e Capture devono disporre di autorizzazione APF a prescindere dal fatto che utilizzino ARM o meno.)

### Informazioni su questa attività

ARM è una funzione di recupero di z/OS che è in grado di migliorare la disponibilità di determinati lavori di batch o attività avviate. Quando un lavoro o un'attività non riesce, oppure ottiene un esito negativo sul sistema sul quale è in esecuzione, ARM è in grado di riavviare il lavoro senza alcun intervento da parte dell'operatore.

ARM utilizza nomi elemento per l'identificazione delle applicazioni con le quali funziona. Ciascuna applicazione con ARM abilitato genera infatti un nome elemento univoco che la identifica e che utilizzerà in tutte le comunicazioni con ARM. ARM rintraccia il nome elemento e dispone di una politica di riavvio definita per quanto riguarda i nomi elemento. Per maggiori dettagli sulla configurazione di ARM, consultare *z/OS MVS Sysplex Services Guide*.

### Procedure

Per utilizzare ARM per il riavvio automatico dei programmi di replica e pubblicazione:

1. Specificare uno dei seguenti nomi elemento quando viene eseguita la configurazione di ARM:

Programma	Nome elemento
Q Capture	ASNQCxxxxyyyy
Q Apply	ASNQAxxxxyyyy
Capture	ASNTC xxxxyyyy
Apply	ASNTA xxxxyyyy
Controllo degli avvisi di replica	ASNAM xxxxyyyy

In cui *xxxx* è il nome del sottosistema DB2 e *yyyy* è il nome del membro per la condivisione dei dati (quest'ultimo è necessario esclusivamente per configurazioni di condivisione dei dati). Il nome elemento è sempre di 16 caratteri, completato con spazi vuoti.

2. Opzionale: Se si dispone di più di un'istanza di un programma di replica o pubblicazione in esecuzione all'interno di un membro per la condivisione dei dati, specificare il parametro **arm** all'avvio dei programmi, per creare un nome elemento ARM univoco per ciascuna istanza del programma.

Il parametro **arm** assume un valore di tre caratteri che viene aggiunto ai nomi elemento elencati nella tabella precedente. La sintassi è **arm=zzz**, in cui *zzz* può essere una stringa alfanumerica di lunghezza qualsiasi. Ciononostante, il programma di replica incatenerà solo fino a tre caratteri al nome corrente, completandolo, se necessario, con spazi bianchi che permettano di ottenere un nome univoco di 16 byte.

I programmi di replica utilizzano il nome elemento per effettuare la registrazione con ARM durante l'inizializzazione. Non forniscono ad ARM alcun evento exit all'atto della registrazione. L'evento exit non è necessario poiché i programmi di replica non vengono eseguiti come sottosistema z/OS. ARM riavvia i programmi registrati nel caso in cui termineranno in modo anomalo (ad esempio, in caso di violazione di un segmento). La registrazione di programma di replica registrato viene annullata qualora questo venga terminato in modo anomalo (ad esempio, a causa di un comando STOP), oppure se questo rileva una registrazione non valida.

**Suggerimento:** Avviando i programmi Q Capture, Q Apply, Capture, Apply o Replication Alert Monitor con il parametro **term=n**, il programma non verrà arrestato quando DB2 è in stato di sospensione o arresto. In questo caso, il programma non annullerà la registrazione da ARM. Continuerà infatti ad essere in esecuzione, tuttavia non eseguirà il lavoro effettivo fino all'annullamento della sospensione o all'avvio di DB2.

---

## Migrazione dell'ambiente di replica in modalità di condivisione dei dati (z/OS)

z/OS

Se il programma Capture è in esecuzione in modalità di non condivisione dei dati, tuttavia l'utente effettua la migrazione dell'installazione in modalità di condivisione dei dati, è necessario preparare i sistemi per l'esecuzione in un Sysplex mediante l'esecuzione dell'utilità ASNPLXFY, da effettuarsi una volta.

### Prima di iniziare

Utilizzare lo stesso ID utente impiegato per l'esecuzione del programma Capture, oppure un ID che disponga degli stessi privilegi. Assicurarsi che l'utilità ASNPLXFY disponga di autorizzazione APF. È necessario eseguire il bind del piano ASNPLXFY al sottosistema. Inoltre, il sottosistema deve essere in esecuzione in modalità di condivisione dei dati. Per i dettagli sul bind di questa utilità, consultare la directory Programmi.

### Informazioni su questa attività

Eseguire questa utilità nella configurazione per la condivisione dei dati, prima dell'avvio a caldo del programma Capture, affinché quest'ultimo venga avviato nell'LRSN corretto. Questa utilità effettua la migrazione dei dati nella tabella

IBMSNAP\_RESTART. Converte, inoltre, i numeri della sequenza del log senza condivisione di dati (RBA) nei numeri di sequenza equivalenti (LRSN) in un ambiente di condivisione dei dati.

### **Procedure**

Per eseguire l'utilità ASNPLXFY nell'ambiente USS per la condivisione dei dati:

1. Arrestare il programma Capture.
2. Immettere il comando ASNPLXFY da una riga comandi. Ad esempio:

```
ASNPLXFY sottosistemapersonale schemaCapture
```

in cui è obbligatorio specificare il nome del sottosistema, mentre lo schema Capture è facoltativo. Lo schema Capture predefinito è ASN.

3. Avvio a caldo del programma Capture.

---

## Capitolo 12. Modifica di un ambiente di replica SQL

Di seguito vengono illustrate le procedure per apportare modifiche quotidiane ad un ambiente di replica Q.

---

### Registrazione di nuovi oggetti

In qualsiasi momento, nell'ambiente di replica in uso è possibile registrare una nuova tabella, vista o nickname. Non è necessario reinizializzare il programma Capture.


#### Informazioni su questa attività

Un nuovo oggetto registrato viene inizializzato automaticamente dal programma Capture la prima volta che il programma Apply elabora una serie di richieste che fa riferimento a tale oggetto. Il programma Apply segnala al programma Capture di iniziare la cattura delle modifiche per questo nuovo oggetto.

#### Procedure

Per registrare nuovi oggetti:

Utilizzare uno dei seguenti metodi per registrare nuovi oggetti:

Metodo	Descrizione
Programma della riga comandi ASNCLP	Utilizzare il comando CREATE REGISTRATION per registrare una tabella di origine, vista o nickname. Ad esempio, i seguenti comandi impostano l'ambiente e registrano la tabella DEPARTMENT nel database DB2 SAMPLE per la replica di aggiornamento completo.  SET SERVER CAPTURE TO DB SAMPLE; SET OUTPUT CAPTURE SCRIPT "registernew.sql"; SET LOG "registernew.err"; SET RUN SCRIPT LATER; CREATE REGISTRATION (DB2ADMIN.DEPARTMENT) FULL REFRESH ONLY;
Centro di replica	Utilizzare una delle seguenti finestre: <ul style="list-style-type: none"><li>• Notebook Proprietà della tabella registrata</li><li>• Notebook Proprietà della vista registrata</li><li>• Notebook Proprietà del nickname registrato</li></ul> Per visualizzare le finestre, fare clic sulla cartella <b>Tabelle registrate</b> , <b>Viste registrate</b> o <b>Nickname registrati</b> nella struttura ad albero degli oggetti su un server di controllo Capture, fare clic con il tasto destro del mouse sull'oggetto registrato nel pannello del contenuto e selezionare <b>Proprietà</b> .
 Comando di sistema ADDDPRREG	Utilizzare il comando di registrazione Add DPR (ADDDPRREG) per registrare una nuova tabella su System i.

---

## Modifica degli attributi di registrazione per gli oggetti registrati

È possibile modificare gli attributi di registrazione degli oggetti esistenti registrati in qualsiasi momento.

### Procedure

Per modificare gli attributi di registrazione degli oggetti registrati:

1. Modificare gli attributi utilizzando uno dei seguenti metodi.

Metodo	Descrizione
<b>Programma della riga comandi ASNCLP</b>	Utilizzare il comando ALTER REGISTRATION per modificare le proprietà di un oggetto registrato. Ad esempio, i seguenti comandi impostano l'ambiente e modificano la registrazione per la tabella STAFF nel database DB2 SAMPLE, in modo che gli aggiornamenti vengano catturati come coppie delete-insert:  <pre>SET SERVER CAPTURE TO DB SAMPLE; SET OUTPUT CAPTURE SCRIPT "register.sql"; SET LOG "register.err"; SET RUN SCRIPT LATER; ALTER REGISTRATION (DB2ADMIN.STAFF) UPDATE AS DELETE INSERT ON;</pre>
<b>Centro di replica</b>	Utilizzare una delle seguenti finestre: <ul style="list-style-type: none"><li>• Notebook Proprietà della tabella registrata</li><li>• Notebook Proprietà della vista registrata</li><li>• Notebook Proprietà del nickname registrato</li></ul> Per visualizzare le finestre, fare clic sulla cartella <b>Tabelle registrate</b> , <b>Viste registrate</b> o <b>Nickname registrati</b> nella struttura ad albero degli oggetti su un server di controllo Capture, fare clic con il tasto destro del mouse sull'oggetto registrato nel pannello del contenuto e selezionare <b>Proprietà</b> .

2. Dopo avere modificato gli attributi, reinizializzare il programma Capture.

---

## Aggiunta di colonne a tabelle di origine

Se è necessario aggiungere colonne ad un tabella di origine registrata, considerare per primo come la replica DB2 utilizza questa tabella. Se è necessario replicare le nuove colonne in questa tabella di origine, si deve assicurare che i programmi Capture e Apply riconoscano le nuove colonne e continuino l'elaborazione senza interruzione.

### Prima di iniziare

Prima di utilizzare questa procedura, prendere dimestichezza con le strutture dell'origine in uso, le tabelle CD (Change-Data) e di destinazione e con le registrazioni e le serie di richieste definite sul sistema in uso.

### Restrizioni

- **System i** La modifica di una tabella di origine in System i per l'aggiunta di nuove colonne non è supportata. System i crea una voce giornale EJ (end journaling) prima di eseguire la modifica della tabella di origine mediante l'operazione ALTER. Quando si provvede all'aggiunta di una nuova colonna in una tabella di origine, è necessario eliminare e ricreare la registrazione e la sottoscrizione per la tabella. Quando si aggiungono colonne a



una tabella System i che utilizza un RRN (Relative Record Number) come chiave primaria, rimuovere la registrazione, aggiungere la colonna alla tabella di origine, quindi riaggiungere tale tabella come una nuova registrazione. Specificare che verrà acquisito un RRN.

- Non è possibile utilizzare questa procedura per aggiungere colonne alle origini registrate su database relazionali non DB2. Una registrazione per un'origine relazionale non DB2 include una serie di trigger utilizzati per la cattura delle modifiche. Non è possibile modificare tali trigger. Quindi, se è necessario aggiungere nuove colonne a questa tabella di origine ed è necessario replicare i dati in queste colonne, si deve eliminare e ricreare l'origine registrata esistente.

### Informazioni su questa attività

Potrebbe essere necessario eseguire una determinata procedura a seconda se si desidera o meno replicare i dati nelle nuove colonne.

#### Non replicato

Se non si desidera replicare i dati nelle nuove colonne, non è necessario eseguire alcuna determinata procedura. Il programma Capture riconosce immediatamente le modifiche e continua l'esecuzione.

#### Replicato

Se si desidera replicare i dati in queste nuove colonne, attenersi a questa procedura per assicurare che i dati della nuova colonna vengano catturati e che i programmi Capture e Apply continuino l'esecuzione senza errori.

### Procedure

Per aggiungere colonne alle tabelle di origine:



1. Sospendere tutte le attività della tabella di origine che si desidera modificare.
2. Arrestare il programma Capture.
3. Opzionale: Se, durante questa procedura, è necessario mantenere attivo il programma Capture, inserire il segnale USER nella tabella IBMSNAP\_SIGNAL dopo aver arrestato l'attività della tabella di origine. Attendere che il programma Capture elabori il segnale USER. Una volta che il programma Capture elabora il segnale USER, il programma Capture non ha più attività da elaborare della tabella CD associata e non richiede più l'accesso a questa tabella CD.
4. Disattivare tutte le serie di richieste che richiedono questa tabella di origine dal Centro di replica.

**Nota:** Se non si desidera disattivare le serie di richieste durante questo processo, verificare che nessun programma Apply associato a tali serie sia in esecuzione nella tabella di origine quando si stanno aggiungendo le nuove colonne. In alternativa, accertarsi che i programmi Apply abbiano elaborato i dati fino all'LSN (Signal Log Sequence Number) associato al precedente segnale PRIOR.

I metodi in questo passo assicurano l'accesso esclusivo alla tabella CD in modo che è possibile modificare la tabella.

5. Inoltrare un'istruzione ALTER TABLE ADD in SQL per aggiungere le nuove colonne alla tabella di origine.
6. Aggiungere le nuove colonne alla tabella CD utilizzando il comando ALTER REGISTRATION nel programma della riga comandi ASNCLP o il notebook Proprietà tabella registrata nel Centro di replica. Il programma Capture

reinizializza automaticamente la registrazione e cattura le modifiche su queste nuove colonne quando il programma Capture legge per primo i dati di registrazione con le nuove colonne.

7. Inoltrare un'istruzione ALTER TABLE ADD in SQL per aggiungere le nuove colonne alla tabella di destinazione.
8. Disattivare le serie di richieste associate che non sono già state disattivate dal Centro di replica. Se assolutamente necessario, è possibile a questo punto riprendere l'attività su questa tabella di origine. Tuttavia, poiché le serie di richieste associate non sono ancora state modificate, è necessario tenerle disattivate in modo da non perdere eventuali modifiche eseguite su queste nuove colonne.
9. Aggiungere le nuove colonne ai membri della serie di richieste associati utilizzando il comando ALTER MEMBER ADD COLS nel programma della riga comandi ASNCLP o la finestra Aggiungi colonna alla tabella di destinazione nel Centro di replica.
10.   Se si esegue il programma Apply con **opt4one** impostato a *y*, arrestare e quindi riavviare il programma Apply.
11. Riattivare le serie di richieste.

---

## Arresto della cattura delle modifiche per gli oggetti registrati

È necessario disattivare un oggetto registrato prima di eliminarlo per accertarsi che i programmi Capture terminino eventuali elaborazioni necessarie dell'oggetto. Inoltre, è possibile disattivare un oggetto registrato se si desidera arrestare temporaneamente la cattura delle modifiche per tale oggetto, ma è necessario mantenere in esecuzione i programmi Capture per altri oggetti registrati.

### Restrizioni

È possibile disattivare solo oggetti registrati DB2 che sono definiti come origini programma Capture.

Non è possibile disattivare oggetti database relazionali non DB2 che sono usati da trigger Capture.

### Informazioni su questa attività

Il programma Capture arresta la cattura delle modifiche per gli oggetti origine che sono stati disattivati; tuttavia, le tabelle CD (Change-Data), gli attributi di registrazione e le serie di richieste associati a tali oggetti origine rimangono sul sistema.

Prima di disattivare un oggetto registrato, è necessario disattivare tutte le serie di richieste associate a tale oggetto registrato. Ciò garantisce che i programmi Apply in uso non interferiscano con il processo di disattivazione riattivando automaticamente l'oggetto prima che lo si elimini o prima che si sia pronti a riattivarlo.

Tutte le serie di richieste associate all'oggetto registrato sono implicate quando l'oggetto viene disattivato e quando la replica SQL interrompe la cattura delle modifiche per tale oggetto. Se si desidera continuare l'esecuzione di tali serie di richieste, è necessario eliminare i membri di serie di richieste che utilizzano tale oggetto registrato come origine dalle serie di richieste disattivate.

## Procedure

Per disattivare un oggetto registrato:

1. Disattivare tutte le serie di richieste associate utilizzando il Centro di replica. Fare clic sulla cartella **Serie di richieste**, fare clic con il tasto destro del mouse sulle serie di richieste attive nel pannello dei contenuti e selezionare **Disattiva**.
2. Disattivare l'oggetto registrato utilizzando uno dei seguenti metodi:

Metodo	Descrizione
Centro di replica	Fare clic sulla cartella <b>Tabelle registrate</b> , fare clic con il tasto destro del mouse sulla tabella registrata nel pannello dei contenuti e selezionare <b>Arresta cattura modifiche</b> .
SQL	Inserire manualmente un segnale CAPSTOP nella tabella IBMSNAP_SIGNAL.

---

## Esecuzione di registrazioni idonee per la riattivazione

Quando si riattiva una registrazione, il programma Capture riattiva la registrazione dopo che il programma Apply invia un segnale CAPSTART. Se, tuttavia, il programma Capture disattiva una registrazione a causa di un errore imprevisto, è necessario intraprendere una determinata azione per riattivarla.

### Prima di iniziare

Leggere i messaggi di errore generati dal programma Capture relativi a qualsiasi registrazione disattivata.

Prendere dimestichezza con la struttura delle tabelle di controllo Capture e con i programmi Capture in esecuzione sul sistema in uso.

### Informazioni su questa attività

Errori imprevisti possono far sì che il programma Capture imposti il valore della colonna STATE su S (Arrestato) nella tabella IBMSNAP\_REGISTER se il valore della colonna STOP\_ON\_ERROR per questa registrazione è impostato su N. Il valore di questa colonna STATE indica che il programma Capture ha interrotto l'elaborazione di questa registrazione e che la registrazione deve essere risolta. Il programma Apply non emette un segnale CAPSTART per alcuna registrazione che è in uno stato di arresto.

## Procedure

Per risolvere gli errori imprevisti e rendere idonee le registrazioni per la riattivazione:

1. Modificare la registrazione utilizzando le informazioni contenute nei messaggi di errori.
2. Dal server di controllo Capture, eseguire lo script SQL riportato di seguito per ripristinare la colonna STATE nella IBMSNAP\_REGISTER:

```
UPDATE schema.IBMSNAP_REGISTER
  SET STATE = 'I'
WHERE
  SOURCE_OWNER = 'SrcSchema' AND
  SOURCE_TABLE = 'SrcTbl' AND
  SOURCE_VIEW_QUAL = 'SrcVwQual' AND
  STATE = 'S';
```

dove *schema* è il nome dello schema Capture, *SrcSchema* è lo schema della tabella di origine registrata, *SrcTbl* è il nome della tabella di origine registrata e *SrcVwQual* è il qualificatore della vista di origine per questa tabella di origine. Una volta che la colonna STATE è impostata su I (Inattivo), il programma Capture è pronto ad iniziare la cattura dei dati non appena viene ricevuto un segnale CAPSTART, di solito dal programma Apply.

Si supponga che la tabella di origine per una registrazione attiva sia stata modificata inavvertitamente su DATA CAPTURE NONE (mentre dovrebbe essere DATA CAPTURE CHANGES). Inoltre, si supponga che questa registrazione sia stata definita con STOP\_ON\_ERROR = 'N', che specifica che il programma Capture non si arresterà quando rileva degli errori. Al successivo riavvio o reinizializzazione del programma Capture, tale programma riconoscerà questa condizione non corretta della tabella di origine e imposterà la colonna STATE su S (Arrestato) nella tabella IBMSNAP\_REGISTER per questa registrazione. Si riceverà un messaggio di errore quando il programma Apply tenta di elaborare la serie di richieste corrispondente, poiché la registrazione sarà in uno stato di arresto. È necessario:

- Correggere l'impostazione della tabella di origine mediante SQL inoltrando un'istruzione ALTER TABLE che ripristina l'opzione di tabella su DATA CAPTURE CHANGES.
- Ripristinare manualmente la registrazione da uno stato di arrestato ad uno stato inattivo, utilizzando lo script SQL precedente.

Il programma Apply eseguirà quindi un aggiornamento completo dell'intera serie di richieste.

---

## Rimozione delle registrazioni

Se si rimuove una registrazione, la replica SQL rimuove la registrazione dell'oggetto, elimina le tabelle CD (Change-Data) o CCD (Consistent-Change Data) associate ed elimina il nickname dell'oggetto CCD e qualsiasi trigger Capture per origini di database relazionali non DB2. La vista o tabella di origine corrente rimane nel database.

### Prima di iniziare

- Disattivare la registrazione per assicurare che il programma Capture finisca qualsiasi elaborazione attuale di questo oggetto.
- Disattivare le serie di richieste associate all'origine.

### Informazioni su questa attività

**Importante:** la disattivazione è un processo asincrono. Accertarsi che il processo di disattivazione finisca prima di rimuovere l'oggetto.

Se le modifiche vengono eseguite mentre il programma Capture è in esecuzione, tali modifiche non saranno riconosciute dal programma finché non viene reinizializzato o arrestato e riavviato.

### Procedure

Per rimuovere le registrazioni, utilizzare uno dei seguenti metodi:

Metodo	Descrizione
<b>Programma della riga comandi ASNCLP</b>	Utilizzare il comando DROP REGISTRATION per eliminare una o più registrazioni. Ad esempio, i seguenti comandi impostano l'ambiente ed eliminano la registrazione per la tabella DEPARTMENT nel database DB2 SAMPLE:  SET SERVER CAPTURE TO DB SAMPLE; SET OUTPUT CAPTURE SCRIPT "dropregis.sql"; SET LOG "dropregis.err"; SET RUN SCRIPT LATER; DROP REGISTRATION (DB2ADMIN.DEPARTMENT);
<b>Centro di replica</b>	Utilizzare le finestre Elimina tabelle registrate o Elimina viste registrate. Per visualizzare la finestra, fare clic sulla cartella <b>Tabelle registrate</b> o <b>Viste registrate</b> nella struttura ad albero dell'oggetto su un server di controllo Capture, fare clic con il tasto destro del mouse sull'oggetto registrato nel pannello del contenuto e selezionare <b>Elimina</b> .
<b>System i</b>  Comando di sistema RMVDPRREG	Utilizzare il comando RMVDPRREG (Remove DPR registration) per rimuovere una tabella di origine dalla tabella IBMSNAP_REGISTER.

## Modifica degli schemi Capture

È possibile modificare uno schema Capture esistente.

### Prima di iniziare

- Prendere dimestichezza con le tabelle di controllo della replica SQL e con le serie di richieste definite nel sistema in uso.
- Determinare il nome del nuovo schema Capture.
- Verificare che il server di controllo Capture in uso e tutti i server di controllo Apply associati a tale server siano stati migrati alla Versione 8 o successiva.

### Restrizioni

Non utilizzare questa procedura se il server di origine non è un database relazionale DB2.

### Informazioni su questa attività

**Suggerimento:** Se si impostano definizioni di controllo o programmi Replication Alert Monitor avviati nello schema Capture che si sta per modificare, eliminare queste definizioni di controllo. Una volta modificato lo schema Capture, ricreare le definizioni di controllo con il nome del nuovo schema Capture. Quindi, è possibile inizializzare nuovamente i controlli associati mediante il comando di sistema asnmcmd reinit. È possibile inoltre arrestare i controlli utilizzando il comando di sistema asnmcmd stop e quindi riavviare i programmi utilizzando il comando di sistema asnmon.

### Procedure

Per modificare gli schemi Capture:

1. Creare le tabelle di controllo per un nuovo schema Capture.

2. Arrestare il programma Capture.
3. Disattivare tutte le serie di richieste associate utilizzando il Centro di replica.
4. Dal server di controllo Apply, eseguire la seguente istruzione SQL per modificare i nomi di schema Capture per le serie di richieste associate alle tabelle di origine che appartengono a questo schema Capture:

```
UPDATE ASN.IBMSNAP_SUBS_SET
   SET CAPTURE_SCHEMA = 'NewSchema'
 WHERE
   CAPTURE_SCHEMA = 'ExistingSchema';
```

dove *NewSchema* è il nome del nuovo schema Capture e *ExistingSchema* è il nome dello schema Capture che si sta modificando.

5. Se sono state create serie di richieste con tabelle di destinazione (ad esempio, CCD o tabelle tipo replica) che sono registrate in questo schema Capture, eseguire la seguente istruzione SQL dal server di controllo Apply per modificare il nome dello schema di destinazione di tali serie di richieste:

```
UPDATE ASN.IBMSNAP_SUBS_SET
   SET TGT_CAPTURE_SCHEMA = 'NewSchema'
 WHERE
   TGT_CAPTURE_SCHEMA = 'ExistingSchema';
```

dove *NewSchema* è il nome del nuovo schema Capture e *ExistingSchema* è il nome dello schema Capture che si sta modificando.

6. Dal server di controllo Capture, eseguire un'istruzione SQL per copiare le informazioni attive da ciascuna tabella di controllo Capture esistente su ciascuna nuova tabella di controllo Capture corrispondente creata nel passo 1. Ad esempio, per copiare le informazioni attive nella tabella IBMSNAP\_REGISTER:

```
INSERT INTO NewSchema.IBMSNAP_REGISTER
   SELECT * FROM
     ExistingSchema.IBMSNAP_REGISTER;
```

dove *NewSchema* è il nome del nuovo schema Capture e *ExistingSchema* è il nome dello schema Capture che si sta modificando.

Ripetere questo passo per ciascuna tabella di controllo Capture esistente, comprese alcune o tutte le seguenti tabelle:

- IBMSNAP\_CAPMON
- IBMSNAP\_CAPPARMS
- IBMSNAP\_CAPTRACE
- IBMSNAP\_PRUNCNTL
- IBMSNAP\_PRUNE\_SET
- IBMSNAP\_REG\_EXT (solo System i)
- IBMSNAP\_REGISTER
- IBMSNAP\_RESTART
- IBMSNAP\_SIGNAL
- IBMSNAP\_UOW

Non è necessario ripetere questo passo per la tabella di controllo IBMSNAP\_CAPENQ (su UNIX, Windows, z/OS) o IBMSNAP\_PRUNE\_LOCK, poiché queste tabelle non contengono righe. *Non* modificare le tabelle CD.

7. Eliminare lo schema esistente e le tabelle di controllo Capture associate utilizzando il Centro di replica o ASNCLP.
8. Riavviare il programma Capture con il nome del nuovo schema.

9. Riattivare le serie di richieste associate utilizzando il Centro di replica.

---

## Creazione di nuove serie di richieste

È possibile creare nuove serie di richieste e aggiungere nuovi membri della serie di richieste nelle serie in qualsiasi momento per un oggetto registrato esistente.

### Prima di iniziare

Prima di creare una nuova serie di richieste, registrare le tabelle o le viste che si desidera utilizzare come origini.

### Restrizioni


Se il programma Apply corrispondente è attivo, non attivare la nuova serie di richieste finché non viene definita completamente la serie di richieste.

### Informazioni su questa attività

Questa procedura è relativa all'aggiunta di una nuova serie di richieste con o senza i membri della serie di richieste.

### Procedure

Per creare una nuova serie di richieste, utilizzare uno dei seguenti metodi:

Metodo	Descrizione
Programma della riga comandi ASNCLP	Utilizzare il comando CREATE SUBSCRIPTION SET per creare una serie vuota.
Centro di replica	Utilizzare il blocco note <b>Crea serie di richieste</b> per creare una serie e aggiungere un membro o per creare una serie vuota.  Per visualizzare il blocco note, espandere il server di controllo Apply su cui verrà definita la serie, fare clic con il tasto destro del mouse sulla cartella <b>Serie di richieste</b> e fare clic su <b>Crea</b> .
 Comando di sistema ADDDPRSUB	Utilizzare il comando per l'aggiunta della serie di richieste DPR (ADDDPRSUB) per creare una serie di richieste con un membro o nessun membro.

---

## Aggiunta di nuovi membri di serie di richieste alle serie di richieste esistenti

È possibile aggiungere uno o più membri che utilizzano ognuno la stessa tabella di origine su una o più serie di richieste esistenti. Ad esempio, se si selezionano tre serie di richieste, è possibile aggiungere un membro ad ognuna di queste serie, le quali utilizzano tutte la stessa origine di replica.

### Informazioni su questa attività

Quando si aggiunge un membro ad una serie di richieste, si stanno inserendo informazioni sui nuovi membri nelle tabelle di controllo Apply. Nella maggior parte dei casi, il programma Apply legge queste informazioni all'inizio del successivo ciclo Apply.

Tuttavia, se si aggiunge un membro ad una serie di richieste elaborata con l'opzione OPT4ONE su Linux, UNIX, Windows o z/OS o con l'opzione OPTSNGSET su System i, è necessario arrestare il programma Apply per la serie di richieste e quindi riavviarlo. Se si elabora una serie con l'opzione OPT4ONE, il programma Apply legge in memoria le informazioni della tabella di controllo per la serie in modo che non è necessario passare alle tabelle di controllo per leggere le informazioni per la serie all'inizio di ogni ciclo Apply.

Se la tabella di origine per il membro è registrata per la replica differenziale e il programma Capture è già in esecuzione, non è necessario arrestare o reiniziare il programma Capture prima di aggiungere il membro. Poiché il membro aggiunto deve usare una tabella registrata come relativa origine, il programma Capture catturerà le modifiche per esso.

### Procedure

Per aggiungere nuovi membri di serie di richieste alle serie di richieste esistenti, utilizzare uno dei seguenti metodi:

Metodo	Descrizione
<b>Programma della riga comandi ASNCLP</b>	Utilizzare il comando CREATE MEMBER per aggiungere un membro della serie di richieste ad una serie esistente.
<b>Centro di replica</b>	Utilizzare il notebook <b>Aggiungi membro a serie di richieste</b> .  Per visualizzare il notebook, fare clic sulla cartella <b>Tabelle registrate</b> . Nel pannello di contenuti, fare clic con il tasto destro del mouse su una tabella registrata che si desidera utilizzare e fare clic su <b>Aggiungi membro</b> .
<b>System i</b> Comando di sistema ADDDPRSUBM	Utilizzare il comando ADDDPRSUBM (Aggiunta membro della serie di richieste DPR) per aggiungere un membro ad una serie di richieste esistente.

## Disabilitazione dei membri della serie di richieste dalle serie di richieste esistenti

Se si desidera che il programma Apply ignori un membro della serie di richieste che presenta un errore e continui ad elaborare la parte restante della serie di richieste, è necessario disabilitare il membro della serie di richieste che presenta l'errore.

### Informazioni su questa attività

In caso di errori durante la replica in una tabella della serie di richieste, il programma Apply inserisce un messaggio di errore nella tabella IBMSNAP\_APPLYTRAIL e continua ad elaborare gli altri membri nel ciclo di Apply.

### Procedure

Per disabilitare un membro della serie di richieste, eseguire la seguente istruzione UPDATE di SQL:

```
UPDATE ASN.IBMSNAP_SUBS_MEMBR
SET MEMBER_STATE = 'D'
WHERE APPLY_QUAL= apply_qualifier
```



```

        SET_NAME = set_name
        WHOS_ON_FIRST = whos_on_first
        SOURCE_OWNER = source_owner
    SOURCE_TABLE = source_table
    SOURCE_VIEW_QUAL = source_view_qualifier
    TARGET_OWNER = target_owner
    TARGET_TABLE = target_table

```

Il programma Apply non elaborerà questo membro finché quest'ultimo non verrà riabilitato.

---

## Abilitazione dei membri della serie di richieste nelle serie di richieste esistenti

È possibile aggiungere o riabilitare i membri disabilitati in una serie di richieste impostando MEMBER\_STATE su N (nuovo).

### Procedure

Per riabilitare un membro della serie di richieste, eseguire la seguente istruzione UPDATE di SQL:

```

UPDATE ASN.IBMSNAP_SUBS_MEMBR
SET MEMBER_STATE = 'N'
WHERE APPLY_QUAL= apply_qualifier
      SET_NAME = set_name
      WHOS_ON_FIRST = whos_on_first
      SOURCE_OWNER = source_owner
      SOURCE_TABLE = source_table
      SOURCE_VIEW_QUAL = source_view_qualifier
      TARGET_OWNER = target_owner
      TARGET_TABLE = target_table

```

---

## Modifica delle proprietà delle serie di richieste

È possibile modificare le proprietà di una serie di richieste, mentre Apply continua ad eseguire e ad elaborare altre serie, e riattivare la serie prima del successivo ciclo di Apply.

### Informazioni su questa attività

Nel seguente elenco vengono descritti gli attributi che potrebbe essere necessario modificare:

- Pianificazioni per l'applicazione degli aggiornamenti (replica in base all'ora o sugli eventi)
- Istruzioni delle richieste
- Predicati della clausola WHERE dei membri della serie di richieste
- Conteggio commit
- Valore di blocco dei dati (MAX\_SYNCH\_MINUTES)

Disattivando prima la serie di richieste, si impedisce al programma Apply di elaborare la serie mentre vengono apportate le modifiche. Il programma Apply riconosce le modifiche della serie di richieste durante il successivo ciclo di Apply, dopo avere riattivato la serie.

### Procedure

Per modificare le proprietà di una serie di richieste:

1. Disattivare la serie di richieste mediante il Centro di replica.
2. Utilizzare uno dei seguenti metodi per modificare la serie di richieste:

Metodo	Descrizione
<b>Programma della riga comandi ASNCLP</b>	Utilizzare il comando ALTER SUBSCRIPTION SET.  I seguenti comandi impostano l'ambiente e modificano la serie di richieste SET00 per ridurre l'intervallo di tempo a 15 minuti:  SET SERVER CAPTURE TO DB SAMPLE; SET SERVER CONTROL TO DB TARGET; SET OUTPUT CAPTURE SCRIPT "capsubsetchg.sql" CONTROLSCRIPT "appsubsetchg.sql"; SET LOG "subsetchg.err"; SET RUN SCRIPT LATER; ALTER SUBSCRIPTION SET SETNAME SET00 APPLYQUAL A000 SETTYPE R ACTIVATE YES TIMING INTERVAL 15 COMMIT COUNT NULL;
<b>Centro di replica</b>	Utilizzare il blocco note Proprietà della serie di richieste. Per visualizzare il blocco note, fare clic sulla cartella <b>Serie di richieste</b> su un server di controllo Apply, fare clic con il tasto destro del mouse sulla serie di richieste nel pannello del contenuto e fare clic su <b>Proprietà</b> .

3. Riattivare la serie di richieste.

**z/OS** **Linux UNIX Windows** Se si imposta il parametro **opt4one** del programma Apply su y, arrestare e riavviare il programma Apply, altrimenti le modifiche non verranno riconosciute.

## Modifica dei nomi delle serie di richieste

È possibile modificare il nome di una serie di richieste senza dovere eliminare e ricreare la serie di richieste e tutti i suoi membri.

### Prima di iniziare

Prima di eseguire le istruzioni SQL, è necessario acquisire familiarità con la struttura delle tabelle di controllo delle repliche SQL e con le serie di richieste definite sul sistema.

**Suggerimento:** Se sono state configurate le definizioni di controllo o sono stati avviati i programmi Replication Alert Monitor per rilevare le condizioni di segnalazione per la serie di richieste, eliminare queste definizioni. Dopo avere modificato il nome della serie di richieste, ricreare le definizioni di controllo mediante il Centro di replica o ASNCLP. Quindi, è possibile inizializzare nuovamente i controlli mediante il comando del sistema asnmcmd reinit. È anche possibile arrestare i controlli mediante il comando asnmcmd stop e riavviare i programmi mediante il comando asnmon.

### Procedure

Per modificare il nome di una serie di richieste:

1. Utilizzare il Centro di replica per disattivare la serie di richieste.
2. Dal server di controllo Apply, eseguire le seguenti istruzioni SQL per modificare il nome della serie di richieste nelle tabelle IBMSNAP\_SUBS\_SET, IBMSNAP\_SUBS\_MEMBR e IBMSNAP\_SUBS\_COLS:

```

UPDATE ASN.IBMSNAP_SUBS_SET
  SET SET_NAME      = 'NewSetName'
WHERE
  APPLY_QUAL       = 'ApplyQual'      AND
  SET_NAME         = 'ExistSetName'   AND
  WHOS_ON_FIRST   = 'Val';

UPDATE ASN.IBMSNAP_SUBS_MEMBR
  SET SET_NAME      = 'NewSetName'
WHERE
  APPLY_QUAL       = 'ApplyQual'      AND
  SET_NAME         = 'ExistSetName'   AND
  WHOS_ON_FIRST   = 'Val';

UPDATE ASN.IBMSNAP_SUBS_COLS
  SET SET_NAME      = 'NewSetName'
WHERE
  APPLY_QUAL       = 'ApplyQual'      AND
  SET_NAME         = 'ExistSetName'   AND
  WHOS_ON_FIRST   = 'Val';

```

dove *NewSetName* è il nome della nuova serie di richieste, *ApplyQual* è il qualificatore Apply, *ExistSetName* è il nome esistente della serie di richieste e *Val* è F o S.

3. Se questa serie di richieste utilizza le istruzioni SQL before o after oppure le chiamate di procedura, eseguire il seguente script SQL dal server di controllo Apply per modificare il nome della serie di richieste nella tabella IBMSNAP\_SUBS\_STMTS:

```

UPDATE ASN.IBMSNAP_SUBS_STMTS
  SET SET_NAME      = 'NewSetName'
WHERE
  APPLY_QUAL       = 'ApplyQual'      AND
  SET_NAME         = 'ExistSetName'   AND
  WHOS_ON_FIRST   = 'Val';

```

dove *NewSetName* è il nome della nuova serie di richieste, *ApplyQual* è il qualificatore Apply, *ExistSetName* è il nome esistente della serie di richieste e *Val* è F o S.

4. Da server di controllo Capture, eseguire le seguenti istruzioni SQL per modificare il nome della serie di richieste nelle tabelle IBMSNAP\_PRUNE\_SET e IBMSNAP\_PRUNCNTL:

```

UPDATE Schema.IBMSNAP_PRUNE_SET
  SET SET_NAME      = 'NewSetName'
WHERE
  APPLY_QUAL       = 'ApplyQual'      AND
  SET_NAME         = 'ExistSetName'   AND
  TARGET_SERVER    = 'Target_Server';

UPDATE Schema.IBMSNAP_PRUNCNTL
  SET SET_NAME      = 'NewSetName'
WHERE
  APPLY_QUAL       = 'ApplyQual'      AND
  SET_NAME         = 'ExistSetName'   AND
  TARGET_SERVER    = 'Target_Server';

```

dove *Schema* è il nome dello schema Capture, *NewSetName* è il nuovo nome della serie di richieste, *ApplyQual* è il qualificatore Apply, *ExistSetName* è il nome esistente della serie di richieste e *Target\_Server* è il percorso del database delle tabelle di destinazione.

5. Se si esegue il programma Apply in Linux, UNIX, Windows o z/OS, con **opt4one** impostato su *y*, arrestare e riavviare il programma Apply.
6. Riattivare la serie di richieste dal Centro di replica.

---

## Suddivisione di una serie di richieste

È possibile suddividere una serie di richieste in due o più serie senza rimuovere e ricreare le informazioni relative alla serie di richieste.

### Prima di iniziare

- Prima di eseguire le istruzioni SQL, è necessario acquisire familiarità con la struttura delle tabelle di controllo delle repliche SQL e con le serie di richieste definite sul sistema.
- Identificare i membri della serie di richieste che si desidera suddividere e determinare le tabelle di origine e di destinazione associate a questi membri della serie di richieste.
- Identificare il server di controllo Capture, il server di destinazione e il server di controllo Apply della serie di richieste che si desidera suddividere. È necessario utilizzare questi percorsi del server di controllo Capture, del server di destinazione e del server di controllo Apply per la nuova serie di richieste che si desidera creare utilizzando questa procedura.

### Informazioni su questa attività

**Suggerimento:** Se sono state configurate le definizioni di controllo o sono stati avviati i programmi Replication Alert Monitor per rilevare le condizioni di segnalazione per la serie di richieste, eliminare queste definizioni. Dopo avere suddiviso la serie di richieste, ricreare le definizioni di controllo mediante il Centro di replica o ASNCLP. Quindi, è possibile inizializzare nuovamente i controlli mediante il comando del sistema `asnmcmd reinit`. È anche possibile arrestare i controlli mediante il comando `asnmcmd stop` e riavviare i programmi mediante il comando `asnmon`.

### Procedure

Per suddividere una serie di richieste:

1. Disattivare la serie di richieste che si desidera suddividere dal Centro di replica. Dalla cartella Serie di richieste, fare clic con il tasto destro del mouse sulla serie di richieste attiva nel pannello del contenuto e selezionare Disattiva.
2. Creare una nuova serie di richieste. La nuova serie è rappresentata da una nuova riga nella tabella `IBMSNAP_SUBS_SET`. Lasciare questa nuova serie di richieste inattiva.
3. Dal server di controllo Apply, eseguire la seguente istruzione SQL per copiare le informazioni dalla serie di richieste esistente nella riga della nuova serie di richieste nella tabella `IBMSNAP_SUBS_SET`:

```
UPDATE ASN.IBMSNAP_SUBS_SET
   SET STATUS      =
   (SELECT STATUS FROM ASN.IBMSNAP_SUBS_SET B
    WHERE APPLY_QUAL = 'ApplyQual' AND
          SET_NAME   = 'ExistName'   AND
          WHOS_ON_FIRST = 'Val'),
   LASTRUN        =
   (SELECT LASTRUN FROM ASN.IBMSNAP_SUBS_SET B
    WHERE APPLY_QUAL = 'ApplyQual' AND
          SET_NAME   = 'ExistName'   AND
          WHOS_ON_FIRST = 'Val'),
   SYNCHPOINT     =
   (SELECT SYNCHPOINT FROM ASN.IBMSNAP_SUBS_SET B
    WHERE APPLY_QUAL = 'ApplyQual' AND
          SET_NAME   = 'ExistName'   AND
```

```

        WHOS_ON_FIRST = 'Val'),
SYNCHTIME =
    (SELECT SYNCHTIME FROM ASN.IBMSNAP_SUBS_SET B
     WHERE APPLY_QUAL = 'ApplyQual' AND
           SET_NAME    = 'ExistName'    AND
           WHOS_ON_FIRST = 'Val'),
LASTSUCCESS =
    (SELECT LASTSUCCESS FROM ASN.IBMSNAP_SUBS_SET B
     WHERE APPLY_QUAL = 'ApplyQual' AND
           SET_NAME    = 'ExistName'    AND
           WHOS_ON_FIRST = 'Val')
WHERE
    APPLY_QUAL    = 'ApplyQual'    AND
    SET_NAME      = 'NewName'      AND
    WHOS_ON_FIRST = 'Val';

```

dove *ApplyQual* è il qualificatore Apply, *ExistName* è il nome della serie di richieste esistente che si sta suddividendo, *Val* è F o S e *NewName* è il nome della nuova serie di richieste che si sta creando.

4. Dal server di controllo Capture, eseguire la seguente istruzione SQL per inserire una nuova riga per la nuova serie di richieste nella tabella IBMSNAP\_PRUNE\_SET:

```

INSERT INTO Schema.IBMSNAP_PRUNE_SET
    (APPLY_QUALIFIER,
     SET_NAME,
     TARGET_SERVER,
     SYNCHTIME,
     SYNCHPOINT
VALUES ('ApplyQual',
       'NewName',
       'Target_Server',
       NULL,
       x'00000000000000000000');

```

dove *Schema* è il nome dello schema Capture, *ApplyQual* è il qualificatore Apply, *NewName* è il nome della nuova serie di richieste che si sta creando e *Target\_Server* è il percorso del database delle tabelle di destinazione.

5. Dal server di controllo Capture, eseguire la seguente istruzione SQL per copiare le informazioni dalla serie di richieste esistente nella riga della nuova serie di richieste nella tabella IBMSNAP\_PRUNE\_SET:

```

UPDATE Schema.IBMSNAP_PRUNE_SET
    SET SYNCHPOINT
        (SELECT SYNCHPOINT FROM Schema.IBMSNAP_PRUNE_SET B
         WHERE APPLY_QUAL = 'ApplyQual' AND
               SET_NAME    = 'ExistName'    AND
               TARGET_SERVER = 'Target_Server'),
    SYNCHTIME =
        (SELECT SYNCHTIME FROM Schema.IBMSNAP_PRUNE_SET B
         WHERE APPLY_QUAL = 'ApplyQual' AND
               SET_NAME    = 'ExistName'    AND
               TARGET_SERVER = 'Target_Server')
WHERE
    APPLY_QUAL    = 'ApplyQual'    AND
    SET_NAME      = 'NewName'      AND
    TARGET_SERVER = 'Target_Server';

```

dove *Schema* è il nome dello schema Capture, *ApplyQual* è il qualificatore Apply, *ExistName* è il nome della serie di richieste esistente che si sta suddividendo, *Target\_Server* è il percorso del database delle tabelle di destinazione e *NewName* è il nome della nuova serie di richieste che si sta creando.

6. Dal server di controllo Apply, eseguire le seguenti istruzioni SQL per modificare il nome della serie di richieste nelle tabelle IBMSNAP\_SUBS\_MEMBR e IBMSNAP\_SUBS\_COLS per ciascun membro della serie di richieste che si sta spostando nella nuova serie di richieste:

```

UPDATE ASN.IBMSNAP_SUBS_MEMBR
  SET SET_NAME      = 'NewName'
  WHERE
    APPLY_QUAL      = 'ApplyQual' AND
    SET_NAME        = 'ExistName' AND
    WHOS_ON_FIRST   = 'Val'        AND
    SOURCE_OWNER    = 'SrcSchema'  AND
    SOURCE_TABLE     = 'SrcTbl'     AND
    SOURCE_VIEW_QUAL = 'SrcVwQual'  AND
    TARGET_OWNER    = 'TgtSchema'  AND
    TARGET_TABLE    = 'TgtTbl';

UPDATE ASN.IBMSNAP_SUBS_COLS
  SET SET_NAME      = 'NewName'
  WHERE
    APPLY_QUAL      = 'ApplyQual'   AND
    SET_NAME        = 'ExistName'   AND
    WHOS_ON_FIRST   = 'Val'         AND
    TARGET_OWNER    = 'TgtSchema'   AND
    TARGET_TABLE    = 'TgtTbl';

```

dove *NewName* è il nome della nuova serie di richieste che si sta creando, *ApplyQual* è il qualificatore Apply, *ExistName* è il nome della serie di richieste esistente che si sta suddividendo, *Val* è F o S, *SrcSchema* è lo schema della tabella di origine, *SrcTbl* è il nome della tabella di origine, *SrcVwQual* è il qualificatore della vista di origine per questa tabella di origine, *TgtSchema* è lo schema della tabella di destinazione e *TgtTbl* è il nome della tabella di destinazione.

Ripetere questa operazione per ciascun membro della serie di richieste che si desidera spostare nella nuova serie di richieste.

7. Se la serie di richieste che si sta suddividendo utilizza le istruzioni SQL before o after o le chiamate di procedura, spostare le istruzioni applicabili nella nuova serie di richieste nella tabella IBMSNAP\_SUBS\_STMTS:
- a. Eseguire il seguente script SQL dal server di controllo Apply per spostare le istruzioni:

```

UPDATE ASN.IBMSNAP_SUBS_STMTS
  SET SET_NAME      = 'NewName'
  WHERE
    APPLY_QUAL      = 'ApplyQual'   AND
    SET_NAME        = 'ExistName'   AND
    WHOS_ON_FIRST   = 'Val'         AND
    STMT_NUMBER     in (Stmt1, Stmt2, .. Stmtn);

```

dove *NewName* è il nome della nuova serie di richieste che si sta creando, *ApplyQual* è il qualificatore Apply, *ExistName* è il nome della serie di richieste esistente che si sta suddividendo, *Val* è F o S e *Stmt1*, *Stmt2* e *Stmtn* corrispondono ai numeri delle istruzioni che l'utente sta spostando nella nuova serie di richieste.

- b. Regolare i valori della colonna AUX\_STMTS nella tabella IBMSNAP\_SUBS\_SET in base al nuovo numero di istruzioni per entrambe le serie di richieste. Rinumerare le istruzioni per eliminare eventuali duplicati, se necessario.
8. Dal server di controllo Capture, eseguire la seguente istruzione SQL per modificare il nome della serie di richieste nella tabella IBMSNAP\_PRUNCNTL per ciascun membro della serie di richieste spostato:



```

UPDATE Schema.IBMSNAP_PRUNCNTL
SET SET_NAME      = 'NewName'
WHERE
  APPLY_QUAL      = 'ApplyQual'    AND
  SET_NAME        = 'ExistName'    AND
  TARGET_SERVER   = 'Target_Server' AND
  SOURCE_OWNER    = 'SrcSchema'    AND
  SOURCE_TABLE    = 'SrcTbl'       AND
  SOURCE_VIEW_QUAL = SrcVwQual     AND
  TARGET_OWNER    = 'TgtSchema'    AND
  TARGET_TABLE    = 'TgtTbl';

```

dove *Schema* è il nome dello schema Capture, *NewName* è il nome della nuova serie di richieste creata nel passo 2, *ApplyQual* è il qualificatore Apply, *ExistName* è il nome della serie di richieste esistente che è stata suddivisa, *Target\_Server* è il percorso del database delle tabelle di destinazione, *SrcSchema* è lo schema della tabella di origine, *SrcTbl* è il nome della tabella di origine, *SrcVwQual* è il qualificatore della vista di origine per questa tabella di origine della replica, *TgtSchema* è lo schema della tabella di destinazione e *TgtTbl* è il nome della tabella di destinazione.

Ripetere questa operazione per ciascun membro della serie di richieste spostato nella nuova serie di richieste.

9.   Se si esegue il programma Apply con **opt4one** impostato a y, arrestare e quindi riavviare il programma Apply.
10. Riattivare entrambe le serie di richieste dal Centro di replica.

---

## Unione delle serie di richieste

È possibile unire due serie di richieste in una sola serie. È utile unire le serie di richieste se si desidera che le tabelle di destinazione in queste due serie di richieste presentino la stessa coerenza della transazione, ma non si desidera eliminare e ricreare le informazioni relative alla serie di richieste.

### Prima di iniziare

Prima di eseguire le istruzioni SQL, è necessario acquisire familiarità con la struttura delle tabelle di controllo delle repliche SQL e con le serie di richieste definite sul sistema.

Identificare il server di controllo Capture, il server di destinazione e il server di controllo Apply di ciascuna serie di richieste che si desidera unire. Verificare che tutte le serie di richieste che si desidera unire siano state create con lo stesso server di controllo Capture, lo stesso server di destinazione e lo stesso server di controllo Apply.

### Restrizioni

Le due serie di richieste che si desidera unire devono derivare i dati di origine dallo stesso server Capture e mediante lo stesso schema Capture.

**Importante:** È necessario che le due serie di sottoscrizioni abbiano elaborato i dati di origine fino allo stesso valore di punto di sincronizzazione, per evitare la perdita di dati durante l'unione delle serie di sottoscrizioni.

### Procedure

Per unire le serie di richieste:

1. Arrestare il programma Capture associato. Attendere che entrambe le serie di sottoscrizioni raggiungano lo stesso punto di sincronizzazione e synchtime indicato nella tabella IBMSNAP\_SUBS\_SET.

**Suggerimento:** Se non si desidera arrestare il programma Capture, inserire un segnale USER nella tabella IBMSNAP\_SIGNAL e generare un evento con END\_SYNCHPOINT (nella tabella IBMSNAP\_SUBS\_EVENT) impostato sul valore della colonna SIGNAL\_LSN nella tabella IBMSNAP\_SIGNAL, in modo da applicare solo i dati fino a quel punto.

2. Disattivare entrambe le serie di richieste dal Centro di replica.
3. Dal server di controllo Apply, eseguire la seguente istruzione SQL per eliminare la riga dalla tabella IBMSNAP\_SUBS\_SET corrispondente alla serie di richieste che si sta spostando nell'altra serie di richieste:

```
DELETE FROM ASN.IBMSNAP_SUBS_SET
WHERE
    APPLY_QUAL      = 'ApplyQual' AND
    SET_NAME        = 'Subset_To_Move' AND
    WHOS_ON_FIRST  = 'Val';
```

dove *ApplyQual* è il qualificatore Apply, *Subset\_To\_Move* è il nome della serie di richieste che si sta spostando in un'altra serie di richieste esistente e *Val* è F o S.

4. Dal server di controllo Capture, eseguire la seguente istruzione SQL per eliminare la riga dalla tabella IBMSNAP\_PRUNE\_SET corrispondente alla serie di richieste che si sta spostando nell'altra serie di richieste:

```
DELETE FROM Schema.IBMSNAP_PRUNE_SET
WHERE
    APPLY_QUAL      = 'ApplyQual' AND
    SET_NAME        = 'Subset_To_Move' AND
    TARGET_SERVER   = 'Target_Server' ;
```

dove *Schema* è il nome dello schema Capture, *ApplyQual* è il qualificatore Apply, *Subset\_To\_Move* è il nome della serie di richieste che si sta spostando in un'altra serie di richieste esistente e *Target\_Server* è il percorso del database delle tabelle di destinazione.

5. Dal server di controllo Apply, eseguire le seguenti istruzioni SQL per impostare il nome della serie di richieste che si sta spostando sul nome dell'altra serie di richieste nelle tabelle IBMSNAP\_SUBS\_MEMBR e IBMSNAP\_SUBS\_COLS:

```
UPDATE ASN.IBMSNAP_SUBS_MEMBR
    SET SET_NAME      = 'Existing_Merged_Subset'
WHERE
    APPLY_QUAL      = 'ApplyQual' AND
    SET_NAME        = 'Subset_To_Move' AND
    WHOS_ON_FIRST  = 'Val';

UPDATE ASN.IBMSNAP_SUBS_COLS
    SET SET_NAME      = 'Existing_Merged_Subset'
WHERE
    APPLY_QUAL      = 'ApplyQual' AND
    SET_NAME        = 'Subset_To_Move' AND
    WHOS_ON_FIRST  = 'Val';
```

dove *Existing\_Merged\_Subset* è il nome della serie di richieste esistente che viene unita con la serie di richieste che si sta spostando, *ApplyQual* è il qualificatore Apply, *Subset\_To\_Move* è il nome della serie di richieste che si sta spostando nella serie di richieste esistente e *Val* è F o S.

6. Se la serie di richieste che si sta spostando utilizza le istruzioni SQL before o after oppure le chiamate di procedura, modificare il nome della serie di richieste nella tabella IBMSNAP\_SUBS\_STMTS:



- a. Eseguire il seguente script SQL dal server di controllo Apply per modificare il nome della serie di richieste:



```
UPDATE ASN.IBMSNAP_SUBS_STMTS
  SET SET_NAME      = 'Existing_Merged_Subset'
  WHERE
    APPLY_QUAL      = 'ApplyQual' AND
    SET_NAME        = 'Subset_To_Move' AND
    WHOS_ON_FIRST  = 'Val';
```

dove *Existing\_Merged\_Subset* è il nome della serie di richieste esistente che viene unita alla serie di richieste che si sta spostando, *ApplyQual* è il qualificatore Apply, *Subset\_To\_Move* è il nome della serie di richieste che si sta spostando nella serie di richieste esistente e *Val* è F o S.

- b. Regolare il valore della colonna AUX\_STMTS nella tabella IBMSNAP\_SUBS\_SET per riflettere il nuovo conteggio di istruzioni nella serie di richieste unita esistente. Rinumerare le istruzioni per eliminare eventuali duplicati, se necessario.
7. Dal server di controllo Capture, eseguire la seguente istruzione SQL per impostare il nome della serie di richieste che è stata spostata sul nome della serie di richieste unita nella tabella IBMSNAP\_PRUNCNTL:

```
UPDATE Schema.IBMSNAP_PRUNCNTL
  SET SET_NAME      = 'Existing_Merged_Subset'
  WHERE
    APPLY_QUAL      = 'ApplyQual' AND
    SET_NAME        = 'Subset_To_Move' AND
    TARGET_SERVER   = 'Target_Server' ;
```

dove *Schema* è il nome dello schema Capture, *Existing\_Merged\_Subset* è il nome della serie di richieste esistente che viene unita con la serie di richieste che si sta spostando, *ApplyQual* è il qualificatore Apply, *Subset\_To\_Move* è il nome della serie di richieste che si sta spostando in un'altra serie di richieste esistente e *Target\_Server* è il percorso del database delle tabelle di destinazione.

8.   Se si esegue il programma Apply con **opt4one** impostato a y, arrestare e quindi riavviare il programma Apply.
9. Riattivare la serie di richieste unita dal Centro di replica.

---

## Modifica dei qualificatori Apply delle serie di richieste

Per modificare il qualificatore Apply di una serie di richieste, è possibile utilizzare SQL per apportare la modifica senza eliminare e ricreare la serie di richieste.

### Prima di iniziare

Prima di eseguire le istruzioni SQL, è necessario acquisire familiarità con la struttura delle tabelle di controllo delle repliche SQL e con le serie di richieste definite sul sistema.

Inoltre, è necessario determinare le seguenti informazioni:

- Il nome del nuovo qualificatore Apply.
- Le serie di richieste che si desidera spostare dal qualificatore Apply esistente nel nuovo qualificatore Apply.
- Eventuali istruzioni SQL before o after oppure le chiamate di procedura definite per queste serie di richieste.

### Informazioni su questa attività

Se diverse serie di richieste utilizzano lo stesso qualificatore Apply, è possibile spostare alcune delle serie di richieste in un nuovo qualificatore Apply per bilanciare i carichi di lavoro dei programmi Apply.

**Suggerimento:** Se sono state configurate le definizioni di controllo o sono stati avviati i programmi Replication Alert Monitor per rilevare le condizioni di segnalazione per il qualificatore Apply, eliminare queste definizioni. Dopo avere modificato il qualificatore, ricreare le definizioni di controllo mediante il Centro di replica o ASNCLP. Quindi, è possibile inizializzare nuovamente i controlli mediante il comando del sistema `asnmcmd reinit`. È anche possibile arrestare i controlli mediante il comando `asnmcmd stop` e riavviare i programmi mediante il comando `asnmon`.

È necessario eseguire le istruzioni SQL in questa procedura per ciascuna serie di richieste che si desidera spostare.

### Procedure

Per modificare i qualificatori Apply delle serie di richieste:

1. Disattivare le serie di richieste da modificare mediante il Centro di replica.
2. Dal server di controllo Apply, eseguire le seguenti istruzioni SQL per modificare il qualificatore Apply della serie di richieste nelle tabelle `IBMSNAP_SUBS_SET`, `IBMSNAP_SUBS_MEMBR` e `IBMSNAP_SUBS_COLS`:

```
UPDATE ASN.IBMSNAP_SUBS_SET
  SET APPLY_QUAL = 'NewApplyQual'
WHERE
  APPLY_QUAL = 'ExistApplyQual' AND
  SET_NAME = 'Name' AND
  WHOS_ON_FIRST = 'Val';

UPDATE ASN.IBMSNAP_SUBS_MEMBR
  SET APPLY_QUAL = 'NewApplyQual'
WHERE
  APPLY_QUAL = 'ExistApplyQual' AND
  SET_NAME = 'Name' AND
  WHOS_ON_FIRST = 'Val';

UPDATE ASN.IBMSNAP_SUBS_COLS
  SET APPLY_QUAL = 'NewApplyQual'
WHERE
  APPLY_QUAL = 'ExistApplyQual' AND
  SET_NAME = 'Name' AND
  WHOS_ON_FIRST = 'Val';
```

dove *NewApplyQual* è il nuovo qualificatore Apply, *ExistApplyQual* è il qualificatore Apply esistente, *Name* è il nome della serie di richieste e *Val* è F o S.

3. Se questa serie di richieste utilizza le istruzioni SQL `before` o `after` oppure le chiamate di procedura, eseguire le seguenti istruzioni SQL sul server di controllo Apply per modificare il qualificatore Apply della serie di richieste nella tabella `IBMSNAP_SUBS_STMTS`:

```
UPDATE ASN.IBMSNAP_SUBS_STMTS
  SET APPLY_QUAL = 'NewApplyQual'
WHERE
  APPLY_QUAL = 'ExistApplyQual' AND
  SET_NAME = 'Name' AND
  WHOS_ON_FIRST = 'Val';
```

dove *NewApplyQual* è il nuovo qualificatore Apply, *ExistApplyQual* è il qualificatore Apply esistente, *Name* è il nome della serie di richieste e *Val* è F o S.

4. Dal server di controllo Capture, eseguire le seguenti istruzioni SQL per modificare il qualificatore Apply della serie di richieste nelle tabelle IBMSNAP\_PRUNE\_SET e IBMSNAP\_PRUNCNTL:

```
UPDATE Schema.IBMSNAP_PRUNE_SET
SET APPLY_QUAL = 'NewApplyQual'
WHERE
  APPLY_QUAL = 'ExistApplyQual' AND
  SET_NAME = 'Name' AND
  TARGET_SERVER = 'Target_Server';

UPDATE Schema.IBMSNAP_PRUNCNTL
SET APPLY_QUAL = 'NewApplyQual'
WHERE
  APPLY_QUAL = 'ExistApplyQual' AND
  SET_NAME = 'Name' AND
  TARGET_SERVER = 'Target_Server';
```

dove *Schema* è il nome dello schema Capture, *NewApplyQual* è il nuovo qualificatore Apply, *ExistApplyQual* è il qualificatore Apply esistente, *Name* è il nome della serie di richieste e *Target\_Server* è il percorso del database delle tabelle di destinazione.

5. Ripetere le operazioni da 2 a 4 per ciascuna serie di richieste rimanente che si desidera spostare.
6. Se il programma Apply è in esecuzione con l'opzione **opt4one** impostata su *y* in Linux, UNIX, Windows o z/OS, arrestare e riavviare il programma Apply.
7. Riattivare le serie di richieste mediante il Centro di replica.

---

## Disattivazione delle serie di richieste

È possibile disattivare una serie di richieste senza rimuoverla. Quando si disattiva una serie di richieste, il programma Apply completa il suo ciclo di elaborazione corrente e sospende le operazioni per quella serie di richieste.

### Prima di iniziare

Prima di eseguire le istruzioni SQL, è necessario acquisire familiarità con la struttura delle tabelle di controllo delle repliche SQL e con le serie di richieste definite sul sistema.

### Informazioni su questa attività

Può essere necessario eseguire una manutenzione speciale su queste serie di richieste disattivate, a seconda dell'intervallo di tempo in cui rimangono disattivate:

#### Periodo di tempo breve

Non vi sono requisiti di elaborazione speciali per le serie di richieste disattivate temporaneamente. Generalmente, le serie di richieste vengono disattivate temporaneamente durante la modifica dei relativi attributi o durante la correzione di errori nelle tabelle di destinazione.

Utilizzare il Centro di replica per disattivare, modificare e riattivare una serie di richieste.

#### Periodo di tempo lungo

È possibile disattivare una serie di richieste che al momento non è

necessaria, ma che potrebbe essere utilizzata in futuro. Tuttavia, è necessario eseguire ulteriori operazioni se questa serie di richieste deve rimanere disattivata per un lungo periodo di tempo, durante il quale i dati modificati si accumulano e vengono rallentate le prestazioni dei programmi Capture ed Apply.

Il programma Capture utilizza le informazioni provenienti dai programmi Apply attivi durante il processo di eliminazione. Se i programmi Apply non sono attivi o le serie di richieste vengono disattivate per lunghi periodi di tempo, le informazioni di eliminazione diventano obsolete e le tabelle UOW (unit-of-work) e CD (change-data) non potranno essere eliminate rapidamente e in modo efficace, se le registrazioni attive associate alle serie di richieste disattivate non vengono eliminate. Queste informazioni obsolete possono rallentare notevolmente le prestazioni dei rimanenti programmi Apply attivi e causare un utilizzo non necessario e costoso della CPU da parte del processo di eliminazione. Le tabelle UOW e CD vengono eliminate in base al limite di conservazione (il valore predefinito è sette giorni) del programma Capture. Tuttavia, è possibile che grandi quantità di dati si accumulino in questo periodo, a seconda della dimensione dell'ambiente della replica.

Per impedire tali problemi di eliminazione, è possibile utilizzare SQL per reimpostare le informazioni di eliminazione per una serie di richieste che deve rimanere disattivata per un lungo periodo di tempo.

Se sono state disattivate tutte le serie di richieste associate ad un oggetto registrato, è necessario disattivare anche l'oggetto registrato per impedire al programma Capture di catturare i dati non necessari.

### Procedure

1. Dal Centro di replica, disattivare la serie. Fare clic sulla cartella **Serie di richieste**, fare clic con il tasto destro del mouse sulla serie di richieste attiva nel pannello del contenuto e selezionare **Disattiva**.
2. Dal server di controllo Capture, eseguire le seguenti istruzioni SQL per reimpostare le informazioni di eliminazione nelle tabelle IBMSNAP\_PRUNE\_SET e IBMSNAP\_PRUNCNTL per la serie di richieste disattivata:

```
UPDATE Schema.IBMSNAP_PRUNE_SET
  SET SYNCHPOINT = x'00000000000000000000' AND
      SYNCHTIME   = NULL
WHERE
  APPLY_QUAL      = 'ApplyQual' AND
  SET_NAME        = 'Name'        AND
  TARGET_SERVER   = 'Target_Server';

UPDATE Schema.IBMSNAP_PRUNCNTL
  SET SYNCHPOINT = NULL AND
      SYNCHTIME   = NULL
WHERE
  APPLY_QUAL      = 'ApplyQual' AND
  SET_NAME        = 'Name'        AND
  TARGET_SERVER   = 'Target_Server';
```

dove *Schema* è il nome dello schema Capture, *ApplyQual* è il qualificatore Apply, *Name* è il nome della serie di richieste e *Target\_Server* è il percorso del database delle tabelle di destinazione.

---


## Rimozione delle serie di richieste

Se non è più necessario replicare i dati in una determinata serie di richieste, è possibile rimuovere la serie di richieste. Tuttavia, se il programma Apply sta elaborando la serie di richieste che viene rimossa, il lavoro del programma Apply viene interrotto ed eventuali altre serie di richieste in quel lavoro non vengono elaborate finché non viene riavviato il lavoro.

### Procedure

Per rimuovere le serie di richieste:

1. Per garantire che il programma Apply abbia completato l'elaborazione corrente per la serie di richieste, disattivare la serie di richieste prima di rimuoverla, dal Centro di replica Fare clic sulla cartella **Serie di richieste**, fare clic con il tasto destro del mouse sulla serie di richieste attiva nel pannello del contenuto e selezionare **Disattiva**.
2. Utilizzare uno dei seguenti metodi per rimuovere una serie di richieste disattivata:

Metodo	Descrizione
Programma della riga comandi ASNCLP	Utilizzare il comando DROP SUBSCRIPTION SET.  I seguenti comandi impostano l'ambiente e rilasciano una serie di richieste denominata SET00 con un qualificatore Apply AQ00. SET SERVER CAPTURE TO DB SAMPLE; SET SERVER CONTROL TO DB TARGET; SET OUTPUT CAPTURE SCRIPT "drpcapsubset.sql" CONTROLSCRIPT "drpappsubset.sql"; SET LOG "drpsubset.err"; SET RUN SCRIPT LATER; DROP SUBSCRIPTION SET SETNAME SET00 APPLYQUAL AQ00;
Centro di replica	Utilizzare la finestra Elimina serie di richieste. Per visualizzare la finestra, fare clic sulla cartella <b>Serie di richieste</b> , fare clic con il tasto destro del mouse sulla serie di richieste attiva nel pannello del contenuto e selezionare <b>Eliminazione</b> .
 Comando di sistema RMDVPRSUB	Utilizzare il comando RMDVPRSUB (per la rimozione della serie di richieste DPR) per rimuovere una serie di richieste.

Il programma Capture continua a catturare i dati e a scrivere le righe nella tabella CD (change-data), anche se si rimuovono tutte le serie di richieste per l'oggetto registrato. Per impedire questa continua elaborazione da parte del programma Capture, disattivare o rimuovere l'oggetto registrato dopo avere rimosso le relative serie di richieste.

---

## Coordinamento degli eventi della replica con gli eventi delle applicazioni del database

È possibile coordinare gli eventi della replica e del database inserendo manualmente le righe nella tabella IBMSNAP\_SIGNAL. Queste righe, note come segnali, indicano ai programmi Capture in esecuzione di eseguire determinate azioni.

## Impostazione di un evento END\_SYNCHPOINT utilizzando il segnale tipo USER

È possibile impostare il valore della colonna SIGNAL\_TYPE su USER per stabilire un determinato punto sulla registrazione di recupero DB2 e per coordinare un evento replica con un evento applicazione database.

### Informazioni su questa attività

Ad esempio, se si sta eseguendo la replica di dati OLTP (Online Transaction Processing) su una memoria di dati gestita separatamente, si potrebbe voler mantenere i dati della memoria piuttosto stabili per un'elaborazione di interrogazioni appropriata al contesto. Si aggiornano quindi i dati della memoria solo con le modifiche verificatesi in un determinato momento nell'applicazione OLTP. In tale caso, l'evento applicazione database è la conclusione logica della giornata di lavoro. L'evento replica sarà l'applicazione delle modifiche dalla chiusura di una determinata giornata di lavoro a quella seguente. Si supponga che le serie di richieste siano configurate solo per l'elaborazione di eventi.

### Procedure

Per creare un segnale tipo USER:

1. Creare un segnale tipo USER Capture inserendo la seguente riga nella tabella IBMSNAP\_SIGNAL:

```
INSERT INTO Schema.IBMSNAP_SIGNAL
      (signal_type,
       signal_subtype,
       signal_state)
VALUES ('USER',
       'USER APPLY EVENT SIGNAL',
       'P');
```

Eeguire questa istruzione SQL INSERT quando si verifica l'evento applicazione database (in tale caso alla fine della giornata di lavoro dell'applicazione).

Il programma Capture agisce su questo record di registrazione di tabella di segnali dopo che il programma Capture individua questo record sulla registrazione di recupero del database e solo quando il programma Capture rileva il record di commit corrispondente per questo inserimento, verificando che è stato eseguito il commit di questo evento.

Quando viene eseguito il commit di un segnale tipo USER, il programma Capture aggiorna i seguenti valori della colonna IBMSNAP\_SIGNAL che corrispondono al record di registrazione di inserimento da elaborare:

- SIGNAL\_STATE = 'R' (ricevuto dal programma Capture)
  - SIGNAL\_LSN = numero di sequenza di registrazione dal record di registrazione di commit per l'unità di lavoro DB2 che contiene questo inserimento di riga di segnale
2. Utilizzare il valore attualmente nella colonna SIGNAL\_LSN della riga del segnale inserito per inserire un valore END\_SYNCHPOINT nella tabella di controllo IBMSNAP\_SUBS\_EVENT. Tale nuovo valore segnala al programma Apply che il programma Capture ha raccolto tutti i dati relativi alla nuova giornata di lavoro e che il programma Apply deve estrarre ed applicare i dati solo fino al valore della colonna SIGNAL\_LSN.

È possibile automatizzare l'inserimento nella tabella IBMSNAP\_SUBS\_EVENT creando un trigger di aggiornamento sulla tabella IBMSNAP\_SIGNAL:

```

CREATE TRIGGER EVENT_TRIG
NO CASCADE AFTER UPDATE ON Schema.IBMSNAP_SIGNAL
REFERENCING NEW AS N
FOR EACH ROW MODE DB2SQL
WHEN (N.SIGNAL_SUBTYPE = 'USER APPLY EVENT SIGNAL')
INSERT INTO ASN.IBMSNAP_SUBS_EVENT VALUES
('WH_APPLY_EVENT',
(CURRENT_TIMESTAMP + 2 MINUTES),
N.SIGNAL_LSN,
null);

```

Tale trigger si manifesta ogni volta che la tabella IBMSNAP\_SIGNAL viene aggiornata dal programma Capture. Quando una colonna SIGNAL\_SUBTYPE viene aggiornata su 'USER APPLY EVENT SIGNAL', il trigger inserisce una riga nella tabella IBMSNAP\_SUBS\_EVENT. Tale riga indica al programma Apply che deve estrarre ed applicare il lavoro dall'ultima giornata (di cui è stato eseguito il commit prima sul valore SIGNAL\_LSN come calcolato dal programma Capture) dopo che sono trascorsi due minuti.

## Uso del segnale CMD STOP di Capture

È possibile impostare il valore della colonna SIGNAL\_TYPE su CMD e il valore della colonna SIGNAL\_SUBTYPE su STOP per arrestare un processo del programma Capture in un determinato punto sulla registrazione di recupero DB2.

È possibile utilizzare il segnale CMD STOP di Capture nelle seguenti situazioni:

- Per coordinare il programma Capture con qualsiasi modifica della tabella di origine che presenta precedenti record di registrazione non leggibili. Ciò potrebbe verificarsi se è stata eliminata e quindi ricreata una tabella o se è stata riorganizzata una tabella senza impostare l'opzione KEEPDICTIONARY su YES.
- Per coordinare un punto di recupero comune tra sistemi di database distribuiti replicati.

## Coordinamento di una modifica della tabella di origine con il programma Capture

È possibile utilizzare un segnale sottotipo STOP del tipo CMD di Capture per chiudere un programma Capture e coordinare le modifiche della tabella di origine.

### Procedure

Per coordinare modifiche della tabella di origine:

1. Creare un segnale sottotipo STOP del tipo CMD di Capture inserendo una riga nella tabella IBMSNAP\_SIGNAL utilizzando la seguente istruzione SQL:

```

INSERT INTO Schema.IBMSNAP_SIGNAL
(signal_type,
signal_subtype,
signal_state)
VALUES('CMD',
'STOP',
'P');

```

Inserire questa riga quando si verifica l'evento dell'applicazione di database, dopo che l'attività della tabella di origine è stata posta in stato di sospensione, ma prima che l'attività provochi modifiche del record di registrazione problematiche.

Il programma Capture agisce su questo record di registrazione di tabella di segnali dopo che il programma Capture individua questo record sulla registrazione di recupero del database e solo quando il programma Capture

rileva il record di commit corrispondente per questo inserimento, verificando che è stato eseguito il commit di questo evento.

Il programma Capture chiude in modo ordinato tutti i thread di Capture dopo aver eseguito il commit di tutti i dati catturati dalle transazioni sulla registrazione che sono precedenti al record di registrazione del commit per l'unità di lavoro DB2 che contiene questa riga IBMSNAP\_SIGNAL inserita. Prima di terminare, il programma Capture aggiorna inoltre i seguenti valori nella riga della tabella IBMSNAP\_SIGNAL corrispondente al record di registrazione di inserimento da elaborare:

- SIGNAL\_STATE = 'R' (ricevuto dal programma Capture)
- SIGNAL\_LSN = numero di sequenza di registrazione dal record di registrazione di commit per l'unità di lavoro DB2 che contiene questo inserimento di riga di segnale

Tutti i record di registrazione per la tabella di origine di modifica vengono elaborati dal programma Capture quando questo termina.

2. A seconda dello scenario in uso, eliminare e ricreare la tabella di origine oppure riorganizzare e comprimere la tabella di origine senza impostare l'opzione KEEPDICTIONARY su YES.
3. Se sono state eliminate o modificate colonne replicate, è necessario ora modificare le serie di richieste e le registrazioni corrispondenti create per questa tabella di origine. Tali modifiche, se necessario, possono essere coordinate ulteriormente con il programma Apply attendendo le serie di richieste interessate per rilevare il programma Capture attualmente arrestato. Una serie di richieste è sincrona rispetto al programma Capture quando il valore della colonna SYNCHPOINT nella tabella IBMSNAP\_SUBS\_SET è uguale al valore della colonna MAX\_COMMITSEQ nella tabella *Schema*.IBMSNAP\_RESTART.

## Impostazione di un punto di recupero distribuito

È possibile utilizzare un segnale sottotipo STOP del tipo CMD di Capture per impostare i database origine e destinazione in uso su punti di recupero equivalenti e recuperare i database in un punto di coerenza comune.

### Prima di iniziare

Prima di utilizzare questa procedura, verificare che nel database di destinazione siano state create le tabelle di controllo Apply in uso.

Inoltre, verificare che tutta l'attività sul database origine sia stata posta in sospensione prima dell'inserimento della riga nella tabella IBMSNAP\_SIGNAL. Tuttavia, non creare la copia immagine o di backup delle tabelle di database finché non si inserisce la riga nella tabella IBMSNAP\_SIGNAL.

Se le serie di richieste in uso non sono configurate tipicamente per l'elaborazione dell'evento, è necessario impostarle provvisoriamente sul tempo basato sull'evento. Utilizzare la seguente istruzione SQL per inserire una riga nella tabella IBMSNAP\_SUBS\_EVENT degli eventi di sottoscrizione:

```
INSERT INTO ASN.IBMSNAP_SUBS_EVENT
VALUES('RECOVERY_EVENT',
      CURRENT_TIMESTAMP + 2 MINUTES,
      SIGNAL_LSN_value,
      NULL);
```

dove *SIGNAL\_LSN\_value* è il numero di sequenza di registrazione impostata dal programma Capture e memorizzata nella tabella IBMSNAP\_SIGNAL.



## Procedure

Per impostare un punto di recupero distribuito:

1. Creare un segnale sottotipo STOP del tipo CMD di Capture inserendo una riga nella tabella IBMSNAP\_SIGNAL utilizzando la seguente istruzione SQL:

```
INSERT INTO Schema.IBMSNAP_SIGNAL
      (signal_type,
       signal_subtype,
       signal_state)
VALUES ('CMD',
       'STOP',
       'P');
```

Il programma Capture agisce su questo record di registrazione di tabella di segnali dopo che il programma Capture individua questo record sulla registrazione di recupero del database e solo quando il programma Capture rileva il record di commit corrispondente per questo inserimento, verificando che è stato eseguito il commit di questo evento.

Il programma Capture chiude in modo ordinato tutti i thread di Capture dopo aver eseguito il commit di tutti i dati catturati dalle transazioni sulla registrazione che sono precedenti al record di registrazione del commit per l'unità di lavoro DB2 che contiene questa riga IBMSNAP\_SIGNAL inserita. Prima di terminare, il programma Capture aggiorna inoltre i seguenti valori nella riga della tabella IBMSNAP\_SIGNAL corrispondente al record di registrazione di inserimento da elaborare:

- SIGNAL\_STATE = 'R' (ricevuto dal programma Capture)
- SIGNAL\_LSN = numero di sequenza di registrazione dal record di registrazione di commit per l'unità di lavoro DB2 che contiene questo inserimento di riga di segnale

Tutti i record di registrazione per il database origine vengono elaborati dal programma Capture quando questo termina.

2. Eseguire i programmi di utilità copia immagine o backup database origine.
3. Utilizzare il valore nella colonna SIGNAL\_LSN dalla riga di tabella IBMSNAP\_SIGNAL inserita come valore END\_SYNCHPOINT nella tabella IBMSNAP\_SUBS\_EVENT. Tale valore segnala al programma Apply che il programma Capture ha raccolto tutti i dati di cui è stato eseguito il commit prima del punto di backup e che il programma Apply deve estrarre ed applicare i dati solo fino al valore della colonna SIGNAL\_LSN. Le serie di richieste elaborano tutti i dati fino al valore SIGNAL\_LSN.
4. Eseguire i programmi di utilità copia immagine o backup database destinazione. I database origine e destinazione hanno ora punti di recupero equivalenti ed è possibile recuperare entrambi i database in un punto di coerenza comune.

È possibile riprendere tutta l'attività del database origine non appena gli eventi Apply non vengono impostati e l'attività del programma di utilità copia immagine o backup database origine è completa. È possibile inoltre avviare il programma Capture. Una volta che l'attività del programma di utilità copia immagine o backup database destinazione è completa, è possibile riportare le opzioni di pianificazione delle serie di richieste in uso alle impostazioni di origine (in base all'ora, basate sull'evento o entrambe).

**System i** È possibile inviare il segnale STOP per arrestare un singolo lavoro giornale o arrestarli tutti. Per arrestarne uno, inserire il segnale nella tabella dei segnali designata per tale giornale (la tabella IBMSNAP\_SIGNAL\_XXXX\_YYYY,

dove *xxxx* è la libreria giornale e *yyyy* è il nome giornale. Per arrestare tutti i lavori giornale, inserire il segnale nella tabella *schema.IBMSNAP\_SIGNAL*. Per arrestare un singolo lavoro giornale in una configurazione di giornale remota, inserire il segnale nella tabella dei segnali di giornale sul server origine. Fare riferimento alla descrizione su come creare tabelle dei segnali di giornale in una configurazione di giornale remota.

## Esecuzione di un segnale handshake CAPSTART esternamente al programma Apply

Prima che il programma Apply possa utilizzare qualsiasi serie di richieste per estrarre ed applicare modifiche dalle tabelle CD, deve verificarsi un "handshake" (comunicazione sincronizzata) tra i programmi Capture e Apply di ciascun membro in tale serie di richieste.

### Informazioni su questa attività

Il programma Apply inizia l'handshake inserendo un segnale sottotipo CAPSTART di tipo CMD nella tabella *IBMSNAP\_SIGNAL*. Il programma Apply inserisce tale segnale prima di eseguire un aggiornamento completo di qualsiasi membro della serie di richieste con una tabella di destinazione definita come completa.

### Procedure

Per eseguire un segnale handshake CAPSTART esternamente al programma Apply:

Creare un segnale sottotipo CAPSTART del tipo CMD di Capture inserendo una riga nella tabella *IBMSNAP\_SIGNAL* utilizzando la seguente istruzione SQL:

```
INSERT INTO Schema.IBMSNAP_SIGNAL
    (signal_type,
     signal_subtype,
     signal_input_in,
     signal_state)
VALUES('CMD',
      'CAPSTART',
      mapid,
      'P');
```

dove *mapid* è il valore della colonna *MAP\_ID* della tabella *Schema.IBMSNAP\_PRUNCNTL* e corrisponde alla riga per il membro della serie di richieste che richiede l'handshake.

**Nota:** Eseguire questa istruzione SQL INSERT prima di eseguire un aggiornamento completo del membro della serie di richieste, se necessario.

Il programma Capture agisce su questo record di registrazione di tabella di segnali dopo che il programma Capture individua questo record sulla registrazione di recupero del database e solo quando il programma Capture rileva il record di commit corrispondente per questo inserimento, verificando che è stato eseguito il commit di questo evento.

Il programma Capture verifica se la registrazione associata è già inserita nella memoria in base all'uso precedente della tabella registrata. Se la tabella registrata non è in uso, il programma Capture legge le informazioni di registrazione associate nella memoria e imposta i valori nella tabella *IBMSNAP\_REGISTER* per mostrare che questa tabella registrata è ora attiva e in uso.

Indipendentemente se la tabella registrata è in uso o meno, il programma Capture imposta i valori delle colonne *SYNCHPOINT* e *SYNCHTIME* nella riga associata nella tabella *Schema.IBMSNAP\_PRUNCNTL* sul numero di sequenza di

registrazione dal record di registrazione di commit per l'unità di lavoro DB2 che contiene questa riga di segnale inserita e sulla data/ora dallo stesso record di registrazione di commit, rispettivamente.

Il programma Capture aggiorna i seguenti valori nella riga della tabella IBMSNAP\_SIGNAL corrispondente al record di registrazione di inserimento da elaborare:

- SIGNAL\_STATE = 'C' (ricevuto e completato dal programma Capture)
- SIGNAL\_LSN = numero di sequenza di registrazione dal record di registrazione di commit per l'unità di lavoro DB2 che contiene questo inserimento di riga di segnale

## Esecuzione di un segnale CAPSTOP

È possibile avviare un segnale CAPSTOP se si desidera arrestare manualmente la cattura delle modifiche per una registrazione. È possibile utilizzare questo segnale quando si disattiva una registrazione o prima che se ne elimina una.

### Procedure

Per eseguire un segnale CAPSTOP:

1. Creare un segnale sottotipo CAPSTOP del tipo CMD di Capture inserendo una riga nella tabella IBMSNAP\_SIGNAL utilizzando la seguente istruzione SQL:

```
INSERT INTO Schema.IBMSNAP_SIGNAL
        (signal_type,
         signal_subtype,
         signal_input_in,
         signal_state)
VALUES('CMD',
       'CAPSTOP',
       source_owner.source_table,
       'P');
```

dove *Schema* è il nome dello schema Capture e *source\_owner.source\_table* è il nome completo della tabella che non richiede più modifiche catturate.

Il programma Capture agisce su questo record di registrazione di tabella di segnali dopo che il programma Capture individua questo record sulla registrazione di recupero del database e solo quando il programma Capture rileva il record di commit corrispondente per questo inserimento, verificando che è stato eseguito il commit di questo evento.

Il programma Capture verifica se la registrazione associata è già inserita nella memoria in base all'uso precedente della tabella registrata. Se la tabella registrata non è attualmente in uso, il programma Capture ignora il segnale CAPSTOP.

Se la tabella registrata è in uso, il programma Capture elimina quanto contenuto nella memoria associata a tale registrazione e disattiva la registrazione (impostando la colonna STATE nella tabella IBMSNAP\_REGISTER su 'I'). Il programma Capture arresta quindi la cattura delle modifiche per questa tabella registrata.

Il programma Capture aggiorna i seguenti valori di colonna nella riga della tabella IBMSNAP\_SIGNAL corrispondente al record di registrazione di inserimento da elaborare:

- SIGNAL\_STATE = 'C' (ricevuto e completato dal programma Capture)
- SIGNAL\_LSN = numero di sequenza di registrazione dal record di registrazione di commit per l'unità di lavoro DB2 che contiene questo inserimento di riga di segnale

2. Opzionale: Facoltativo: eliminare la registrazione.
3. **System i** Opzionale: È possibile anche inviare un segnale CAPSTOP per arrestare l'acquisizione di modifiche per una registrazione inserendo il segnale nella tabella IBMSNAP\_SIGNAL\_XXXX\_yyyy, dove XXXX è la libreria giornale e yyyy è il nome del giornale. Per arrestare la cattura delle modifiche per una registrazione in una configurazione di giornale remota, inserire il segnale CAPSTOP sul server origine.

---

## Adjusting for Daylight Savings Time (System i)

### System i

On System i, the Capture program uses a timestamp and the journal sequence number when reading changes from a journal. This process can create problems when it is necessary to adjust the system clock for U.S. Daylight Savings Time in the autumn and spring.

#### Informazioni su questa attività

System i systems provide two methods for adjusting to Daylight Savings Time:

**V5R3** The system either slows down its clock (autumn) or speeds up (spring) to avoid skipping or duplicating any timestamps. If you are running the Capture program on System i V5R3 and use this new method to make the time change, you do not need to use the procedure below.

#### Before V5R3

You must stop all activity on the system for one hour and then move the clock back one hour in autumn. With this method, you need to use the procedure below.

#### Procedure

To do adjust for Daylight Savings Time:

1. Follow these steps when you must turn back the clock by one hour in autumn:
  - a. Stop the Capture program and any applications that update the source tables.
  - b. Wait for the system time to move forward by at least an hour without any new journal entries to the source journal.
  - c. Set the system time back by an hour.
  - d. Restart the Capture program.

The following example demonstrates the use of this procedure:

- a. At 12:00 you stop the Capture program and all applications.
- b. You wait until 13:00 so that the journal entry timestamps only have values up to 12:00.
- c. You set the system time back to 12:00.
- d. You make a change. The journal entry timestamp for the change will be 12:01.
- e. You restart Capture. Capture will start from 12:00 and therefore will capture the change that came at 12:01 (Daylight Savings Time), which is 13:01 in Standard time.

The Capture program restarts with a timestamp that is lower than the current system time. No journal entries will be added until the new system time is greater than the system time just before the time change, so there is no possibility of missing any data.

**Recommendation:** Although the time change has no effect on the Apply program, stop and restart the Apply program during the time change also.

2. Follow this procedure when you must move the clock forward by one hour in spring:
  - a. Stop the Capture program and make the time change. The Capture program responds as though an hour passed with no changes to the source tables.

---

## Opzioni per la promozione della configurazione della replica su un altro sistema

Quando si definiscono oggetti registrati o serie di sottoscrizioni su un sistema (ad esempio, un sistema di prova), ed è necessario copiare l'ambiente della replica su un altro sistema (ad esempio, un sistema di produzione), è possibile utilizzare le funzioni di promozione del Centro di replica.

Le funzioni di promozione decodificano gli oggetti registrati o le serie di richieste per creare file di script con appropriati DDL (Data Definition Language) e DML (Data Manipulation Language). È possibile copiare le definizioni di replica su un altro database senza dover registrare di nuovo le origini o ricreare le serie di richieste.

Ad esempio, utilizzare le funzioni di promozione per definire serie di richieste per database di destinazione remoti. Una volta definito un sistema di destinazione modello nell'ambiente di prova, è possibile creare script di serie di richieste (e modificare il qualificatore Apply utilizzato e così via) per i sistemi di destinazione remoti in uso, che non sono altrimenti supportati da un punto di controllo centrale.

**Importante:** Le funzioni di promozione non collegano al sistema di destinazione e non convalidano i parametri di configurazione della replica di tale sistema.

Di seguito vengono descritte le tre opzioni per la promozione della configurazione della replica su un altro sistema.

### Promuovi tabelle registrate

Questa funzione promuove le informazioni di registrazione per le tabelle specificate. Facoltativamente, questa funzione promuove definizioni di tablespace, indice e tabella base. È possibile specificare un diverso schema Capture e un diverso nome server per le tabelle che si promuovono. Inoltre, è possibile modificare il nome dello schema per le tabelle CD (Change-Data) associate alle tabelle di origine promosse.

È possibile promuovere più tabelle registrate contemporaneamente. I nuovi nomi di schema forniti vengono applicati a tutte le tabelle promosse.

Questa funzione promuove solo le tabelle che sono registrate in DB2 Versione 8 o successive.

### Promuovi viste registrate

Questa funzione promuove le informazioni di registrazione per le viste specificate. Facoltativamente, questa funzione promuove definizioni di tablespace, indice, tabella base non registrata (su cui si basa una vista) e vista base. È possibile specificare un diverso schema Capture e un diverso

nome server per le viste che si promuovono. Inoltre, è possibile modificare il nome dello schema per le viste CD associate alle viste di origine promosse e tabelle CD su cui tali viste CD si basano.

È possibile promuovere più viste registrate contemporaneamente. I nuovi nomi di schema forniti vengono applicati a tutte le viste promosse.

**Importante:** se la vista che si sta promuovendo si basa su una tabella di origine registrata, è necessario promuovere la tabella di origine registrata separatamente utilizzando la funzione di promozione delle tabelle registrate. Le tabelle di origine registrate non vengono promosse automaticamente dalla funzione di promozione della vista registrata. Tuttavia, le tabelle di base non registrate, su cui si basa questa vista, vengono promosse da questa funzione, se richiesto.

#### **Promuovi serie di richieste**

Questa funzione promuove le serie di richieste. Questa funzione abilita la copia di una serie di richieste (con tutti i relativi membri di serie di richieste) da un database ad un altro.

È necessario utilizzare la funzione di promozione della serie di richieste con la funzione di promozione delle tabelle registrate.

**Importante:** È possibile utilizzare le funzioni di promozione per promuovere oggetti registrati e serie di richieste che risiedono su tutti i sistemi operativi supportati. Le funzioni di promozione copiano le definizioni di replica solo tra sistemi simili, ad esempio da un DB2 per il sistema z/OS su un altro DB2 per il sistema z/OS.

Non è possibile utilizzare le funzioni di promozione per copiare definizioni della replica su o da database relazionali non DB2. Inoltre, non è possibile utilizzare le funzioni di promozione per copiare definizioni della replica che includono giornali remoti System i.

---

## Capitolo 13. Gestione di un ambiente di replica SQL

È necessario gestire i sistemi di origine, le tabelle di controllo e le tabelle di destinazione che risiedono sul database in uso e che vengono utilizzati dalla replica SQL.

La replica SQL opera con il sistema di database e richiede un numero limitato di modifiche alle attività del database esistenti. Tuttavia, per garantire che l'intero sistema continui a funzionare correttamente e per evitare problemi potenziali, è necessario determinare i requisiti di elaborazione dell'ambiente di replica in uso e l'impatto potenziale di tali requisiti sul sistema di database.

I seguenti argomenti trattano la gestione dei requisiti dei sistemi di origine, le tabelle di controllo e le tabelle di destinazione.

---

### Gestione di sistemi origine

Il sistema di origine della replica contiene il meccanismo di modifica e acquisizione, le tabelle di origine che si desidera replicare (compresi eventuali giornali remoti utilizzati su System i), i dati della registrazione utilizzati dal programma Capture ed eventuali trigger Capture usati su origini di database relazionali non DB2.

Questi argomenti spiegano come gestire correttamente le tabelle di origine e i file di log e come accertarsi che tali tabelle e file siano sempre accessibili alla replica SQL.

### Accesso alle viste e alle tabelle di origine

È necessario considerare la disponibilità delle tabelle di origine per la replica SQL in modo che i programmi Capture e Apply siano sempre in grado di continuare.

Gli oggetti di origine di replica sono viste e tabelle di database che richiedono la stessa gestione delle altre viste e tabelle di database sul sistema in uso. Continuare ad eseguire i programmi di utilità e le routine di gestione esistenti su tali oggetti.

La replica SQL non richiede accesso diretto alle tabelle di origine durante la maggior parte dell'elaborazione della replica. Tuttavia, la replica SQL deve accedere direttamente alle tabelle di origine o agli spazi tabella quando il programma Apply esegue un aggiornamento completo:

### Registrazioni origine e ricevitori di giornale

Le registrazioni di recupero DB2 servono a due scopi: fornire funzionalità di recupero DB2 e fornire informazioni ai programmi Capture in esecuzione.

È necessario conservare dati di registrazione sia per il recupero DB2 che per la replica SQL e, prima di eliminare tali dati, si deve essere assolutamente certi che i programmi Capture e DB2 siano completamente finiti con una serie di registrazioni o ricevitori di giornale.

**Nota:** La replica SQL non utilizza dati di registrazione da database relazionali non DB2.

## Conservazione dei dati di registrazione (Linux, UNIX, Windows)

Linux UNIX Windows

I dati della registrazione risiedono nei buffer di registrazione, nelle registrazioni attive o nelle registrazioni dell'archivio. Ogni volta che il programma Capture viene avviato a caldo, richiede tutte le registrazioni DB2 create dal momento in cui è stato arrestato e le registrazioni DB2 non elaborate completamente.

### Prima di iniziare

**Nota:** È necessario configurare il database all'uso di archiviazioni uscita utente affinché i programmi Capture utilizzati richi amino i dati da registrazioni archiviate.

### Informazioni su questa attività

Se il programma Capture viene eseguito ogni volta che DB2 è in esecuzione, tale programma è generalmente aggiornato con le registrazioni di recupero di DB2. Se si eseguono programmi Capture ogni volta che DB2 è attivo o si conservano record di registrazione per una settimana o per un periodo superiore, è possibile continuare ad utilizzare le procedure di conservazione delle registrazioni esistenti. Tuttavia, è necessario modificare le procedure di conservazione delle registrazioni per accogliere la replica SQL se:

- I record della registrazione vengono eliminati appena DB2 completa un backup e tali record di registrazione non sono più necessari per il recupero.
- Sono presenti vincoli di memorizzazione ed è necessario eliminare spesso le registrazioni di recupero archiviate.

### Procedure

Per determinare quali record di registrazione devono essere conservati per essere utilizzati dal programma Capture e quali possono essere eliminati:

1. Eseguire la seguente istruzione SQL per ottenere il valore MIN\_INFLIGHTSEQ dalla tabella IBMSNAP\_RESTART:

**Per database con partizioni:** In un ambiente con più partizioni, questa procedura deve essere estesa a ciascuna partizione in quanto ogni partizione gestisce la propria serie di file di log. Utilizzare la colonna SEQUENCE della tabella IBMSNAP\_PARTITIONINFO per determinare queste informazioni per ciascuna partizione.

```
SELECT MIN_INFLIGHTSEQ
FROM ASN.IBMSNAP_RESTART
WITH UR;
```

Viene visualizzato il valore MIN\_INFLIGHTSEQ. Il valore MIN\_INFLIGHTSEQ è una colonna CHAR(10) PER DATI BIT, che risulta come 20 caratteri esadecimali. Ad esempio:

```
00000000123456123456
```

Prendere nota degli ultimi 12 caratteri del valore MIN\_INFLIGHTSEQ.

Nell'esempio:

```
123456123456
```



**Attenzione:** il programma Capture aggiorna IBMSNAP\_RESTART ogni volta che esegue il commit dei dati, in base al valore del parametro **commit\_interval**. Poiché l'istruzione SELECT utilizzata in questa procedura specifica un UR (Uncommitted Read), si potrebbe ricevere un valore di cui non è stato eseguito il commit per MIN\_INFLIGHTSEQ. Per accertarsi che si dispone del valore più preciso, eseguire l'istruzione SELECT, attendere che trascorra l'intervallo di commit e quindi eseguire nuovamente l'istruzione SELECT. Utilizzare il valore inferiore per MIN\_INFLIGHTSEQ per il resto di questa procedura.

- Da una riga comandi, digitare il comando db2 get db cfg per ottenere il percorso per i file di log attivi. Ad esempio:

```
db2 get db cfg for yourdbname
```

dove *yourdbname* è il nome del database. Dall'output visualizzato sullo schermo, prendere nota del percorso per i file di log attivi. Ad esempio:

```
Path to log files =C:\DB2\NODE0000\SQL00001\SQLLOGDIR\
```

- Da una riga comandi DB2, digitare il comando db2f1sn ed immettere gli ultimi 12 caratteri del valore MIN\_INFLIGHTSEQ. Ad esempio:

```
C:\DB2\NODE0000\SQL00001>db2f1sn 123456123456
```

Per eseguire il comando db2f1sn, occorre avere accesso al file SQLOGCTL.LFH.1 o alla sua copia mirror, SQLOGCTL.LFH.2. Entrambi i file si trovano nella directory del database. Il sistema richiama e visualizza il nome del file che contiene il record di registrazione identificato dal numero di sequenza di registrazione. Ad esempio:

```
Given LSN is contained in the log file S000123.LOG
```

## Access to journal receivers (System i)

### System i

It is important to retain all journal receivers that are required by the Capture program.

When you restart the Capture program with the RESTART(\*YES) parameter, the Capture program continues processing from where it ended previously and requires all the journal receivers used by one or more of the source tables.

To make certain your Capture program can access all required journal receivers, use the delete journal receiver exit program, which was registered automatically when you installed DB2 DataPropagator for System i. This exit program is invoked any time you or one of your applications programs attempts to delete a journal receiver. This exit program then determines whether or not a journal receiver can be deleted.

**Recommendation:** Specify DLTRCV(\*YES) and MNGRCV(\*SYSTEM) on the CHGJRN or CRTJRN command to use the delete journal receiver exit program and leave journal management to the system.

If the journal receiver is used by one or more source tables, the delete journal receiver exit program checks that the receiver being deleted does not contain entries that have not been processed by the Capture program. The exit program *disapproves* the deletion of the receiver if the Capture program still needs to process entries on that receiver.

## Considerazioni per la gestione dei dizionari di compressione (z/OS)

z/OS

Se si utilizzano le utilità del dizionario di compressione DB2, è necessario coordinare l'uso di tali utilità con i propri programmi Capture.

### Aggiornamento dei dizionari di compressione DB2 (z/OS)

Quando il programma Capture richiede i record di log, il DB2 deve decomprimere i record di log relativi a ciascuna tabella memorizzata in uno spazio tabella compresso. DB2 utilizza il dizionario di compressione corrente per la decompressione. In alcuni casi il dizionario di compressione può non essere disponibile. Il programma Capture intraprende azioni differenti in ciascun caso:

#### Se il dizionario di compressione è temporaneamente non disponibile

DB2 restituisce un errore al programma Capture. Il programma Capture compie vari tentativi per continuare l'elaborazione. Se il dizionario è ancora non disponibile, il programma Capture emette un messaggio ASN0011E e verrà terminato.

#### Se il dizionario di compressione non è disponibile in modo permanente

È possibile che si perda un dizionario di compressione se si utilizza il programma di utility REORG senza specificare `KEEPDICTIONARY=YES`. In questo caso, il programma Capture segue l'azione di errore specificata per l'opzione `STOP_ON_ERROR` relativa alla registrazione. Se `STOP_ON_ERROR=N` (no), Capture disattiva la registrazione. Se `STOP_ON_ERROR=Y` (sì), il programma Capture emette un messaggio ASN0011E e verrà terminato.

Con l'APAR PK19539 (DB2 per z/OS, Versione 8), DB2 manterrà una copia di backup del dizionario di compressione in memoria quando si utilizza il programma di utilità REORG senza specificare `KEEPDICTIONARY=YES`. Non è quindi necessario specificare `KEEPDICTIONARY=YES` se non:

- Si riavvia il DB2.
- Il programma di utilità REORG deve essere utilizzato due volte per lo stesso spazio tabella, prima che il programma Capture legga tutti i record di log precedenti relativi a tale tabella.

Per evitare queste situazioni in DB2 per z/OS, Versione 7, attendere che il programma Capture completi l'elaborazione di tutti i record di log relativi a una tabella, prima di eseguire attività che influenzino il dizionario di compressione della tabella in questione. Possono influenzare i dizionari di compressione, alcune delle attività seguenti:

- Il cambiamento di uno spazio tabella per modificare l'impostazione di compressione
- L'utilizzo di `DSN1COPY` per copiare gli spazi tabella compressi da un sottosistema ad un altro, comprendenti ambienti con condivisione o non condivisione dei dati
- L'esecuzione del programma di utilità REORG sullo spazio tabella

### Blocco dei dizionari di compressione DB2 (z/OS)

È necessario, inoltre, considerare la disponibilità del dizionario di compressione. Quando il programma Capture legge i record di log compressi, DB2 pone un vincolo sullo spazio tabella di origine compresso,

per accedere al dizionario. Il programma Capture verrà arrestato se lo spazio tabella compresso sul sistema di origine è in stato STOPPED quando DB2 LRILog Read Interface necessita di questo vincolo. Al contrario, un'utilità che richiede l'accesso completo allo spazio tabella di origine, o che richiede che quest'ultimo sia in uno stato STOPPED, può essere esclusa dal blocco causato dal vincolo mantenuto dal programma Capture durante la lettura del dizionario.


Per prevenire eventuali blocchi temporanei dovuti a un vincolo non disponibile, sospendere il programma Capture quando uno spazio tabella compresso deve essere utilizzato esclusivamente da un'utilità DB2 (o vendor).

---

## Gestione delle tabelle di controllo

La replica SQL utilizza tabelle di controllo per memorizzare definizioni origine, definizioni di serie di richieste e altre informazioni di controllo specifiche della replica. Sebbene la dimensione di alcune tabelle di controllo resti statica, altre tabelle di controllo possono svilupparsi (e successivamente ridursi) dinamicamente in base alla dimensione del database ed ai requisiti della replica.

La dimensione delle seguenti tabelle cambia di frequente durante l'elaborazione normale:

-  IBMSNAP\_APPLY\_JOB
- IBMSNAP\_APPLYTRACE
- IBMSNAP\_APPLYTRAIL
- IBMSNAP\_CAPMON
- IBMSNAP\_CAPTRACE
- tabelle CD
- tabelle CCD
- IBMSNAP\_ALERTS
- IBMSNAP\_MONTRACE
- IBMSNAP\_MONTRAIL
- IBMSNAP\_SIGNAL
- BMSNAP\_SUBS\_EVENT
- IBMSNAP\_UOW

La dimensione e lo sviluppo di queste tabelle di controllo dinamiche può influire sulle prestazioni del sistema in uso.

## Programma di utilità RUNSTATS per la replica SQL (Linux, UNIX, Windows, z/OS)

 z/OS

 Linux UNIX Windows

Il programma di utilità RUNSTATS aggiorna le statistiche relative alle caratteristiche fisiche delle tabelle e degli indici associati.

È necessario continuare ad eseguire il programma di utilità RUNSTATS sulle tabelle esistenti con la stessa frequenza con cui veniva eseguito prima di utilizzare la replica SQL. Tuttavia, è necessario eseguire il programma di utilità RUNSTATS sulle tabelle CD (Change-Data), IBMSNAP\_UOW e altre tabelle controllo dinamiche solo una volta quando tali tabelle contengono quantità elevate di dati.

RUNSTATS riporta informazioni utili relative a tali tabelle dinamiche quando le tabelle sono alle dimensioni massime del livello di produzione e l'ottimizzatore raccoglie le statistiche necessarie per determinare la migliore strategia per l'accesso ai dati.

## Rebind di pacchetti e piani (z/OS, Linux, UNIX, Windows)

z/OS

Linux UNIX Windows

L'esecuzione del bind dei pacchetti e dei piani in uso con il livello di isolamento impostato su UR (Uncommitted Reads) garantisce prestazioni di sistema ottimali.

Molti piani e pacchetti di replica SQL vengono collegati utilizzando l'isolamento UR. Se si deve eseguire il rebind dei pacchetti e dei piani in uso, tenere presente che i programmi di gestione interni utilizzati per il rebind automatico di tali pacchetti e piani possono causare problemi di conflitto tra il programma Capture ed il programma Apply se tali programmi eseguono il rebind dei pacchetti di replica con opzioni standard come CS (Cursor Stability). I pacchetti per la replica SQL devono rimanere collegati con l'isolamento UR per mantenere prestazioni di sistema ottimali.

## Riorganizzazione delle tabelle di controllo

È necessario riorganizzare regolarmente le tabelle di controllo dinamiche che vengono aggiornate di frequente.

### Informazioni su questa attività

Le tabelle CD e IBMSNAP\_UOW ricevono diversi INSERTS durante l'acquisizione della modifica e diversi DELETES durante l'eliminazione. La dimensione delle tabelle IBMSNAP\_CAPMON, IBMSNAP\_CAPTRACE e IBMSNAP\_APPLYTRAIL può variare considerevolmente a seconda della velocità di aggiornamento delle tabelle di origine della replica in uso.

**suggerimento:** Riorganizzare le seguenti tabelle di controllo dinamiche una volta alla settimana:

- tabelle CD
- IBMSNAP\_ALERTS
- IBMSNAP\_APPLYTRACE
- IBMSNAP\_APPLYTRAIL
- IBMSNAP\_CAPMON
- IBMSNAP\_CAPTRACE
- IBMSNAP\_MONTRAIL
- IBMSNAP\_MONTRACE
- IBMSNAP\_UOW

Non è necessario eseguire eventuali programmi di utilità che richiedono spazio non utilizzato o generare statistiche dell'ottimizzatore aggiornate di frequente sulle altre tabelle di controllo.

### Procedure

Per riorganizzare le tabelle di controllo in uso, utilizzare uno dei metodi riportati di seguito:

Metodo	Descrizione
<b>z/OS</b> Utilità REORG con l'opzione PREFORMAT	L'opzione PREFORMAT di questo programma di utilità velocizza il processo di inserimento del programma Capture.
<b>System i</b> Comando RGZPFM (Reorganize Physical File Member)	È possibile riorganizzare la tabella UOW e le tabelle CD attive quando il programma Capture termina specificando il parametro RGZCTLTBL(*YES) sul comando ENDDPRCAP.
<b>Linux UNIX Windows</b> Comando REORG	Utilizzare questo comando per eliminare i dati frammentati e recuperare spazio.

## Eliminazione di tabelle di controllo dinamiche gestite dai programmi Capture (Linux, UNIX, Windows, z/OS)

**z/OS**

**Linux UNIX Windows**

È possibile eliminare manualmente o automaticamente le tabelle la cui dimensione varia.

### Informazioni su questa attività

È necessario controllare lo sviluppo e considerare i vari metodi di eliminazione disponibili per le seguenti tabelle di controllo dinamiche:

- tabelle CD
- IBMSNAP\_UOW
- IBMSNAP\_CAPMON
- IBMSNAP\_CAPTRACE
- IBMSNAP\_SIGNAL
- **System i** IBMSNAP\_AUTHTKN

È possibile impostare i programmi Capture in uso all'eliminazione di tali tabelle automaticamente ad intervalli regolari. Oppure è possibile eseguire l'eliminazione a richiesta, avviando una volta il processo di eliminazione; il programma Capture non esegue una nuova eliminazione finché non si immette un altro comando di eliminazione.

### Procedure

Per eliminare tabelle di controllo dinamiche gestite dal programma Capture:

1. Se si desidera eliminare automaticamente le tabelle di controllo dinamiche, impostare il parametro **autoprune** su Sì utilizzando uno dei seguenti metodi:

Metodo	Descrizione
Avvio di un programma con l'eliminazione automatica.	Immettere il comando di sistema asncap con <b>autoprune=y</b> . Impostare il parametro <b>prune_interval</b> per specificare la frequenza del processo di eliminazione automatica.

Metodo	Descrizione
<b>Abilitare l'eliminazione automatica per un programma Capture in esecuzione.</b>	Immettere il comando di sistema <code>asnccmd chgparms con autoprune=y</code> . Impostare il parametro <code>prune_interval</code> per specificare la frequenza del processo di eliminazione automatica.

2. Se si desidera eliminare le tabelle di controllo dinamiche una volta, utilizzare uno dei seguenti metodi:

Metodo	Descrizione
<b>Centro di replica</b>	Utilizzare la finestra Elimina tabelle di controllo Capture per eliminare le tabelle in una sola volta. Per visualizzare la finestra, fare clic sulla cartella <b>Server di controllo Capture</b> nel ramo <b>Operazioni</b> della struttura ad albero dell'oggetto, nel pannello dei contenuti fare clic con il tasto destro del mouse su un server quindi fare clic su <b>Elimina Capture</b> .
<b>Avviare l'eliminazione una volta da un programma Capture in esecuzione.</b>	Immettere il comando di sistema <code>asnccmd</code> con il parametro di eliminazione.

## Eliminazione tabella UOW e CD

Durante ogni ciclo di eliminazione, sia se richiamato automaticamente che a richiesta, il programma Capture elimina le tabelle CD e UOW in base all'avanzamento riportato dai programmi Apply.

L'avanzamento dell'eliminazione è indicato dal valore della colonna SYNCHPOINT nella tabella IBMSNAP\_PRUNE\_SET. Questa normale eliminazione si basa sul valore minimo per il punto di sincronizzazione su tutti i programmi Apply che sottoscrivono ciascuna tabella CD e sul valore minimo complessivo per il punto di sincronizzazione relativo alla tabella UOW.

L'eliminazione normale, tuttavia, non elimina efficacemente le tabelle CD e UOW se le serie di richieste associate vengono eseguite molto raramente. Tenere presente l'efficacia dell'eliminazione quando si decide la frequenza di esecuzione dei programmi Apply associati, quando si arrestano tali programmi Apply e quando si disattivano le serie di richieste per più di un breve periodo di tempo.

Se le serie di richieste vengono eseguite molto raramente o si arrestano i programmi Apply, le tabelle CD e UOW possono assumere grandi dimensioni e diventare idonee per l'eliminazione del limite di conservazione. Il limite di conservazione è un parametro operativo del programma Capture, con un valore predefinito di una settimana. Esso determina la durata di permanenza dei vecchi dati nelle tabelle prima che divengano idonei per l'eliminazione del limite di conservazione.

Se il processo di eliminazione normale è inibito a causa di serie di richieste disattivate o eseguite raramente, i dati possono rimanere nella tabella per lunghi periodi di tempo. Se tali dati divengono più obsoleti della data/ora attuale DB2 meno il valore limite di conservazione, il processo di eliminazione del limite di conservazione elimina tale dati dalle tabelle.

Cercare di evitare le condizioni che richiedono l'eliminazione del limite di conservazione, in quanto l'accumulazione di vecchi dati può causare l'overflow della memoria e un degrado delle prestazioni.

**suggerimento:** Eseguire i programmi Apply almeno una volta al giorno per tutte le serie di richieste.

Se il server di origine fornisce dati modificati ad una varietà di sistemi di destinazione, ognuno con requisiti molto diversi ed alcuni con programmi Apply eseguiti raramente per poche origine registrate, considerare l'uso di più programmi Capture. È possibile utilizzare più programmi Capture e gestire i vari requisiti di elaborazione con diversi schemi Capture, utilizzando uno schema Capture per isolare quelle tabelle che vengono eliminate raramente a causa di specifici requisiti temporali di serie di richieste e utilizzando un altro schema Capture per le tabelle di origine rimanenti.

## **Suggerimenti per l'eliminazione di altre tabelle di controllo dinamiche**

È necessario eliminare regolarmente le tabelle di controllo replica per rimuovere dati obsoleti e per migliorare le prestazioni del sistema.

Il programma Capture esegue operazioni di eliminazione solo sulle tabelle che gestisce. Il programma Apply gestisce tabelle CCD (Consistent-Change Data); quindi, il programma Capture non elimina automaticamente tali tabelle. Alcuni tipi di tabelle CCD non richiedono l'eliminazione. Le tabelle CCD complete e compresse vengono aggiornate sul posto.

Gli unici record che si potrebbe voler rimuovere dalle tabelle CCD complete e compresse sono quelli con valore di colonna IBMSNAP\_OPERATION D (Delete) che sono già stati replicati sulle tabelle di destinazione dipendenti. Le tabelle CCD non compresse contengono dati cronologici e possono avere un grande sviluppo. Poiché tali dati devono essere conservati per fini di controllo, non si devono eseguire operazioni di eliminazione sulle tabelle CCD non compresse.

È necessario, tuttavia, considerare l'eliminazione delle tabelle CCD interne. Tali tabelle possono svilupparsi rapidamente se sul sistema si verifica un'elevata attività di aggiornamento. Dalle tabelle CCD interne vengono estratte solo le modifiche più recenti, non è quindi necessario conservare le righe più obsolete.

Per abilitare l'eliminazione per le tabelle CCD interne, considerare di aggiungere istruzioni dopo SQL alle serie di richieste associate per eliminare dati di modifica che sono già stati applicati a tutte le destinazioni dipendenti. In alternativa, per eliminare le righe da tali tabelle, è anche possibile aggiungere le necessarie istruzioni SQL DELETE alle funzioni di pianificazione automatica.

È necessario inoltre eliminare manualmente le tabelle IBMSNAP\_APPLYTRAIL and IBMSNAP\_APPLYTRACE. Se si definiscono e utilizzano più serie di richieste con programmi Apply eseguiti di frequente, la tabella IBMSNAP\_APPLYTRAIL si sviluppa rapidamente e richiede un'eliminazione frequente. Il modo ottimale di gestione dello sviluppo di tali tabelle consiste nell'aggiungere un'istruzione dopo SQL o un richiamo della procedura ad una delle serie di richieste. In alternativa, è possibile aggiungere un'istruzione SQL DELETE alle funzioni di pianificazione automatica.

## Impedimento malfunzionamenti di replica e recupero da errori

Questi argomenti descrivono i metodi per impedire e recuperare da malfunzionamenti di replica che possono interessare le tabelle di controllo ed i dati di replica.

### Impedimento di avvisi a freddo del programma Capture

L'avvio a freddo del programma Capture deve essere eseguito solo se si sta avviando il programma per la prima volta o se si ha necessità di aggiornare le tabelle di destinazione e controllo in uso. Se il programma Capture viene avviato a freddo, vengono aggiornate tutte le tabelle di destinazione nell'ambiente di replica.

**z/OS** **Linux UNIX Windows** Quando un programma Capture viene avviato con l'opzione warmns o warmsi, il programma cerca di richiamare i record di registrazione in base al punto di riavvio presente nella tabella IBMSNAP\_RESTART. Se il programma Capture non individua la registrazione, il relativo avvio a caldo ha esito negativo.

Per impedire un avvio a freddo del programma Capture, considerare i seguenti suggerimenti.

- **System i** Avviare il programma Capture con il parametro RESTART(\*YES). Il programma Capture continua l'elaborazione dal punto in cui ha terminato in precedenza. Conservare sufficienti dati di registrazione DB2 o ricevitori di giornale sul sistema e fare in modo che questi dati siano disponibili alla replica SQL.
- Utilizzare Replication Alert Monitor o altro meccanismo per verificare lo stato dei dati cronologici dai programmi Capture. È possibile quindi utilizzare tali informazioni per verificare che i programmi Capture siano sempre in esecuzione se DB2 è attivo.
- Accertarsi di conservare sufficienti dati di registrazione DB2 o ricevitori di giornale sul sistema in uso e che questi dati siano disponibili alla replica SQL.

### Recupero da errori di I/O e malfunzionamenti di connettività sulle tabelle di controllo in uso

Se la replica perde la connettività ad una tabella di controllo, è possibile recuperare la tabella, per altri errori i programmi di replica verranno chiusi.

#### Informazioni su questa attività

Se il programma Capture rileva un errore di I/O o un malfunzionamento di connettività, il programma emette un appropriato messaggio di errore e si chiude.

Il programma Apply si chiude se rileva errori gravissimi sulle tabelle di controllo. Se il programma Apply rileva errori sulle tabelle di destinazione o errori con la connettività di rete, scrive l'errore sulla tabella IBMSNAP\_APPLYTRAIL e quindi continua l'elaborazione.

#### Procedure

Per il recupero da errori e malfunzionamenti di connettività sulle tabelle di controllo:

1. Se si rileva un errore di I/O o un malfunzionamento di connettività su una tabella di controllo, utilizzare una procedura di recupero DB2 standard per avviare il recupero della tabella. La tabella non perderà alcun dato.



2. Se il programma si chiude, riavviare il programma Capture dal punto di malfunzionamento quindi riavviare il programma Apply.

### Richiamo di dati origine persi

Se si perde l'origine, è possibile richiamarla mediante un metodo punto di recupero o un aggiornamento completo.

#### Informazioni su questa attività

Se una tabella di origine viene quindi recuperata nel punto di malfunzionamento, la replica SQL procede normalmente. Una volta ripristinata la tabella, il programma Capture continua la raccolta delle modifiche dei dati per la tabella.

Tuttavia, i programmi Capture e Apply non individuano il recupero di un determinato momento di una tabella di destinazione di sola lettura. Se si recupera una tabella di origine, il programma Apply potrebbe aver replicato le modifiche nelle tabelle di destinazione, non più esistenti all'origine, lasciando così delle incongruenze tra le tabelle di origine e le tabelle di destinazione se non si può riportare le tabelle di destinazione allo stesso momento logico.

Tale scenario diventa ancora più complesso quando sono presenti più livelli di replica. Come metodo di scelta di recupero è necessario sviluppare un meccanismo che fornisca punti di recupero corrispondenti tra i vari livelli o utilizzare un aggiornamento completo.

#### Procedure

Recuperare i dati di origine utilizzando uno dei seguenti metodi:

Metodo	Descrizione
<b>Meccanismo del punto di recupero</b>	Sviluppare un meccanismo che fornisca punti di recupero corrispondenti tra i vari livelli di replica.
<b>Aggiornamento completo</b>	Utilizzare un aggiornamento completo come metodo di scelta di recupero

### Eliminazione tabella IBMSNAP\_CAPMON e IBMSNAP\_CAPTRACE

I valori del parametro di funzionamento in uso determinano l'eliminazione delle tabelle IBMSNAP\_CAPMON e IBMSNAP\_CAPTRACE.

Durante ogni ciclo di eliminazione, il programma Capture elimina le tabelle IBMSNAP\_CAPMON e IBMSNAP\_CAPTRACE in base ai valori dei seguenti parametri di funzionamento del programma Capture:

- Il parametro **monitor\_limit** (Linux, UNIX, Windows, z/OS) e il parametro **MONLMT** (System i) determinano la durata di permanenza delle righe nella tabella IBMSNAP\_CAPMON
- Il parametro **trace\_limit** (Linux, UNIX, Windows, z/OS) e il parametro **TRCLMT** (System i) determinano la durata di permanenza delle righe nella tabella IBMSNAP\_CAPTRACE

Sia il limite di controllo che il limite di traccia hanno un valore predefinito di una settimana. È possibile modificare tali valori a seconda di quanto è necessario conservare la cronologia della latenza Capture e le informazioni di velocità di trasmissione nella tabella IBMSNAP\_CAPMON ed i dati di controllo e risoluzione dei problemi dalla tabella IBMSNAP\_CAPTRACE.

## Eliminazione tabella IBMSNAP\_SIGNAL

Poiché durante la replica vengono costantemente aggiunte righe, la tabella IBMSNAP\_SIGNAL viene eliminata automaticamente.

La tabella IBMSNAP\_SIGNAL viene eliminata anche durante ogni ciclo di eliminazione. Una riga dei segnali è idonea per l'eliminazione se il valore della colonna SIGNAL\_STATE è uguale a C. Un valore C indica che le informazioni sul segnale sono complete e non più necessarie al programma Capture o per qualsiasi elaborazione utente e sono idonee per l'eliminazione. Una riga segnali con un valore colonna SIGNAL\_TIME più obsoleto della data/ora attuale DB2 meno il valore del parametro del limite di conservazione è idonea per l'eliminazione del limite di conservazione.

---

## Manutenzione delle tabelle di destinazione

Gestire le tabelle sul server di destinazione nello stesso modo in cui vengono gestite le altre tabelle nel sistema di database.

Utilizzare le routine di gestione e backup correnti su tali tabelle di destinazione, indipendentemente dal fatto che le tabelle di destinazione siano tabelle del database esistenti o tabelle specificate in modo da essere generate automaticamente dalla replica SQL.

**Nota:** Disattivare i programmi Apply prima di impostare una tabella di destinazione su fuori linea per eseguire i programmi di utilità.

---

## Capitolo 14. Table differencing and repair

The `asntdiff` and `asntrep` utilities allow you to detect and repair differences between source and target tables in Q replication and SQL replication without manually comparing the tables or performing a load (full refresh) of the target.

### Informazioni su questa attività

Source and target tables can lose synchronization, for example if a target table is unexpectedly changed by a user or application, or if you experienced an extended network or target system outage.

The `asntdiff` and `asntrep` utilities run independently of the Q Capture, Q Apply, Capture, and Apply programs. They use DB2 SQL to fetch data from the source table and the target table and do not use WebSphere MQ queues. The utilities do not depend on logs, triggers, or isolation level.

### Procedure

To detect and repair differences between source and target tables, run the `asntdiff` utility, and then run the `asntrep` utility.

---

## Table difference utility (`asntdiff`)

The `asntdiff` utility compares all columns in a source table to their corresponding columns in a target table and generates a list of differences between the two tables in the form of a DB2 table.

To use the `asntdiff` utility, you run the `asntdiff` command and specify the name of a Q subscription (Q replication) or subscription set member (SQL replication) that contains the source and target tables that you want to compare.

The following sections explain how to use the `asntdiff` command:

- “Overview of the `asntdiff` command”
- “Difference table” a pagina 210
- “Suppressed delete operations” a pagina 211
- “Different data types in sources and targets” a pagina 212
- “Comparing the GRAPHIC data type” a pagina 212
- “Predicates” a pagina 212
- “When to use the `asntdiff` utility” a pagina 212
- “Using an input file to specify tables to compare” a pagina 213
- “Specifying location and size of temporary files or data sets” a pagina 213

### Overview of the `asntdiff` command

You can run the `asntdiff` command on Linux, UNIX, Windows, and z/OS operating systems. The command compares tables on Linux, UNIX, Windows, z/OS, or System i operating systems. The `asntdiff` command can be used with federated sources and targets if the corresponding columns in the two tables have the same data types.

**Nota:** The ASNTDIFF sample job in the SASNSAMP data set provides further information that is specific to the z/OS platform.

For Q replication, the target must be a table and not a stored procedure. For SQL replication, the target must be a user table, point-in-time table, replica table, or user-copy table.

When you run the command, you specify an SQL WHERE clause that uniquely identifies the Q subscription or subscription set member:

### Q replication

The WHERE clause identifies a row in the IBMQREP\_SUBS control table at the Q Capture server, based on the value of the SUBNAME column. For example:

```
where="subname = 'my_qsub'"
```

### SQL replication

The WHERE clause identifies a row in the IBMSNAP\_SUBS\_MEMBR table at the Apply control server, based on the value of the SET\_NAME column. For example:

```
where="set_name = 'my_set' and source_table='EMPLOYEE'"
```

You might need to use more predicates in the WHERE clause to uniquely identify the subscription set member. For example, you might need to add the APPLY\_QUAL, the SOURCE\_OWNER, the TARGET\_OWNER, or the TARGET\_TABLE column from the IBMSNAP\_SUBS\_MEMBR table to the clause.

## Difference table

The asntdiff command creates a difference table in the source database or subsystem for Q replication and SQL replication.

The difference table is named *schema*.ASNTDIFF, where *schema* is the value specified in the DIFF\_SCHEMA parameter. If the schema is not specified, it defaults to ASN. You can also use the DIFF parameter to specify a table name.

By default, the difference table is created in the default DB2 user table space. You can specify a different, existing table space by using the DIFF\_TABLESPACE parameter.

The difference table has two or more columns. One column is named DIFF, with a blank space at the end on Linux, UNIX, and Windows. The value in the DIFF column is a character that indicates an insert, update, or delete operation followed by a numeric value that indicates which table contains a row with differences. The other columns contain the value of replication key columns. There is one row in the difference table for each unmatched row in the target table.

The difference table uses three identifiers that indicate the operation that is needed to change the target table so that it matches the source table:

#### D (delete)

Indicates that a row with the key value exists only at the target and not at the source.

#### U (update)

Indicates that rows with the same key value exist at both the source and target, but at least one non-key column is different at the target.

### **I (insert)**

Indicates that a row with the key value exists only at the source and not at the target.

A value of ? 1 indicates that there is an invalid character in one or more source columns.

A value of ? 2 indicates that there is an invalid character in one or more target columns.

### **Example:**

The following list of values is returned by comparing an EMPLOYEE table at the source with a target copy of the same table. The key column for replication is the employee number, EMPNO:

```
DIFF  EMPNO
U 2    000010
I 2    000020
I 2    000040
D 2    000045
I 2    000050
D 2    000055
```

The first row in the example shows that a row with the key value 000010 exists at both the source and target tables, but at least one non-key column at the target has a different value. The next two rows show that rows with the key values 000020 and 000040 exist only at the source. The fourth row shows that a row with the key value 000045 exists only at the target.

The values ? 1 and ? 2 are not shown in the example.

## **Suppressed delete operations**

In Q replication, you can choose to suppress replication of delete operations from the source table. If you do not replicate delete operations, rows that exist in the target table might not exist in the source table. When the SUPPRESS\_DELETES value for a Q subscription is Y, the asntdiff utility ignores the rows that are unique to the target and reports no differences. A warning is issued to indicate how many rows were suppressed.

The asntdiff -f (input file) option does not support SUPPRESS\_DELETES because it bases the table comparison on a SQL SELECT statement rather than the Q subscription definition.

## **Restrictions for key columns at source and target**

The asntdiff utility supports multiple-byte character sets when the database is defined with SYSTEM or IDENTITY. However, the columns that are used as keys for replication at the source and target tables must use single-byte characters for the utility to compare the tables.

In a Linux, UNIX, or Windows database that uses Unicode, the characters in key data cannot be greater than the base U.S. English ASCII subset (first 256 ASCII characters) or the asntdiff utility cannot compare the tables.

## Different data types in sources and targets

The `asntdiff` utility builds two `SELECT SQL` statements that are based on the description of a subscription. To obtain the differences between the source and target tables, the utility compares the data that result from executing both statements. The data types and lengths of the columns for both `SQL` statements must be the same.

### SQL replication

The utility builds the `SQL` statement for the source by using the `EXPRESSION` column in the `IBMSNAP_SUBS_COLS` table.

### Q replication

The data types for both the source and the target must be the same.

## Comparing the GRAPHIC data type

Columns with the `GRAPHIC` data type at the source and target might not match when you use the `asntdiff` utility to compare the source and target tables. `DB2` columns with the `GRAPHIC` data type have blank padding after the graphic data. This padding might be single-byte or double-byte spaces, depending on the code page that the database was created in. This padding might cause data to not match between the source and the target tables, especially if the source and target tables are in different code pages. This padding applies only to `GRAPHIC` data types and not other graphic data types such as `VARGRAPHIC` or `LONG VARGRAPHIC`.

To compare columns with `GRAPHIC` data types, you must remove the blank padding in the data before you compare the source and target tables by using the `DB2` scalar function `rtrim(<column>)`. This function eliminates the code page differences for single-byte or double-byte spaces and ensures that the `asntdiff` utility compares the `GRAPHIC` data in a consistent manner.

## Predicates

In some cases, differences between source and target tables are intentional, for example, if you use a search condition in `Q` replication to filter which rows are replicated. The utility will not show differences between source and target tables that are a result of predicates.

### SQL replication

The utility uses the `PREDICATES` column in the `IBMSNAP_SUBS_MEMBR` table to select rows from the source tables. The value of the `UOW_CD_PREDICATES` column is ignored (`asntdiff` looks directly at the source table, where the Apply program looks at the `CD` table).

### Q replication

The utility uses the value of the `SEARCH_CONDITION` column in the `IBMQREP_SUBS` table to build the `WHERE` clause for the `SELECT` statement.

## When to use the asntdiff utility

The best time to use the `asntdiff` utility is when the source and target tables are stable. You might want to run the utility when the `Q` Capture and `Q` Apply programs or Capture and Apply programs are idle. For example, you could run the utility when the `Q` Capture program reached the end of the `DB2` recovery log and all changes are applied at the target. If applications are still updating the source, the comparison might not be accurate.

If the replication programs are running, you might need to run the `asntdiff` command more than once to get a complete picture of evolving differences between the source and target tables.

## Using an input file to specify tables to compare

The `asntdiff -f` command option enables you to do differencing by using SQL `SELECT` statements that are read from an input file. This option provides greater flexibility to do differencing between two generic tables. The `asntdiff -f` option does not use replication definitions to determine which tables and rows to compare as the standard `asntdiff` command does.

The `asntdiff -f` option works for all tables on Linux, UNIX, Windows, and z/OS. For details on this option, see “`asntdiff -f` (input file) command option” a pagina 307.

In addition to the `SELECT` statements, the input file contains the source and target database information, the difference table information, and optional parameters that specify methods for processing the differences. You can use a password file that is created by the `asnpwd` command to specify a user ID and password for connecting to the source and target databases.

**Nota:** To compare DB2 XML columns by using the `asntdiff -f` option, you need to serialize the XML column as a character large-object (CLOB) data type by using the `XMLSERIALIZE` scalar function. For example, this `SELECT` statement in the input file compares the `XMLColumn` column in the source table `Table1` to the same column in another database table (the `TARGET_SELECT` would use the same function):

```
SOURCE_SELECT="select ID, XMLSERIALIZE(XMLColumn AS CLOB) AS XMLColumn
from Table1 order by 1"
```

## Specifying location and size of temporary files or data sets

The `asntdiff` command creates temporary files or data sets for spilling data and for writing differences before inserting them into the difference table. You specify the location of the temporary files or data sets differently, depending on the platform:

### z/OS

On z/OS, the temporary files are written by default to the UNIX System Services (USS) hierarchical file system (HFS), in the home directory of the user ID that executes the `asntdiff` command. The default names are `DD:DIFFFILE` and `DD:SPILLFILE`. You can use a `DIFFFILE DD` statement to specify an alternative HFS path and file name for those files, as shown in this example:

```
//DIFFFILE DD PATH='/u/oeusr01/tdiffi12',
//          PATHDISP=(KEEP,KEEP),
//          PATHOPTS=(ORDWR,OCREAT),
//          PATHMODE=(SIRWXU,SIRGRP,SIROTH)
```

Redirecting the HFS requires you to create an empty file that can be written to or to use the above `PATHDISP` and `PATHOPTS` settings to create a new file if one does not exist.

If you want `ASNTDIFF` to write to z/OS data sets, add these two `DD` statements to your `ASNTDIFF JCL`, modifying the size specifications to match the size of your source table:

```

//SPLFILE DD DSN=&&SPILL,DISP=(NEW,DELETE,DELETE),
//          UNIT=VIO,SPACE=(CYL,(11,7)),
//          DCB=(RECFM=VS,BLKSIZE=6404)
//DIFFFILE DD DSN=&&DIFFLE,DISP=(NEW,DELETE,DELETE),
//          UNIT=VIO,SPACE=(CYL,(11,7)),
//          DCB=(RECFM=VS,BLKSIZE=6404)

```

#### Linux UNIX

You can use the TMPDIR environment variable to specify the location of the temporary spill files.

---

## Table repair utility (asntrep)

The asntrep utility repairs differences between source and target tables on all DB2 servers by deleting, inserting, and updating rows in the target table. The utility runs on Linux, UNIX, or Windows operating systems.

The asntrep utility uses the difference table that is generated by the asntdiff utility to do the following:

- Delete rows from the target table that have no matching key in the source table
- Insert rows that are in the source table but have no matching key in the target table
- Update target rows that have matching keys in the source but different non-key data

For Q replication, the target must be a table; it cannot be a stored procedure. For SQL replication, the target must be a user table, a point-in-time table, a replica table, or a user-copy table. If you use the asntrep utility with a Q subscription for peer-to-peer replication, you must repair all of the copies of a logical table two copies at a time.

To use the asntrep utility, you run the asntrep command after you run the asntdiff command. The asntrep command copies the difference table from the source database or subsystem to the target, and then uses the copy to repair the target table.

The asntrep command does not drop the difference table from the target database or subsystem. You must drop the table manually.

To use the asntrep command, you provide the same WHERE clause that you used for the asntdiff command to identify the Q subscription or subscription set member that contains the source and target tables that you want to synchronize.

During the repair process, referential integrity constraints on the target table are not dropped. An attempt to insert or delete a row from a target table can fail if the insert or delete operation violates a referential integrity constraint. Also, a duplicate source row might be impossible to repair at the target if the target has a unique index.



---

## Capitolo 15. Replication Alert Monitor

You can use the Replication Alert Monitor to monitor an SQL replication, Q replication, or event publishing environment.

The Replication Alert Monitor cannot check the status of Classic replication sources but it can monitor the DB2 or federated target servers in a Classic replication configuration.

The following topics explain how the Replication Alert Monitor works and how to configure and operate monitors for your replication or publishing environment.

---

### Monitoring replication with the Replication Alert Monitor

The Replication Alert Monitor is a program that can alert you to changes in the status of your replication environment.

When the Replication Alert Monitor is running, it automatically checks the status of replication and notifies you about certain conditions that occur in your replication environment. For example, in SQL replication, the Replication Alert Monitor can notify you when any Apply program terminates. Similarly, in Q replication, the Replication Alert Monitor can notify you when any Q Capture program deactivates a Q subscription.

**Restriction:** The Replication Alert Monitor cannot check the status of Classic replication sources but it can monitor the DB2 or federated target servers in a Classic replication configuration.

You can configure the Replication Alert Monitor in one of two ways:

#### One monitor

Typically you use one monitor when you have few replication programs to monitor. If you set up a single monitor, all the control information is stored on one server. Each monitor can monitor multiple replication programs, but the monitor checks for alerts on each server one at a time. It must check all of the other servers that it monitors before it returns to any one server.

#### Multiple monitors

Use additional monitors to monitor many replication programs, prioritize the monitoring of certain programs, or split up the monitoring workload. You create independent monitors to check the servers in your system. These monitors do not communicate with each other, but they each send alerts about the servers. When you set up multiple monitors, the control information for each monitor is stored on the server that it is assigned to monitor. Use multiple monitors to:

- **Monitor some replication programs more frequently than others.** Set up a monitor with a smaller `monitor_interval` to check replication programs for alert conditions more frequently. For example, you can assign one monitor to monitor one Capture server for the `CAPTURE_WARNINGS` alert condition every 15 minutes. You can assign another monitor to monitor another Capture server for the `CAPTURE_WARNINGS` alert condition every 50 minutes.

- **Monitor different applications separately.** Set up monitors for each replication application. For example, separate monitors can send alerts to different groups or help an administrator separate the alerts for two different applications. Similarly, separate monitors can be assigned to check for different alert conditions.
- **Prioritize alert conditions.** For example, you might want to monitor the status of a Q Apply program every 10 minutes by using the QAPPLY\_STATUS alert condition. But, you might also want to monitor the memory of the same Q Apply program every 300 minutes by using the QAPPLY\_MEMORY alert condition.

The following terms describe components of the Replication Alert Monitor:

**Monitor**

A monitor is one instance or occurrence of the Replication Alert Monitor. You can set up a monitor to check the status of the replication programs that are running on a server or servers. Each monitor checks the replication activity on the server, or servers, that it is assigned to.

**Monitor qualifier**

A monitor qualifier is a name that you specify for a monitor. Every monitor has a unique monitor qualifier.

**Monitor control server**

A monitor control server is any server containing control information for the Replication Alert Monitor.

**Alerts** Alerts are notices that tell you about events and conditions in your replication environment. The Replication Alert Monitor sends alerts by means of e-mail or pager. You can also send alerts to the z/OS console.

**Alert conditions**

Alert conditions are conditions of the replication environment that cause the Replication Alert Monitor to send alerts. There are three kinds of alert conditions: alert conditions that are triggered by status, alert conditions that are triggered by events, and alert conditions that are triggered by thresholds.

**Alert conditions that are triggered by status**

Status alert conditions inform you about the state of the replication programs. For example, if you specify the APPLY\_STATUS alert condition, the Replication Alert Monitor will send an alert if an Apply program is not running.

**Alert conditions that are triggered by events**

Event alert conditions tell you when specific events in replication happen. For example, if you specify the QAPPLY\_ERRORS alert condition, the Replication Alert Monitor will send an alert anytime the Q Apply program records an error in the IBMQREP\_APPLYTRACE table.

**Alert conditions that are triggered by thresholds**

Threshold alert conditions tell you when a threshold has been exceed in your replication environment. For example, if you specify the QCAPTURE\_MEMORY alert condition, the Replication Alert Monitor will notify you anytime the Q Capture program uses more memory than its threshold allows.

**Contacts**

A contact is an e-mail address or a pager address where alerts from the

Replication Alert Monitor are sent. Alerts can also be directed to the z/OS console. The ASNMAIL exit routine sends e-mail notifications for the monitor. You can modify this exit routine to put the alerts elsewhere such as in a problem management system.

### Contact groups

A contact group is a collection of contacts that receive the same alerts.

#### z/OS

You can also specify that alerts are sent to the z/OS console.

The Replication Alert Monitor monitors servers on DB2 for Linux, UNIX, Windows, or z/OS operating systems.

A monitor that runs from a z/OS server can monitor the status of replication programs that are running locally or remotely on other z/OS data-sharing systems as well as on other Linux, UNIX, or Windows servers. The monitor will check the status by querying the appropriate Q Capture, Q Apply, Capture, or Apply monitor tables.

You can monitor replication programs whose control tables are at Version 8 architecture or later.

### Restrictions

- For non-DB2 relational databases, the Replication Alert Monitor does not monitor triggers that are associated with such databases used as sources in a federated database system.
- **z/OS** The Replication Alert Monitor can send e-mail notifications by using an SMTP server, but cannot use the ASNMAIL exit routine to handle notification.
- **System i** To monitor System i servers, the Replication Alert Monitor must run on a Linux, UNIX, or Windows server and monitor the System i server remotely. You cannot set up Monitor control servers on DB2 for i5/OS servers.

---

## Alert conditions and notifications for the Replication Alert Monitor

The Replication Alert Monitor can send notifications when certain alert conditions occur.

### Alert conditions for the Replication Alert Monitor

Alert conditions are conditions of the replication environment that cause a monitor to send alerts. Alerts are messages that describe the status, event or threshold that triggers an alert condition.

Some alerts also report relevant parameter values. For example, the message for the QCAPTURE\_MEMORY alert condition reports the amount of memory that the Q Capture program is using and the memory threshold value that was exceeded.

The following sections describe alert conditions that you can use to monitor your replication environment.

- “Alert conditions for the Q Capture program” a pagina 218
- “Alert conditions for the Q Apply program” a pagina 218
- “Alert conditions for the Capture program” a pagina 219
- “Alert conditions for the Apply program” a pagina 219

## Alert conditions for the Q Capture program

Tabella 17 describes the alert conditions for the Q Capture program.

*Tabella 17. Alert conditions for the Q Capture program*

Alert condition	Description
QCAPTURE_STATUS	The Replication Alert Monitor sends an alert when a Q Capture program is not running.
QCAPTURE_ERRORS	The Replication Alert Monitor sends an alert when it finds a row with the value of ERROR in the OPERATION column of the IBMQREP_CAPTRACE table.
QCAPTURE_WARNINGS	The Replication Alert Monitor sends an alert when it finds a row with the value of WARNING in the OPERATION column in the IBMQREP_CAPTRACE table.
QCAPTURE_LATENCY	Q Capture latency measures the difference between the time that data was written to the database and the time that the Q Capture program passes it on. The Replication Alert Monitor sends an alert when the Q Capture latency exceeds the threshold that you specify. Q Capture latency is measured in seconds.
QCAPTURE_MEMORY	The Replication Alert Monitor sends an alert when a Q Capture program uses more memory than the threshold that you specify. Memory is measured in megabytes.
QCAPTURE_TRANSIZE	The Replication Alert Monitor sends an alert when a transaction that the Q Capture program is processing uses more memory than the threshold that you specify. Memory is measured in megabytes.
QCAPTURE_SUBSINACT	The Replication Alert Monitor sends an alert when a Q Capture program deactivates a Q subscription.

## Alert conditions for the Q Apply program

Tabella 18 describes the alert conditions for the Q Apply program.

*Tabella 18. Alert conditions for the Q Apply program*

Alert condition	Description
QAPPLY_STATUS	The Replication Alert Monitor sends an alert when a Q Apply program is not running.
QAPPLY_ERRORS	The Replication Alert Monitor sends an alert when it finds a row with the value of ERROR in the OPERATION column in the IBMQREP_APPLYTRACE table.
QAPPLY_WARNINGS	The Replication Alert Monitor sends an alert when it finds a row with the value of WARNING in the OPERATION column in the IBMQREP_APPLYTRACE table.
QAPPLY_LATENCY	Q Apply latency measures the time it takes for a transaction to be applied to a target table after the Q Apply program gets the transaction from a receive queue. The Replication Alert Monitor sends an alert when the Q Apply latency exceeds the threshold that you specify. Q Apply latency is measured in milliseconds.
QAPPLY_EELATENCY	Q Apply end-to-end latency measures the total time that replication requires to capture changes and apply those changes to a target database. The Replication Alert Monitor sends an alert when the Q Apply end-to-end latency exceeds the threshold that you specify. Q Apply end-to-end latency is measured in seconds.

Tabella 18. Alert conditions for the Q Apply program (Continua)

Alert condition	Description
QAPPLY_MEMORY	The Replication Alert Monitor sends an alert when a Q Apply program uses more memory than the threshold that you specify. Memory is measured in megabytes.
QAPPLY_EXCEPTIONS	The Replication Alert Monitor sends an alert when a row is inserted in the IBMQREP_EXCEPTIONS table because of a conflict or SQL error at a target.
QAPPLY_RECVQINACT	The Replication Alert Monitor sends an alert when a receive queue is deactivated.
QAPPLY_SPILLQDEPTH	The Replication Alert Monitor sends an alert when the fullness of the spill queue exceeds the threshold that you specify. Fullness is expressed as a percentage. This alert condition is not supported when the Q Apply program is remote from the target database or subsystem.
QAPPLY_QDEPTH	The Replication Alert Monitor sends an alert when the fullness of any queue exceeds the threshold that you specify. Fullness is expressed as a percentage. This alert condition is not supported when the Q Apply program is remote from the target database or subsystem.

## Alert conditions for the Capture program

Tabella 19 describes the alert conditions for the Capture program.

Tabella 19. Alert conditions for the Capture program

Alert condition	Description
CAPTURE_STATUS	The Replication Alert Monitor sends an alert when a Capture program is not running.
CAPTURE_ERRORS	The Replication Alert Monitor sends an alert when it finds a row with the value of ERROR in the OPERATION column of the IBMSNAP_CAPTRACE table.
CAPTURE_WARNINGS	The Replication Alert Monitor sends an alert when it finds a row with the value of WARNING in the OPERATION column of the IBMSNAP_CAPTRACE table.
CAPTURE_LASTCOMMIT	The Replication Alert Monitor sends an alert when the time that elapsed from the last commit of a Capture program exceeds the threshold that you specify. Time elapsed is measured in seconds.
CAPTURE_CLATENCY	The current capture latency measures the difference between the time that data was written to the database and the time that the Q Capture program passes it on. The Replication Alert Monitor sends an alert when the current Capture latency exceeds the threshold that you specify.
CAPTURE_HLATENCY	Historic Capture latency is a composite of every Capture latency measurement since the monitor last checked a server for alert conditions. The Replication Alert Monitor sends an alert when the historic Capture latency exceeds the threshold that you specify.
CAPTURE_MEMORY	The Replication Alert Monitor sends an alert when a Capture program uses more memory than the threshold that you specify. Memory is measured in megabytes.

## Alert conditions for the Apply program

Tabella 20 a pagina 220 describes the alert conditions for the Apply program.

Tabella 20. Alert Conditions for the Apply program

Alert condition	Description
APPLY_STATUS	The Replication Alert Monitor sends an alert when an Apply program is not running.
APPLY_SUBSFALING	The Replication Alert Monitor sends an alert when a subscription fails.
APPLY_SUBSINACT	The Replication Alert Monitor sends an alert when a subscription is deactivated.
APPLY_ERRORS	The Replication Alert Monitor sends an alert when it finds a row with the value of ERROR in the OPERATION column in the IBMSNAP_APPLYTRACE table
APPLY_WARNINGS	The Replication Alert Monitor sends an alert when it finds a row with the value of WARNING in the OPERATION column in the IBMSNAP_APPLYTRACE table
APPLY_FULLREFRESH	The Replication Alert Monitor sends an alert when there is a full refresh.
APPLY_REJTRANS	The Replication Alert Monitor sends an alert when a transaction is rejected in a subscription set.
APPLY_SUBSDELAY	The Replication Alert Monitor sends an alert when the delay in processing a subscription is longer than the threshold that you specify.
APPLY_REWORKED	The Replication Alert Monitor sends an alert when the Apply program reworks more rows in a subscription set than the threshold that you specify.
APPLY_LATENCY	Apply end-to-end latency measures the total time that replication requires to capture changes and apply those changes to a target database. The Replication Alert Monitor sends an alert when the Apply end-to-end latency exceeds the threshold that you specify. Apply end-to-end latency is measured in seconds.

## E-mail notifications for replication alert conditions

The Replication Alert Monitor can send an e-mail when an alert condition occurs.

The content of the e-mail notification depends on whether the e-mail address that you provided is for a pager. The following examples show the type of information that you can expect in each case, for one set of alerts. The e-mail that is sent to non-pager devices shows the time when each alert condition occurred at the specific server. It also shows the number of times that each alert condition occurred and the associated message. The e-mail that the Replication Alert Monitor sends to pagers contains a summary of the parameters that triggered the alert instead of a complete message. If an alert condition occurred many times, the timestamp reflects the last time that the alert condition occurred.

### Setting ASNSENDER variable to prevent e-mail filtering

Some providers such as pager services require a full valid return address for filtering unsolicited messages. If a valid return address is not provided, the e-mail might be blocked.

If you are not receiving alerts to your e-mail address or pager, take the following steps:

1. Stop the Replication Alert Monitor.
2. Set the ASNSENDER environment variable to a valid e-mail address. For example:

```
SET ASNSENDER=replmon@server.com
```

3. Start the monitor.

### Example e-mail notification to non-pager devices (SQL replication)

```
To:      repladmin@company.com
From:    replmon@server.com
Subject: Monitor: "MONQUAL" Alerts issued
```

```
ASN5129I MONITOR "MONQUAL". The Replication Alert Monitor on
server "WSDB" reports an e-mail alert
```

```
2002-01-20-10.00.00    1 ASN0552E Capture : "ASN" The program
encountered an SQL error. The server name is "CORP". The SQL
request is "PREPARE". The table name "PROD1.INVOICESCD".
The SQLCODE is "-204". The SQLSTATE is "42704". The SQLERRMC
is "PROD1.INVOICESCD". The SQLERRP is "readCD"
```

```
2002-01-20-10.05.00    2 ASN5152W Monitor "MONQUAL". The current
Capture latency exceeds the threshold value. The Capture control
server is "CORP". The schema is "ASN". The Capture
latency is "90" seconds. The threshold is "60" seconds
```

```
2002-01-20-10.05.00    4 ASN5154W Monitor "MONQUAL". The memory
used by the Capture program exceeds the threshold value. The
Capture control server is "CORP". The schema is "ASN".
The amount of memory used is "34" bytes. The threshold is
"30" megabytes.
```

### Example e-mail notification to pagers (SQL replication)

```
To:      repladmin@company.com
From:    replmon@server.com
Subject: Monitor: "MONQUAL" Alerts issued
```

```
MONQUAL - MONDB
```

```
2002-01-20-10.00.00 ASN0552E 1 CAPTURE-ERRORS - CORP - ASN
2002-01-20-10.05.00 ASN5152W 2 CAPTURE_CLATENCY - CORP - ASN - 90 - 60
2002-01-20-10.05.00 ASN5154W 4 CAPTURE_MEMORY - CORP - ASN - 34 - 30
```

In SQL replication, the monitor groups alerts by Capture control servers and Apply control servers when it sends notifications. If a server is both a Capture control server and an Apply control server, then the monitor groups all alerts for that server together.

In Q replication, the monitor groups alerts by Q Capture servers and Q Apply servers when it sends notifications. If a server is both a Q Capture server and a Q Apply server, then the monitor groups all alerts for that server together.

If the size of the e-mail notification exceeds the limit for the type of e-mail, the monitor sends notification in multiple e-mails. The maximum size of a regular e-mail notification is 1024 characters. For a pager e-mail address the limit is 250 characters.

The ASNMAIL exit routine sends e-mail notifications for the monitor. You can modify this exit routine to handle alerts differently. For example, you could have the ASNMAIL user exit routine store the alerts in a problem management system.

## Specifying where to send monitor alerts

The Replication Alert Monitor can send alerts to the z/OS console, to an e-mail address or pager, or to both the console and an e-mail address or pager.

### Procedure

To specify where monitor alerts are sent, choose one of the following options:

To send to ...	Take these steps ...
E-mail only	<p>In the ASNCLP CREATE ALERT CONDITIONS FOR command, use the NOTIFY CONTACT or NOTIFY GROUP keywords to specify a person or group to notify by email.</p> <p>The following example shows the syntax for specifying that alerts are sent to the e-mail contact REPLADMIN:</p> <pre>CREATE CONDITIONS FOR QAPPLY MONITOR QUALIFIER MONQUAL NOTIFY CONTACT REPLADMIN (STATUS DOWN, ERRORS, WARNINGS, LATENCY 360, EXCEPTIONS)</pre>
<div style="background-color: #800000; color: white; padding: 2px; text-align: center; font-weight: bold;">z/OS</div> z/OS console only	<p>In the ASNCLP CREATE ALERT CONDITIONS FOR command, use the NOTIFY OPERATOR CONSOLE keywords to define the z/OS console as the destination for alerts. For example:</p> <pre>CREATE ALERT CONDITIONS FOR QCAPTURE SCHEMA ASN1 MONITOR QUALIFIER MONQUAL NOTIFY OPERATOR CONSOLE (STATUS DOWN, ERRORS, WARNINGS)</pre>
<div style="background-color: #800000; color: white; padding: 2px; text-align: center; font-weight: bold;">z/OS</div> E-mail and z/OS console	<ol style="list-style-type: none"> <li>Follow the steps above to specify an e-mail contact.</li> <li>Start the monitor by using the console=y option. Use one of the following methods:           <p><b>JCL</b> Specify CONSOLE=Y in the job that will start the monitor.</p> <p><b>asnmon command (USS)</b> From a USS command prompt, issue the <code>asnmon</code> command with the <b>console</b> parameter set to Y. For example:</p> <pre>asnmon monitor_server=SAMPLE monitor_qualifier=monqual console=y</pre> </li> </ol>

You can also use the Replication Center to specify that alerts be sent to both e-mail and the z/OS console. In the Alert Conditions window, check the **Send notification to the operator console** check box. Then use the Run Now or Save Command window to edit the operational command that will start the monitor so that the **console** parameter is set to Y, and start the monitor.

## The ASNMAIL exit routine for sending alerts in replication (Linux, UNIX, Windows)

### Linux UNIX Windows

The ASNMAIL exit routine distributes alerts that notify you about specific conditions in your replication environment.

z/OS

 The Replication Alert Monitor cannot use the ASNMAIL exit routine to handle notification on z/OS. An SMTP server can be used instead.

This exit routine takes the following input:

```
asnmail email_serverto_addresssubjectalert_messagealert_message
```



Tabella 21 describes the inputs for the ASNMAIL exit routine.

Tabella 21. Inputs for the ASNMAIL exit routine

Input	Description
<i>email_server</i>	This is the address of an e-mail server that uses the SMTP protocol. This server address is passed from the email_server parameter specified in at the start of the asnmon command.
<i>to_address</i>	This is the e-mail address of the contact to be notified.
<i>subject</i>	This is the subject in the notification.
<i>alert_message</i>	This is the string that contains the alert message.

Instead of sending alerts by e-mail, you can modify the ASNMAIL exit routine to put the alerts elsewhere such as in a problem management system. The `\sqlib\samples\repl\` directory contains a sample of the ASNMAIL exit routine. The `asnmail.smp` sample contains the input parameters and directions for using the sample program.

---

## Setting up the Replication Alert Monitor

Your replication environment consists of the replication programs that run on servers and the control tables that support them. The Replication Alert Monitor monitors this environment for you.

### Informazioni su questa attività

The following topics describe things to consider before setting up the monitor.

- “Authorization requirements for the Replication Alert Monitor” a pagina 224
- “Memory used by the Replication Alert Monitor”
- “Optional: Binding the Replication Alert Monitor program packages (Linux, UNIX, Windows)” a pagina 224

### Procedure

To set up the monitor:

1. Create control tables for each Monitor control server.
2. Define contact information for the monitor.
3. Create one or more monitors.
4. Select alert conditions.
5. Operate the monitor.
6. Optional: Define suspension periods for the monitor.

## Memory used by the Replication Alert Monitor

The Replication Alert Monitor uses memory to store definitions and to keep alerts in memory before they are sent as notifications.

The amount of memory needed for the definitions is directly proportional to the number of definitions. The Replication Alert Monitor reserves 32 KB of memory for storing alert notifications. More memory is requested, as needed, and released when no longer required.

**Recommendation:** Do not set a memory quota for the Replication Alert Monitor. If you need to set one, set it to 3 MB.

## Authorization requirements for the Replication Alert Monitor

All user IDs that run a Replication Alert Monitor must have authorization to access the Q Capture server or Q Apply server that you want to monitor. A user ID must also have access to the Monitor control tables on the Monitor control server.

User IDs that run a monitor must have the following authorities and privileges:

- SELECT, UPDATE, INSERT, and DELETE privileges for the Monitor control tables
- SELECT privileges on the Q Capture and Q Apply control tables on the servers that you want to monitor
- BINDADD authority (required only if you want to use the autobind feature for the monitor packages)
- EXECUTE privilege for the Monitor program packages
- WRITE privilege on the **monitor\_path** directory where the Replication Alert Monitor stores diagnostic files
- **Linux UNIX Windows** Read access to the password file that is used by the Replication Alert Monitor
- **Windows** Authority to create global objects

## Optional: Binding the Replication Alert Monitor program packages (Linux, UNIX, Windows)

**Linux UNIX Windows**

The Replication Alert Monitor program is bound automatically on Linux, UNIX, and Windows during execution. You can bind packages manually if you want to specify bind options, schedule binding, or check that all bind processes completed successfully.

### Procedure

To bind the Monitor program packages:

1. Change to the directory where the Monitor program bind files are located.

Platform	Location of bind files
<b>Windows</b>	<i>drive</i> :\...\sqllib\bnd  Where <i>drive</i> : is the drive where DB2 is installed.
<b>Linux UNIX</b>	<i>db2homedir</i> /sqllib/bnd  Where <i>db2homedir</i> is the DB2 instance home directory.

2. For each Monitor control server, do the following steps:
  - a. Connect to the database by entering the following command:  
`db2 connect to database`

Where *database* is the Monitor control server. If the database is cataloged as a remote database, you might need to specify a user ID and password on the db2 connect to command. For example:

`db2 connect to database user userid using password`

- b. Create and bind the Replication Alert Monitor program package to the database by entering the following commands:
 

```
db2 bind @asnmoncs.lst isolation cs blocking all grant public
db2 bind @asnmonur.lst isolation ur blocking all grant public
```

Where *cs* specifies the list in cursor stability format, and *ur* specifies the list in uncommitted read format.

These commands create packages, the names of which are in the files *asnmoncs.lst* and *asnmonur.lst*.

- 3. For each server that you are monitoring and to which the Replication Alert Monitor program connects, do the following steps:
  - a. Connect to the database by entering the following command:
 

```
db2 connect to database
```

Where *database* is the server that you want to monitor. If the database is cataloged as a remote database, you might need to specify a user ID and password on the *db2 connect to database* command. For example:
 

```
db2 connect to database user userid using password
```

- b. Create and bind the Replication Alert Monitor program package to the database by entering the following command:
 

```
db2 bind @asnmonit.lst isolation ur blocking all grant public
```

Where *ur* specifies the list in uncommitted read format.

These commands create packages, the names of which are in the file *asnmonit.lst*.

## Creating control tables for the Replication Alert Monitor

Before you can use the Replication Alert Monitor, you must create monitor control tables. These tables store alert conditions, contact information, run-time parameters, and other metadata for the monitor.

### Informazioni su questa attività

The server where you create the monitor control tables is called a Monitor control server.

The Monitor control server can be a DB2 for Linux, UNIX, Windows database or a DB2 for z/OS subsystem. In most cases you will need only one Monitor control server, but you can use multiple servers depending on your replication environment. For example, if you want monitors to run on the same system as the replication programs that they monitor, create one set of control tables for each local monitor on the server where the monitor runs.

### Procedure

To create monitor control tables, use one of the following methods:

Method	Description
ASNCLP command-line program	Use the CREATE CONTROL TABLES FOR command. For example: CREATE CONTROL TABLES FOR MONITOR CONTROL SERVER;

Method	Description
Replication Center	Use the Create Monitor Control Tables window. To open the window, right-click the <b>Monitor Control Servers</b> folder and select <b>Create Monitor Control Tables</b> .

## Defining contact information for the Replication Alert Monitor

You must define contact information for the individuals or groups that you want to notify of alert conditions before you use the Replication Alert Monitor for the first time.

### Informazioni su questa attività

Contact information is stored on Monitor control servers. Monitors running on the same Monitor control server can share contacts. If you have multiple Monitor control servers, you must define contacts for each server. You can change contact information after monitors are running.

After you define contacts by specifying the e-mail address for and the name of each contact, you can put contacts into groups. For example, you could set up a contact group called replication administrators that contains the contact information for all your replication administrators. You can also copy contact and group information from one server to another.

Contacts that you create for the Replication Alert Monitor in the Replication Center cannot be used in other DB2 centers such as the Task Center or the Health Center. Contacts created in other DB2 centers cannot be used by the Replication Alert Monitor.

### Procedure

To define contact information for the Replication Alert Monitor:

1. Create contacts and contact groups for the monitors on a Monitor control server by using one of the following methods:

Method	Description
ASNCLP command-line program	Use the CREATE CONTACT command. For example: CREATE CONTACT REPLADMIN EMAIL "repladmin@us.ibm.com" DESCRIPTION "replication administration";
Replication Center	Use the Create Contact window or Create Contact Group window. To open the windows, expand the Monitor control server for which you want to add a contact or contact group, right-click the <b>Contacts</b> folder and select <b>Create Contact</b> → <b>Person</b> or <b>Create Contact</b> → <b>Group</b> .

2. Optional: Copy contact information from one Monitor control server to another by using the Copy Contacts and Contact Groups window in the Replication Center. To open the window, expand the Monitor control server on which the contacts or contact groups are located. Select the **Contacts** folder. In the contents pane, right-click the contacts or contact groups that you want to copy, and select **Copy**.
3. Optional: Use the DELEGATE CONTACTS command in the ASNCLP command-line program to delegate an existing contact to a new contact for a specific period of time. For example:

## Creating monitors for replication or publishing

After you create monitor control tables, you can use the Create Monitor wizard in the Replication Center to create monitors and select the alert conditions that will be used to monitor your replication or publishing environment.

### Prima di iniziare

Before you create monitors, you must set up the Replication Alert Monitor.

### Procedure

To create a monitor:

1. In the Replication Center, open the Create Monitor wizard and specify the name of the monitor and the replication or publishing programs that the monitor will check for alert conditions:
  - a. To open the wizard, expand the Monitor control server on which you want to create a monitor, right-click the **Monitors** folder, and select **Create**.
  - b. On the Start page, specify a monitor qualifier. Then, specify the programs that you want this monitor to check for alert conditions. You can also monitor subscription sets that are used in SQL replication.

The wizard directs you to one or more of the following pages where you can select alert conditions, depending on which replication programs you want this monitor to check for alert conditions:

- Select alert conditions for Q Capture programs
- Select alert conditions for Q Apply programs
- Select alert conditions for Capture programs
- Select alert conditions for Apply programs
- Select alert conditions for subscription sets

See the online help for details. For example, if you specified that you want to monitor Q Capture programs and Q Apply programs, then the Create Monitor wizard directs you to the Select alert conditions for Q Capture programs page and the Select alert conditions for Q Apply programs page.

2. From one of the pages that are listed above, open secondary dialogs where you can:
  - a. Specify the programs or subscription sets that you want to monitor.
  - b. Specify the alert conditions that you want to check for, and the parameters for the appropriate alert conditions. For example, you can set the **monitor\_interval** parameter value to 60 to make the monitor check for alert conditions once every minute.
3. On the Summary page, click **Finish**.

## Selecting alert conditions for the Replication Alert Monitor

When you create a monitor, you select the alert conditions that will prompt that monitor to send alerts. You can select alert conditions for each Q Capture program, Q Apply program, Capture program, Apply program, or subscription set that a monitor is monitoring.

### Informazioni su questa attività

The Replication Alert Monitor monitors the activity of the replication and publishing programs at the following times:

- Each monitor checks for alert conditions immediately when you start it.
- Each monitor checks for alert conditions periodically, at timed intervals that you specify.

### Procedure

To select alert conditions for the Replication Alert Monitor, use one of the following methods:

Method	Description
<b>ASNCLP command-line program</b>	Use one of the following commands: <ul style="list-style-type: none"> <li>• CREATE ALERT CONDITIONS FOR APPLY</li> <li>• CREATE ALERT CONDITIONS FOR CAPTURE</li> <li>• CREATE ALERT CONDITIONS FOR Q CAPTURE</li> <li>• CREATE ALERT CONDITIONS FOR Q APPLY</li> </ul>
<b>Replication Center</b>	Use one or more of the following pages in the Create Monitor wizard in the Replication Center, depending on which program you chose to monitor: <ul style="list-style-type: none"> <li>• Select alert conditions for Q Capture programs</li> <li>• Select alert conditions for Q Apply programs</li> <li>• Select alert conditions for Capture programs</li> <li>• Select alert conditions for Apply programs</li> <li>• Select alert conditions for subscription sets</li> </ul>

Specify thresholds that are compatible with your environment. For example, if a Capture program is running with a commit interval of 30 seconds, specify a threshold for Capture latency that is greater than 30 seconds. Or, if you schedule an Apply program to process subscription sets every 10 minutes, set the threshold of the APPLY\_SUBSDELAY alert condition to a value that is greater than 10 minutes.

## Changing alert conditions for the Replication Alert Monitor

You can change alert conditions while the monitor is running. You do this by changing the alert conditions and then reinitializing the monitor.

### Procedure

To change alert conditions, use one of the following methods:

Method	Description
<b>ASNCLP command-line program</b>	Use one of the following commands: <ul style="list-style-type: none"> <li>• ALTER ALERT CONDITIONS FOR APPLY</li> <li>• ALTER ALERT CONDITIONS FOR CAPTURE</li> <li>• ALTER ALERT CONDITIONS FOR Q CAPTURE</li> <li>• ALTER ALERT CONDITIONS FOR Q APPLY</li> </ul>
<b>Replication Center</b>	Use the Alert Conditions window for a Q Capture program, Q Apply program, Capture program, Apply program, or subscription set. To open the windows, select a monitor in the object tree, right-click a schema or subscription set in the contents pane, and click <b>Change</b> .

After you change the alert conditions, reinitialize the monitor.

## Defining suspension periods for the Alert Monitor

You can define suspension periods for a Replication Alert Monitor program. You can create a repeating suspension (for example, every Sunday morning for two hours) or suspend the monitor for a single time period.

### Informazioni su questa attività

While the monitor is suspended, it will stop checking Q Capture, Q Apply, Capture control, or Apply control servers for all defined alert conditions. When the suspension period ends, the monitor will resume checking.

To define a suspension that repeats, you create a *suspension template*. If you create a template, you can reuse the template on multiple monitored servers.

If you do not create a template, you can specify a start date and time and end date and time for which monitoring on a server is suspended one time.

All dates and times for monitor suspensions are based on the clock at the system where the monitor is running.

### Restrizioni

Suspensions and suspension templates can only be defined through the ASNCLP command-line program and cannot be defined or viewed through the Replication Center.

### Procedure

To suspend the monitor for a defined period:

1. Optional: Use the CREATE MONITOR SUSPENSION TEMPLATE command in the ASNCLP command-line program to create a template to define a repeating suspension.

For example, the following command creates a template that suspends the monitor program from 00:00:00 to 04:00:00 every Sunday:

```
CREATE MONITOR SUSPENSION TEMPLATE SUNDAY START TIME 00:00:00  
REPEATS WEEKLY DAY OF WEEK SUNDAY FOR DURATION 4 HOURS
```

2. Use the CREATE MONITOR SUSPENSION command in the ASNCLP command-line program to define a start and end point for a one-time suspension or use a suspension template.

For example, the following command creates a suspension called S1, which uses the template SUNDAY to suspend the monitor control server QSRVR1:

```
CREATE MONITOR SUSPENSION NAME S1 FOR SERVER QSRVR1 STARTING DATE 2006-12-10  
USING TEMPLATE SUNDAY ENDING DATE 2007-12-31
```

3. Reinitialize the monitor that you want to suspend by using the `asnmcmd reinit` command.

You can also use the Reinitialize Monitor window in the Replication Center. To open the window, right-click a monitor qualifier in the contents pane and click **Reinitialize Monitor**.

4. Optional: Use one of the following commands in the ASNCLP command-line program to list, change, or drop monitor suspensions or suspension templates:

Command	Description
LIST MONITOR SUSPENSION	Generates a list of suspensions on a monitor control server.
ALTER MONITOR SUSPENSION	Allows you to change the following properties of a suspension: <ul style="list-style-type: none"> <li>• The template that is used</li> <li>• The start or end date for using a template</li> <li>• The start or end date for suspending the monitor program one time</li> </ul>
DROP MONITOR SUSPENSION	Deletes a suspension from the monitor control tables.
LIST MONITOR SUSPENSION TEMPLATE	Generates a list of suspension templates on a monitor control server.
ALTER MONITOR SUSPENSION TEMPLATE	Allows you to change the frequency and length of monitor suspensions as defined in a suspension template.
DROP MONITOR SUSPENSION TEMPLATE	Deletes a suspension template from the monitor control tables.

## Operating the Replication Alert Monitor

You can start, stop, suspend, reinitialize, and perform other operations on the Replication Alert Monitor.

### Starting monitors



You can use several methods to start a monitor. You can also decide whether to run the monitor continuously or for only one monitor cycle. You can also set values for parameters, and enter the e-mail address of the person to contact if the monitor itself encounters an error while running.

#### Prima di iniziare

- Create monitor control tables and a monitor, which includes selecting contacts and alert conditions.
- Create a password file.
- Make sure that you have the correct authorization to access the Monitor control tables and servers where the programs that you want to monitor are running.

#### Procedure

To start a monitor, use one of the following methods:

Method	Description
Replication Center	Use the Start Monitor window. To open the window, right-click the Monitor qualifier that identifies the monitor that you want to start, and select <b>Start Monitor</b> .
  asnmon system command	Use this command to start a monitor and optionally specify startup parameters.



Method	Description
<p><b>z/OS</b></p> <p>Automatic Restart Manager</p>	You can set up the ARM recovery system to start a monitor from the z/OS console or TSO.
<p><b>Windows</b></p> <p>Windows service</p>	You can set up the monitor to run as a Windows service.

## Reinitializing monitors

You can reinitialize a monitor while it is running. Reinitializing a monitor causes it to recognize any updates that you have made to contacts, alert conditions, and parameter values. For example, reinitialize a monitor if you added a new e-mail address for a contact while the monitor is running.

### Procedure

To reinitialize a monitor, use one of the following methods.

Method	Description
<b>asnmcmd system command</b>	Use the asnmcmd reinit command to reinitialize a running monitor.
<b>Replication Center</b>	Use the Reinitialize Monitor window to reinitialize a monitor. To open the window, right-click the Monitor qualifier that identifies the monitor that you want to reinitialize, and select <b>Reinitialize Monitor</b> .

## Suspending and resuming a monitor

You can suspend and resume a monitor when you want to temporarily stop monitoring your replication or publishing environment.

### Informazioni su questa attività

You might consider suspending and resuming a monitor instead of stopping and restarting it in the following situations:

- You do not have the authority to stop and start a monitor.
- A server in your replication environment is being serviced. For example, if a monitor named MONITOR1 is monitoring SERVER\_GREEN, which is a Q capture server, and SERVER\_GREEN will be shut down for maintenance between 4 and 7 p.m., you could suspend MONITOR1 at 4 p.m. and resume it at 7 p.m. This prevents MONITOR1 from issuing a QCAPTURE\_STATUS alert condition.

If you suspend the monitor while the Capture, Apply, Q Capture, or Q Apply programs are running, the monitor continues where it left off when you resume it. When a monitor is suspended and then resumed, it will not check for alert conditions or issue alerts for conditions that were met while the monitor was suspended.

### Procedure

To suspend and resume a monitor:

1. Suspend the monitor by issuing the `asnmcmd suspend` command. The monitor stops checking for alert conditions.
2. Resume the monitor by issuing the `asnmcmd resume` command. The monitor resumes checking for alert conditions.

## Ending a monitor suspension

You can end a monitor suspension before its regularly scheduled expiration time by removing the suspension from the monitor control tables and then reinitializing the monitor.

### Procedure

To end a monitor suspension:

1. Use the `DROP MONITOR SUSPENSION` command in the `ASNCLP` command-line program to remove the suspension from the control tables at the monitor server.

For example, the following command removes a suspension named `SUSP1`:

```
DROP MONITOR SUSPENSION NAME SUSP1
```

2. Reinitialize the monitor by using one of the following methods:

Method	Description
<b>asnmcmd command</b>	Use the <code>asnmcmd reinit</code> command to prompt the monitor to read its control tables for the most recent changes. The following command reinitializes the monitor identified by the monitor qualifier <code>myqual</code> at the Monitor control server <code>wsdb</code> :  <code>asnmcmd monitor_server=wsdb monitor_qual=myqual reinit</code>
<b>Replication Center</b>	Use the Reinitialize Monitor window. In the contents pane, right-click the monitor qualifier that identifies the monitor that you want to reinitialize and click <b>Reinitialize Monitor</b> .

**Nota:** You can also stop and then start the monitor to prompt it to read its control tables.

## Stopping monitors

When you stop a monitor, it stops checking the replication or publishing programs for alert conditions. You can use the Replication Center, a system command, or a DB2 replication service to stop a monitor.

### Informazioni su questa attività

If the monitor stops while the Capture, Apply, Q Capture, or Q Apply programs are running, then the next time the monitor starts it performs the following actions:

- Checks for alert conditions that were met while the monitor was stopped.
- Issues alerts for any conditions that were met.

### Procedure

To stop a monitor, use one of the following methods:

Method	Description
<b>asnmcmd stop command</b>	Use this command to stop a monitor.

Method	Description
Replication Center	Use the Stop Monitor window to stop a monitor. To open the window, right-click the Monitor qualifier that identifies the monitor that you want to stop, and select <b>Stop Monitor</b> .
Windows services	Stop the replication service. The monitor stops automatically.

## Reviewing Monitor program messages

Use the Monitor Messages window to review the messages that were inserted in the IBMSNAP\_MONTRACE table over a specified period of time. The IBMSNAP\_MONTRACE table contains rows for significant events such as actions, warnings, and errors that are issued by the Monitor program.

For example, from the Monitor Messages window, you can review all the error and warning messages that are recorded by the Monitor program during one week. You can also print or save data to a file from the Monitor Messages window.

---

## Parameters of the Replication Alert Monitor

You can determine the behavior of the Replication Alert Monitor by setting values for various parameters.

### Default values of Replication Alert Monitor parameters

When you use the replication administration tools to create Monitor control tables, default values are set for the monitor operating parameters.

Tabella 22 shows the default value for each parameter.

*Tabella 22. Default values for Replication Alert Monitor operating parameters*

Operational parameter	Default value
alert_prune_limit	10080 minutes
autoprune	Y
email_server	no default value
max_notification_minutes	60 minutes
max_notifications_per_alert	3
monitor_errors	no default value
monitor_interval	300 seconds
monitor_limit	10080 minutes
monitor_path	the directory where the asnmon command was invoked.
runonce	N
trace_limit	10080 minutes

## Descriptions of the Replication Alert Monitor parameters

This topic describes the following parameters you can use to operate the Replication Alert Monitor:

- “alert\_prune\_limit”
- “autoprune”
- “email\_server”
- “max\_notification\_minutes”
- “max\_notifications\_per\_alert” a pagina 235
- “monitor\_errors” a pagina 235
- “monitor\_interval” a pagina 235
- “monitor\_limit” a pagina 235
- “monitor\_path” a pagina 235
- “runonce” a pagina 235
- “trace\_limit” a pagina 236

## **alert\_prune\_limit**

**Default:** alert\_prune\_limit=10080 minutes (seven days.)

When the Replication Alert Monitor starts a new monitor cycle, it prunes the rows from the IBMSNAP\_ALERTS table that are eligible for pruning. By default, the Replication Alert Monitor prunes the rows that are older than 10080 minutes (seven days). The **alert\_prune\_limit** parameter controls how much old data the Replication Alert Monitor stores in the table. The parameter specifies how old the data must be before the Replication Alert Monitor prunes it.

You can reduce the value of **alert\_prune\_limit** parameter if the storage space on your system is small for the IBMSNAP\_ALERTS table. A lower prune limit saves space, but increases processing costs. Alternatively, you might want to increase the value for the **alert\_prune\_limit** parameter to maintain a history of all the alert activity. In SQL replication only, a higher prune limit requires more space for change-data (CD) tables and UOW tables, but decreases processing costs.

## **autoprune**

**Default:** autoprune=y

The **autoprune** parameter controls automatic pruning. The Replication Alert Monitor automatically prunes rows from the IBMSNAP\_ALERTS table that it has already copied into the Monitor control tables.

## **email\_server**

The **email\_server** parameter enables the ASNMAIL exit routine. The default ASNMAIL routine enables the Replication Alert Monitor to send alerts by using e-mail. Set the value of this parameter to the address of an e-mail server that is set to use the Simple Mail Transfer Protocol (SMTP).

## **max\_notification\_minutes**

**Default:** max\_notifications\_minutes=60

The **max\_notifications\_minutes** parameter specifies how long that a monitor will track an alert condition to see if it occurs more than once. By default, if an alert condition occurs more than once within 60 minutes, the Replication Alert Monitor will send a maximum of three alerts during the 60 minute period. The

**max\_notifications\_per\_alert** parameter tells the Monitor how many notifications to send during the period of time specified by the **max\_notifications\_minutes** parameter for any alert condition.

### **max\_notifications\_per\_alert**

**Default:** **max\_notifications\_per\_alert=3**

The **max\_notifications\_per\_alert** parameter tells the Replication Alert Monitor the maximum number of notifications to send for any one alert. By default, if the Replication Alert Monitor receives an alert condition more than once, it sends a maximum of three notifications for that alert condition in a period of 60 minutes.

### **monitor\_errors**

The Replication Alert monitor stores any errors that occur in the monitoring process. One example of an operational error is when the Replication Alert Monitor cannot connect to the Monitor control server. You must specify an e-mail address for the **monitor\_errors** parameter if you want to receive notification of operational errors. If you do not specify an e-mail address, the Replication Alert Monitor logs operational errors, but it does not send notification of the errors.

The Replication Alert Monitor ignores the **monitor\_errors** parameter if the **email\_server** parameter does not describe a valid e-mail server.

### **monitor\_interval**

**Default:** **monitor\_interval=300** seconds (5 minutes)

The **monitor\_interval** parameter tells the Replication Alert Monitor how often to check for alert conditions. By default, the Replication Alert Monitor checks for all alert conditions for the specific monitor on the server every 300 seconds.

### **monitor\_limit**

**Default:** **monitor\_limit=10080** minutes (7 days)

For Q replication, the **monitor\_limit** parameter specifies how long to keep rows in the **IBMQREP\_CAPMON** and the **IBMQREP\_CAPQMON** tables before the Q Capture program can prune them. For SQL replication, the **monitor\_limit** parameter specifies how long to keeps rows in the **IBMSNAP\_CAPMON** table before the Q Capture program can prune them. At each pruning interval, the Capture and Q Capture programs prune rows in these tables if the rows are older than this limit based on the current timestamp.

### **monitor\_path**

**Default:** **monitor\_path**=the directory where the **asnmon** command was invoked

The **monitor\_path** parameter specifies the location of the log files that the Replication Alert Monitor uses.

### **runonce**

**Default:** **runonce=n**

When you start the Replication Alert Monitor, by default, it runs at intervals to monitor any alert conditions that you selected. You can schedule the Replication Alert Monitor to run hourly, at some other time interval, or even just one time.

When you specify **runonce=y**, the Replication Alert Monitor checks one time for all the alert conditions that you selected and ignores the **monitor\_interval** parameter. You can use **runonce** when you run the Replication Alert Monitor in a batch process. For example, after the Apply program completes, you can use **runonce=y** to determine if any subscription sets failed. Then, if a subscription set did fail, the Replication Alert Monitor sends notification to your contact person or group.

By default, the **monitor\_interval** is 300 seconds (five minutes). The Replication Alert Monitor checks for all the alert conditions for each monitor on the server every 300 seconds. If the Replication Alert Monitor finds an alert condition, it sends notification.

### **trace\_limit**

**Default:** **trace\_limit=10080** minutes (7 days)

The **trace\_limit** parameter tells the Replication Alert Monitor how often to prune the IBMSNAP\_MONTRACE and IBMSNAP\_MONTRAIL tables. The Replication Alert Monitor stores the rows in these tables for 10080 minutes (seven days). The Replication Alert Monitor prunes any rows older than the value that you specify for the **trace\_limit** parameter.

## **Changing runtime parameters for the Replication Alert Monitor**

You can change runtime parameters for the Replication Alert Monitor when you start the monitor or while the monitor is running.

### **Informazioni su questa attività**

You set initial parameter values when you create a monitor. These values are stored in the IBMSNAP\_MONPARMS control table. When you start the monitor, it reads this control table and uses the parameter values.

You can override the saved values at runtime when you start the monitor or while the monitor is running. Any runtime values that you set will only last for the current run. If the monitor is stopped and restarted, it uses the values saved in the control table.

### **Procedure**

1. Change parameters when you start the monitor. Use one of the following methods.

<b>Method</b>	<b>Description</b>
<b>asnmon system command</b>	Specify one or more parameters and values when you use this command to start the monitor.
<b>Replication Center</b>	Use the Specify Monitor Startup Parameters window. To open the window, right-click a Monitor qualifier in the contents pane that identifies the monitor that you want to start, and click <b>Start Monitor</b> . Then click <b>Parameters</b> on the Start Monitor window.

2. Use the `asnmcmd chgparms` system command to change parameters while the monitor is running. You can change the following parameters:

- **monitor\_interval**
- **autoprun**
- **alert\_prune\_limit**
- **trace\_limit**
- **max\_notifications\_per\_alert**
- **max\_notifications\_minutes**

## Specifying how often the Replication Alert Monitor runs

You must decide how often the Replication Alert Monitor will check for alert conditions for your replication environment.

### Procedure

To specify how often the Replication Alert Monitor runs, use the following methods:

- Use the **runonce** parameter of the `asnmon` command to specify if the Replication Alert Monitor will run repeatedly or only once.
- Use the **monitor\_interval** parameter of the `asnmon` command to specify how often the Replication Alert Monitor will run when **runonce=n**.
- Use the Replication Center to specify run times when you start the Replication Alert Monitor.

## Specifying notification criteria for selected alert conditions

The Replication Alert Monitor stores any alert conditions that you select. You can set up notification parameters to notify a contact of the alert conditions automatically with electronic mail (e-mail).

### Procedure

To specify notification criteria for alert conditions, use the following methods:

- Set the **max\_notifications\_per\_alert** parameter to control the maximum notification for a particular time period. Specify the maximum number of notifications you want to receive about a particular alert condition within the time period specified by the **max\_notifications\_minutes** parameter.
- Set the **email\_server** parameter to enable DB2 to notify you by e-mail when an alert condition occurs. Set the value of this parameter to the address of an e-mail server by using the SMTP protocol.
- Optional: Write your own extensions to the `ASNMAIL` exit routine to customize how alert conditions are handled. This option is useful for integrating with problem management and other systems.

## Specifying notification criteria for operational errors

The Replication Alert Monitor sends notification if it causes an error during its operation.

### Procedure

To specify notification criteria for operational errors, set the value of the **monitor\_errors** parameter to an e-mail address. The monitor will send notification of operational errors that it causes to this address. Enter the e-mail address by using the Simple Mail Transfer Protocol (SMTP) protocol.

## Specifying prune intervals for data from the Replication Alert Monitor

The Replication Alert Monitor can automatically prune your Monitor control tables. You must decide whether the Monitor will prune the tables automatically, and if so, how the Monitor will prune the tables.

### Procedure

To specify how often to prune your monitor tables, use the following methods:

- Specify whether you want the Replication Alert Monitor to automatically prune its control tables by using the **autoprune** parameter.
- Change the value for the **alert\_prune\_limit** parameter to control how much historic data you want the Replication Alert Monitor to store in the table. Specify how old the data must be before the Replication Alert Monitor prunes it from the IBMSNAP\_ALERTS table.
- Change the value for the **trace\_limit** parameter to control how long the Replication Alert Monitor stores rows in your monitor tables.



---

## Capitolo 16. Replication services (Windows)

### Windows

You can run the replication programs as a system service on the Windows operating system by using the Windows Service Control Manager (SCM).

---

### Description of Windows services for replication

#### Windows

On the Windows operating system, a replication service is a program that starts and stops the Q Capture, Q Apply, Capture, Apply, or Replication Alert Monitor programs.

When you create a replication service, it is added to the SCM in Automatic mode and the service is started. Windows registers the service under a unique service name and display name.

The following terms describe naming rules for replication services:

#### Replication service name

The replication service name uniquely identifies each service and is used to stop and start a service. It has the following format:

*DB2.instance.alias.program.qualifier\_or\_schema*

Tabella 23 describes the inputs for the replication service name.

Tabella 23. Inputs for the replication service name

Input	Description
<i>instance</i>	The name of the DB2 instance.
<i>alias</i>	The database alias of the Q Capture server, Q Apply server, Capture control server, Apply control server, or Monitor control server.
<i>program</i>	One of the following values: QCAP (for Q Capture program), QAPP (for Q Apply program), CAP (for Capture program), APP (for Apply program), or MON (for Replication Alert Monitor program).
<i>qualifier_or_schema</i>	One of the following identifiers: Q Capture schema, Q Apply schema, Capture schema, Apply qualifier, or Monitor qualifier.

**Example:** The following service name is for a Q Apply program that has the schema ASN and is working with database DB1 under the instance called INST1:

*DB2.INST1.DB1.QAPP.ASN*

#### Display name for the replication service

The display name is a text string that you see in the Services window and it is a more readable form of the service name. For example:

If you want to add a description for the service, use the Service Control Manager (SCM) after you create a replication service. You can also use the SCM to specify a user name and a password for a service.

---

## Creating a replication service

### Windows

You can create a replication service to start a Q Capture program, Q Apply program, Capture program, Apply program, and the Replication Alert Monitor program on Windows operating systems.

#### Prima di iniziare

Before you create a replication service, make sure that the DB2 instance service is running. If the DB2 instance service is not running when you create the replication service, the replication service is created but it is not started automatically.

#### Informazioni su questa attività

When you create a service, you must specify the account name that you use to log on to Windows and the password for that account name.

You can add more than one replication service to your system. You can add one service for each schema on every Q Capture, Q Apply, or Capture control server, and for each qualifier on every Apply control server and Monitor control server, respectively. For example, if you have five databases and each database is an Q Apply control server and a Monitor control server, you can create ten replication services. If you have multiple schemas or qualifiers on each server, you could create more services.

#### Procedure

To create a replication service:

Use the `asnsrct` command.

When you create a service, you must specify the account name that you use to log on to Windows and the password for that account name.

**Tip:** If your replication service is set up correctly, the service name is sent to stdout after the service is started successfully. If the service does not start, check the log files for the program that you were trying to start. By default, the log files are in the directory specified by the `DB2PATH` environment variable. You can override this default by specifying the path parameter (`capture_path,apply_path,monitor_path`) for the program that is started as a service. Also, you can use the Windows Service Control Manager (SCM) to view the status of the service.

---

## Starting a replication service

Windows

After you create a replication service, you can stop it and start it again.

### Informazioni su questa attività

**Important:** If you started a replication program from a service, you will get an error if you try to start the program by using the same schema or qualifier.

### Procedure

To start a replication service, use one of the following methods.

- The Windows Service Control Manager (SCM)
- net stop command

---

## Stopping a replication service

Windows

After you create a replication service, you can stop it and start it again.

### Informazioni su questa attività

When you stop a replication service, the program associated with that service stops automatically. However, if you stop a program by using a replication system command (asnlqcmd, asnqcmd, asncmd, asnacmd, or asnmcmd), the service that started the program continues to run. You must stop it explicitly.

### Procedure

To stop a replication service, use one of the following methods.

- The Windows Service Control Manager (SCM)
- net stop command

---

## Viewing a list of replication services

Windows

You can view a list of all your replication services and their properties by using the asnlist command.

### Procedure

To view a list of replication services, use the asnlist command. You can optionally use the **details** parameter to view a list of replication services and descriptions of each service.

---

## Dropping a replication service

Windows

If you no longer need a replication service you can drop it so that it is removed from the Windows Service Control Manager (SCM).

#### **Informazioni su questa attività**

If you want to change the start-up parameters for a program that is started by a service, you must drop the service and then create a new one using new start-up parameters.

#### **Procedure**

To drop a service for replication commands, use the `asnsdrop` command.

---

## Capitolo 17. Pianificazione di programmi di replica SQL su vari sistemi operativi

Si potrebbe voler pianificare l'avvio del programma Capture, del programma Apply oppure del programma Replication Alert Monitor in orari stabiliti utilizzando i comandi disponibili sul sistema operativo in uso.

---

### Pianificazione di programmi sui sistemi operativi Linux e UNIX

#### Linux UNIX

È possibile pianificare quando avviare i programmi di replica sul sistema operativo Linux e UNIX.

#### Procedure

Per pianificare programmi di replica su Linux e UNIX

Utilizzare il comando **at** per avviare un programma di replica in un determinato orario. La Tabella 24 mostra i comandi utilizzati per avviare i programmi di replica alle 3:00 del pomeriggio di venerdì:

Tabella 24. Comandi di pianificazione per i programmi di replica (Linux, UNIX)

Programma di replica	Comando Linux o UNIX
Capture	at 3pm Friday asncap autoprune=n
Apply	at 3pm Friday asnapply applyqual=myqual
Controllo degli avvisi di replica	at 3pm Friday asnmon monitor_server=db2srv1 monitor_qualifier=mymon

---

### Pianificazione di programmi sui sistemi operativi Windows

#### Windows

È possibile pianificare quando avviare i programmi di replica sul sistema operativo Windows.

#### Procedure

Se non si sta utilizzando Service Control Manager di Windows, utilizzare il comando AT per avviare i programmi ad un determinato orario. Prima di immettere il comando AT, avviare il servizio Schedule Service di Windows. La Tabella 25 illustra i comandi utilizzati per avviare i programmi di replica alle ore 15:00 di Venerdì:

Tabella 25. Comandi di pianificazione per i programmi di replica (Windows)

Programma di replica	Comando Windows
Capture	c:\>at 15:00/interactive"c:\SQLLIB\BIN\db2cmd.exe c:\CAPTURE\asncap.exe"

Tabella 25. Comandi di pianificazione per i programmi di replica (Windows) (Continua)

Programma di replica	Comando Windows
Apply	c:\>AT 15:00 /interactive "c:\SQLLIB\BIN\db2cmd.exe c:\SQLLIB\BIN\asnapply.exe control_server=cntldb apply_qual=qualid1"
Controllo degli avvisi di replica	c:\>AT 15:00 /interactive "c:\SQLLIB\BIN\db2cmd.exe c:\CAPTURE\asnmon.exe monitor_server=db2srv1 monitor_qualifier=mymon"

## Pianificazione dei programmi sui sistemi operativi z/OS

### z/OS

È possibile pianificare quando avviare i programmi di replica sul sistema operativo z/OS utilizzando due comandi differenti.

#### Procedure

Per la pianificazione di programmi sul sistema operativo z/OS, utilizzare il seguente metodo:

1. Creare una procedura che richiama il programma per z/OS nella PROCLIB.
2. Modificare il modulo RACF ICHRIN03 (o le definizioni appropriate relative al proprio pacchetto di sicurezza MVS), per associare la procedura a un ID utente.
3. Eseguire il link e modificare il modulo in SYS1.LPALIB.
4. Utilizzare il comando \$TA JES2 o il comando AT NetView per l'avvio del programma Capture o Apply e Apply ad un'ora specifica. Consultare *MVS/ESA JES2 Commands* per ulteriori informazioni sull'utilizzo del comando \$TA JES2. Consultare *NetView per MVS Command Reference* per ulteriori informazioni sull'utilizzo del comando AT NetView.

## Scheduling programs on the System i operating system

### System i

You can schedule when to start the replication programs on the System i operating system.

#### Procedure

1. If you want to start the Apply program, issue the ADDJOBSCDE command.
2. If you want to start the Capture program, issue the SBMJOB command. For example:  

```
SBMJOB CMD('STRDPRCAP...')SCDDATE(...)SCDTIME(...)
```

---

## Capitolo 18. Modalità di comunicazione dei componenti della replica SQL

I vari componenti della replica vengono eseguiti indipendentemente uno dall'altro, ma dipendono uno dall'altro per le informazioni che ciascuno memorizza nelle tabelle di controllo della replica per comunicare tra di loro.

### Strumenti di gestione

Il Centro di replica o il programma della riga comandi ASNCLP crea script SQL che inseriscono le informazioni iniziali su origini registrate, serie di richieste e condizioni di segnalazione nelle tabelle di controllo.

### Programma Capture o trigger

Il programma Capture e i trigger Capture aggiornano le tabelle di controllo per indicare l'avanzamento della replica e per coordinare l'elaborazione delle modifiche.

### Programma Apply

Il programma Apply aggiorna le tabelle di controllo per indicare l'avanzamento della replica e per coordinare l'elaborazione delle modifiche.

### Replication Alert Monitor

Replication Alert Monitor legge le tabelle di controllo che sono state aggiornate dal programma Capture, programma Apply e i trigger Capture per comprendere eventuali problemi e l'avanzamento su un server.

---

## Centro di replica, ASNCLP, trigger o programma Capture e programma Apply

Quando si registra una tabella, vista o nickname come un'origine di replica, il Centro di replica o il programma della riga comandi ASNCLP crea uno script SQL che memorizza le informazioni per tale origine nella tabella di controllo della replica che contiene tutte le informazioni di registrazione, la tabella IBMSNAP\_REGISTER. Anche lo script SQL generato dagli strumenti di gestione crea le tabelle CD per le origini registrate.

La tabella IBMSNAP\_REGISTER contiene una riga per ogni tabella di origine registrata e una riga per ogni tabella sottostante in una vista registrata. Questa tabella contiene i seguenti tipi di informazione su ogni origine registrata:

- Il nome dello schema e il nome della tabella di origine
- Il tipo di struttura di ciascuna tabella di origine registrata
- Il nome dello schema e il nome della tabella CD
- I nomi delle tabelle CD per le tabelle sottostanti in questa vista (solo per viste registrate e solo se le tabelle sottostanti sono registrate)
- Il nome dello schema e il nome della tabella CCD interna (dove esistente)
- Il livello di rilevamento del conflitto per origini di aggiornamento

I programmi Capture e Apply utilizzano le informazioni nella tabella IBMSNAP\_REGISTER per comunicarsi il relativo stato. Questa tabella ha diverse colonne in più per le informazioni correlate.

Per le origini System i, comprese le tabelle che sono registrate in remoto, è presente anche un'estensione alla tabella IBMSNAP\_REGISTER, IBMSNAP\_REG\_EXT, che contiene altre informazioni che sono univoche per i System i, ad esempio, la libreria di giornale e il nome di giornale.

Quando si crea una serie di richieste e vi si aggiungono dei membri, il Centro di replica crea uno script SQL che memorizza le informazioni per tale serie di richieste nelle tabelle di controllo della replica che contengono tutte le informazioni della serie di richieste, come riportato di seguito:

- Tabella IBMSNAP\_SUBS\_SET
- Tabella IBMSNAP\_SUBS\_MEMBR
- tabella IBMSNAP\_SUBS\_COLS
- tabella IBMSNAP\_SUBS\_STMTS

Se le tabelle di destinazione non esistono già, vengono create dallo script SQL generato dal Centro di replica.

La tabella della serie di richieste principale, IBMSNAP\_SUBS\_SET, contiene una riga per ogni serie di richieste. Questa tabella contiene i seguenti tipi di informazione su ogni serie di richieste:

- Il qualificatore Apply
- Il nome della serie di richieste
- Il tipo di serie di richieste: di sola lettura o lettura/scrittura (aggiornamento)
- I nomi e gli alias dei database di origine e destinazione
- Il tempo di elaborazione della serie di richieste
- Lo stato attuale della serie di richieste

Anche questa tabella ha diverse colonne in più per le informazioni correlate.

Le altre tabelle di serie di richieste contengono informazioni sui membri della serie di richieste, colonne e istruzioni SQL (o procedure memorizzate) elaborate con la serie.

---

## Il programma Capture e il programma Apply

Il programma Capture utilizza alcune delle tabelle di controllo della replica per indicare le modifiche apportate al database di origine e il programma Apply utilizza i valori di tali tabelle di controllo per individuare ciò che deve essere copiato nel database di destinazione.

Il programma Capture non cattura alcuna informazione finché ciò non gli viene segnalato dal programma Apply e il programma Apply non segnala al programma Capture di avviare la cattura delle modifiche finché non si definiscono un'origine della replica e le serie di richieste associate.

Di seguito viene descritto come i programmi Apply e Capture comunicano in un tipico scenario di replica per assicurare l'integrità dei dati:

### Cattura dei dati da un database di origine

1. Il programma Capture legge la tabella IBMSNAP\_REGISTER durante l'avvio per identificare le origini di replica registrate per cui deve catturare le modifiche. Una volta eseguito ciò, conserva in memoria le relative informazioni di registrazione.



2. Il programma Capture legge continuamente il giornale o la registrazione DB2 per individuare la modifica dei record (INSERT, UPDATE e DELETE) per le tabelle di origine registrate o le viste. Rileva inoltre gli inserimenti nella tabella IBMSNAP\_SIGNAL al fine di cogliere azioni di segnali che sono state inizializzate dal programma Apply o da un utente. Quando il programma Apply inserisce un segnale CAPSTART nella tabella IBMSNAP\_SIGNAL e il programma Capture rileva il segnale di cui è stato eseguito il commit, il programma Capture inizializza la registrazione e avvia la cattura delle modifiche per l'origine associata.
3. Una volta che il programma Capture ha avviato la cattura delle modifiche per un'origine registrata, il programma scrive una riga (o due se è stato specificato che gli aggiornamenti devono essere salvati come istruzioni DELETE e INSERT) sulla tabella CD per ogni modifica di cui è stato eseguito il commit che rileva nel giornale o nella registrazione DB2. Il programma Capture mantiene in memoria le modifiche di cui non è stato eseguito il commit finché non viene eseguita tale azione o finché non vengono annullate. Ogni origine di replica registrata che non è una tabella CCD esterna ha una tabella CD associata.
4. Ad ogni intervallo di commit, il programma Capture esegue il commit dei dati che sono stati scritti sulle tabelle CD e UOW e inoltre aggiorna la tabella IBMSNAP\_REGISTER ad indicare quali tabelle CD hanno modifiche di cui è stato eseguito il commit.

#### **Applicazione dei dati ad un database di destinazione**

1. Per tutte le serie di richieste definite di nuovo, il programma Apply innanzitutto segnala al programma Capture di avviare la cattura delle modifiche. Quindi, viene eseguito un aggiornamento completo per ciascun membro della serie (a meno che non è una tabella di destinazione completa).
2. Quando una serie di richieste è idonea per la replica, il programma Apply verifica la tabella IBMSNAP\_REGISTER per determinare se sono presenti modifiche di cui deve essere eseguita la replica.
3. Il programma Apply copia le eventuali modifiche dalla tabella CD alla tabella di destinazione.
4. Il programma Apply aggiorna la tabella IBMSNAP\_SUBS\_SET per registrare i dati che il programma Apply ha copiato per ogni serie di richieste.
5. Il programma Apply aggiorna la tabella IBMSNAP\_PRUNE\_SET con un valore che indica il punto su cui ha letto le modifiche dalla tabella CD.

#### **Eliminazione delle tabelle CD**

Quando il programma Capture elimina le tabelle CD, utilizza le informazioni ubicate nella tabella IBMSNAP\_PRUNE\_SET per determinare le modifiche applicate ed elimina le modifiche di cui è già stata eseguita la replica dalla tabella CD.

---

## **Trigger Capture e programma Apply**

I trigger Capture utilizzano alcune delle tabelle di controllo della replica per indicare le modifiche apportate al database di origine e il programma Apply utilizza i valori di tali tabelle di controllo per individuare ciò che deve essere copiato nel database di destinazione.

I trigger Capture avviano immediatamente la cattura delle informazioni. Diversamente dal programma Capture, non attendono un segnale dal programma Apply.

Di seguito viene descritto come i trigger Capture e il programma Apply comunicano, in un tipico scenario di replica, per assicurare l'integrità dei dati:

#### **Cattura dei dati da un'origine**

1.

Ogni volta che si verifica un'operazione DELETE, UPDATE o INSERT nella tabella di origine di replica registrata, un trigger Capture registra la modifica nella tabella CCD per tale tabella di origine.

#### **Applicazione dei dati ad una destinazione**

1. Per tutte le serie di richieste definite di nuovo, il programma Apply innanzitutto segnala ai trigger Capture di contrassegnare un punto di avvio valido nell'ambito della tabella CCD da cui avviare l'estrazione dei dati modificati. Quindi, viene eseguito un aggiornamento completo per ciascun membro della serie (a meno che non è una tabella di destinazione completa).
2. Quando il programma Apply elabora una serie di richieste per un'origine relazionale non DB2, aggiorna la tabella IBMSNAP\_REG\_SYNCH, ciò dà luogo all'attivazione di un trigger UPDATE su tale tabella. Il trigger aggiorna il valore SYNCHPOINT nella tabella IBMSNAP\_REGISTER per contrassegnare il valore SYNCHPOINT più elevato nelle tabelle CCD copiate nelle destinazioni. Durante il ciclo successivo, il programma Apply elabora nuovi dati nella tabella CCD che ha un valore SYNCHPOINT inferiore o uguale a questo SYNCHPOINT. Poiché la tabella IBMSNAP\_REG\_SYNCH è nel database non DB2, il programma Apply scrive sulla tabella utilizzando il nickname creato dal Centro di replica.
3. Il programma Apply verifica la tabella IBMSNAP\_REGISTER per determinare se sono presenti modifiche di cui deve essere eseguita la replica.
4. Il programma Apply copia le modifiche dalla tabella CCD alla tabella di destinazione.
5. Il programma Apply aggiorna la tabella IBMSNAP\_SUBS\_SET per registrare i dati che il programma Apply ha copiato per ogni serie di richieste.
6. Il programma Apply aggiorna la tabella IBMSNAP\_PRUNCNTL per ogni origine registrata con un valore che indica il punto su cui ha letto le modifiche dalla tabella CCD.

#### **Eliminazione delle tabelle CCD**

Il trigger UPDATE sulla tabella IBMSNAP\_PRUNCNTL verifica tutte le tabelle CCD nel database di origine ed elimina le modifiche di cui è già stata eseguita la replica dalla tabella CCD.

---

## Strumenti di gestione e Replication Alert Monitor

Quando si definisce una condizione di segnalazione con i contatti a cui verrà notificato quando si verifica la condizione, il Centro di replica o i programmi della riga comandi ASNCLP creano uno script SQL che memorizza le informazioni per tale condizione di segnalazione ed i relativi contatti nelle tabelle di controllo della replica che contengono tutte le informazioni di notifica e condizioni di segnalazione.

Vengono aggiornate le seguenti tabelle di controllo:

- Tabella IBMSNAP\_CONDITIONS
- Tabella IBMSNAP\_CONTACTS
- Tabella IBMSNAP\_GROUPS
- Tabella IBMSNAP\_CONTACTGRP

Le tabelle IBMSNAP\_CONDITIONS contengono una riga per ogni condizione che si desidera controllare. La tabella contiene i seguenti tipi di informazione su ogni condizione di segnalazione:

- Il qualificatore Monitor
- Il nome e gli alias del server Capture o Apply che si desidera controllare
- Il componente che si desidera controllare (il programma Capture o Apply)
- Lo schema Capture o il qualificatore Apply
- Il nome della serie di richieste (se si desidera controllare una serie)
- La condizione di segnalazione che si desidera controllare
- Il contatto a cui eseguire la notifica se si verifica la condizione

Questa tabella ha diverse colonne in più per le informazioni correlate.

Le altre tabelle per Replication Alert Monitor contengono informazioni relative a chi verrà eseguita la notifica se si verifica la condizione di segnalazione (un contatto singolo, un gruppo di contatti o la console z/OS), la modalità secondo cui verrà eseguita la notifica al contatto (via e-mail o cercapersone) e la frequenza con cui verranno eseguite le notifiche nel caso la condizione persista.

---

## Replication Alert Monitor, il programma Capture e il programma Apply

Replication Alert Monitor utilizza alcune tabelle di controllo Capture per controllare il programma Capture ed utilizza alcune tabelle di controllo Apply per controllare il programma Apply. Legge diverse tabelle di controllo della replica su ciascun server di controllo Capture o Apply, a seconda di ciò di cui sta eseguendo la modifica.

Replication Alert Monitor non interferisce o comunica con il programma Capture o Apply.

La seguente procedura descrive la modalità secondo cui Replication Alert Monitor controlla le condizioni per il programma Capture o Apply e notifica ai contatti quando si verifica una condizione di segnalazione:

1. Replication Alert Monitor legge le condizioni di segnalazione e il contatto per ogni condizione (per un qualificatore Monitor) nella tabella IBMSNAP\_CONDITIONS.
2. Per ogni server di controllo Capture o Apply che ha una condizione di segnalazione definita, Replication Alert Monitor esegue le seguenti attività:

- a. Replication Alert Monitor si collega al server e legge le tabelle di controllo della replica associate ad ogni condizione di segnalazione per tale server per verificare se qualcuna delle condizioni è soddisfatta.
- b. In caso affermativo, Replication Alert Monitor memorizza i dati correlati a tale condizione in memoria e continua l'elaborazione delle rimanenti condizioni di segnalazione per tale server.
- c. Una volta terminata l'elaborazione di tutte le condizioni di segnalazione per tale server, Replication Alert Monitor si scollega dal server di controllo Capture o Apply, inserisce le segnalazioni nella tabella IBMSNAP\_ALERTS e notifica ai contatti tale condizione.

---

## Capitolo 19. Visualizzazione dei prospetti relativi ai programmi di replica SQL

I seguenti argomenti descrivono i metodi che è possibile utilizzare per eseguire prospetti e analizzare l'ambiente di replica in uso. Utilizzare le informazioni per verificare lo stato attuale dei programmi di replica o per analizzare i dati cronologici per determinare i messaggi recenti e le statistiche di latenza o velocità di trasmissione.

---

### Verifica dello stato dei programmi di replica (z/OS, Linux, UNIX, Windows)

z/OS

Linux UNIX Windows

È possibile valutare velocemente lo stato attuale del programma Capture, Apply o Replication Alert Monitor.

Utilizzare uno dei seguenti comandi per verificare lo stato dei programmi di replica:

#### **Programma Capture**

comando di sistema `asncmd`, parametro **status**

#### **Programma Apply**

comando di sistema `asnacmd`, parametro **status**

#### **Controllo degli avvisi di replica**

comando di sistema `asnmcmd`, parametro **status**

Quando si verifica lo stato di un programma, si ricevono messaggi che descrivono lo stato di ogni thread associato a tale programma:

- Il programma Capture dispone dei seguenti thread:
  - Thread di lavoro
  - Thread di amministrazione
  - Thread di eliminazione
  - Thread di serializzazione
  - Thread del programma di lettura delle transazioni thread (se il parametro di avvio **asynchlogrd** è impostato su sì)
- Il programma Apply dispone dei seguenti thread:
  - Thread di amministrazione
  - Thread di lavoro
  - Thread di serializzazione
- Il programma Replication Alert Monitor dispone dei seguenti tre thread:
  - Thread di amministrazione
  - Thread di lavoro
  - Thread di serializzazione

Utilizzare i messaggi che si ricevono per determinare se i programmi in uso funzionano correttamente. Di solito, i thread di lavoro, di amministrazione e di

eliminazione sono in funzione ed eseguono le attività per la cui esecuzione sono stati predisposti. I thread di serializzazione, gestori di segnali globali, sono tipicamente in stato di attesa dei segnali.

Il thread di eliminazione elimina le tabelle CD e le seguenti tabelle di controllo di replica.

- Tabella IBMSNAP\_UOW
- Tabella IBMSNAP\_CAPTRACE
- Tabella IBMSNAP\_CAPMON
- Tabella IBMSNAP\_SIGNAL

Il thread di lavoro Apply imposta il proprio stato su "in attesa del database" se non è in grado di connettersi al proprio server di controllo Apply e se il programma Apply è stato avviato con il parametro **term=n**. È possibile eseguire il comando di stato **asnacmd** o **MODIFY** su z/OS per controllare se il thread di lavoro Apply è in esecuzione ma non è in grado di connettersi al server di controllo.

Se i messaggi che si ricevono indicano che è in esecuzione un programma, ma si ha evidenza del contrario nell'ambiente in uso, è necessario eseguire ulteriori verifiche. Ad esempio, se si verifica lo stato del programma Apply e si individua che è in funzione il thread di lavoro, ma si nota che i dati non vengono applicati alle tabelle di destinazione come si prevedeva, verificare nella tabella **IBMSNAP\_APPLYTRAIL** la presenza di messaggi per potrebbero spiegare il perché della mancata applicazione dei dati. Problemi relativi a risorse di sistema potrebbero impedire al programma un funzionamento corretto.

---

## Analisi dei dati cronologici per le tendenze

È possibile analizzare i dati cronologici delle recenti operazioni di replica e valutare i dati per le tendenze. Le tendenze che si riconoscono nel tempo potrebbero dimostrare che viene eseguita la replica di un volume fisso di dati o che per migliorare le prestazioni potrebbero essere apportate delle modifiche.

È possibile analizzare i dati cronologici delle recenti operazioni di replica e valutare i dati per le tendenze. Le tendenze che si riconoscono nel tempo potrebbero dimostrare che viene eseguita la replica di un volume fisso di dati o che per migliorare le prestazioni potrebbero essere apportate delle modifiche.

I dati cronologici vengono desunti dalle seguenti tabelle di controllo:

- **IBMSNAP\_APPLYTRAIL**
- **IBMSNAP\_APPLYTRACE**
- **IBMSNAP\_CAPMON**
- **IBMSNAP\_CAPTRACE**

La frequenza con cui vengono eliminate tali tabelle influisce sui prospetti che è possibile generare. Si consiglia di conservare almeno una settimana di dati in tali tabelle in modo da poter esaminare i dati durante la risoluzione dei problemi o la valutazione delle prestazioni.

La Tabella 26 descrive i dati cronologici che è possibile visualizzare.

*Tabella 26. Dove rilevare informazioni cronologiche*

<b>Per rispondere a questa domanda:</b>	<b>Utilizzare la seguente finestra del Centro di replica:</b>
Quali sono i messaggi recenti dai programmi Capture, Apply e Monitor?	Messaggi Capture Messaggi Apply Messaggio Monitor
Relativamente alla media: <ul style="list-style-type: none"> <li>• Quante righe sono state elaborate nella tabella CD in un dato periodo di tempo?</li> <li>• Quante righe sono state eliminate?</li> <li>• Di quante transazioni è stato eseguito il commit?</li> <li>• Quanta memoria utilizza il programma Capture?</li> </ul>	Analisi della velocità di trasmissione di Capture
Relativamente alla media, qual'è la durata approssimativa che intercorre tra l'aggiornamento dei dati all'origine e la relativa cattura mediante il programma Capture?	Latenza Capture
Quali sono i messaggi recenti del programma Capture?	Applica prospetto
Relativamente alla media, <ul style="list-style-type: none"> <li>• Quante righe sono state elaborate nella tabella di destinazione in un dato periodo di tempo?</li> <li>• Quanto tempo è trascorso per l'elaborazione delle serie di richieste?</li> </ul>	Analisi della velocità di trasmissione di Apply
Relativamente alla media, quanto tempo è trascorso approssimativamente tra l'aggiornamento della tabella di origine e quello della corrispondente tabella di destinazione?	Latenza end-to-end

È possibile selezionare un intervallo di tempo per individuare la quantità di dati che si desidera analizzare. Specificare data e ora di inizio e fine di un intervallo di tempo; quindi, specificare che i risultati vengano visualizzati come media delle velocità calcolate. Per raggruppare i risultati, è possibile selezionare intervalli di tempo (un secondo, un minuto, un'ora, un giorno o una settimana). Ad esempio, se si sceglie di analizzare la velocità di trasmissione di Apply tra le 9:00 p.m. e le 9:59 p.m., e si desidera che i dati vengano visualizzati ad intervalli di un minuto, i risultati vengono visualizzati in 60 righe, ognuna delle quali riassume l'attività di un singolo minuto nell'ambito dell'intervallo di 60 minuti. In alternativa, se si sceglie un intervallo di un'ora, i risultati vengono visualizzati in 1 riga, e riportano la velocità di trasmissione media per l'ora specificata. Se non si specifica un intervallo, è possibile visualizzare i dati non elaborati dalla tabella IBMSNAP\_APPLYTRAIL.

Le finestre del Centro di replica mostrano i risultati derivanti dalle informazioni contenute in diverse tabelle di controllo e file di log. I seguenti argomenti descrivono come è possibile utilizzare i dati cronologici per valutare le operazioni di replica con il Centro di replica.

## Analisi dei messaggi del programma Capture

Utilizzare la finestra Messaggi Capture per analizzare i messaggi inseriti nella tabella IBMSNAP\_CAPTRACE in un determinato periodo di tempo. La tabella IBMSNAP\_CAPTRACE contiene una riga per eventi significativi come l'inizializzazione, l'eliminazione, le avvertenze e gli errori emessi dal programma Capture.

Ad esempio, utilizzando la finestra Messaggi Capture, è possibile analizzare tutti i messaggi di errore e avvertenza registrati dal programma Capture in una settimana. Da tale finestra è inoltre possibile stampare o salvare i dati in un file.

## Esame della velocità di trasmissione del programma Capture

Utilizzare la finestra di analisi della velocità di trasmissione di Capture per visualizzare i risultati delle prestazioni di un programma Capture in un determinato periodo di tempo. Il programma Capture registra regolarmente informazioni statistiche nella tabella IBMSNAP\_CAPMON e durante l'eliminazione registra le statistiche di eliminazione nella tabella IBMSNAP\_CAPTRACE.

Utilizzando informazioni provenienti da queste tabelle, la finestra di analisi della velocità di trasmissione di Capture visualizza il calcolo dei risultati della velocità di prestazione di quattro diverse attività. È possibile esaminare i risultati di tutti e quattro i tipi di informazioni per valutare le prestazioni della velocità di trasmissione del programma Capture. Si ha la possibilità di specificare che i risultati seguenti vengano visualizzati in valori assoluti o medi.

- Numero di righe inserite da una registrazione o ignorate
- Numero di righe eliminate da una tabella CD
- Numero di transazioni di cui è stato eseguito il commit
- Uso della memoria

Ad esempio, è possibile utilizzare la finestra Analisi della velocità di trasmissione di Capture per analizzare le prestazioni medie settimanali della velocità di trasmissione del programma Capture. A tal fine, specificare data e ora di inizio e fine di un intervallo di tempo; quindi, specificare che i risultati vengano visualizzati come media delle velocità calcolate.

## Visualizzazione latenza dei dati elaborati dal programma Capture

Utilizzare la finestra Latenza Capture per approssimare il periodo di tempo intercorso tra l'aggiornamento di alcuni dati all'origine e la relativa cattura mediante il programma Capture. Questo tempo trascorso fornisce alcune indicazioni sulla ricorrenza dei dati nelle tabelle CD nel tempo. Tale latenza media deriva dalle informazioni ubicate nella tabella IBMSNAP\_CAPMON, che deriva le relative informazioni dalla tabella IBMSNAP\_REGISTER.

L'attuale latenza Capture viene calcolata come la differenza tra il valore CURRENT\_TIMESTAMP nella colonna SYNCHTIME e il record globale nella tabella IBMSNAP\_REGISTER:

$(CURRENT\_TIMESTAMP) - (SYNCHTIME)$

*Tabella 27. Esempio di valori per il calcolo della latenza Capture attuale*

Parametro	Valore colonna
CURRENT_TIMESTAMP	2006-10-20-10:30:25



Tabella 27. Esempio di valori per il calcolo della latenza Capture attuale (Continua)

Parametro	Valore colonna
SYNCHTIME	2006-10-20-10:30:00

Ad esempio, utilizzando i valori in Tabella 27 a pagina 254, la latenza attuale è 25 secondi:

10:30:25 - 10:30:00

Il periodo di latenza Capture varia al passare del tempo e la cronologia di tali cambiamenti viene memorizzata nella tabella IBMSNAP\_CAPMON. Il Centro di replica utilizza inoltre informazioni ubicate nella tabella di controllo Capture per calcolare la latenza cronologica o media. Tale formula differisce da quella utilizzata per calcolare la latenza attuale poiché per calcolare la latenza media utilizza il valore MONITOR\_TIME anziché il valore CURRENT\_TIMESTAMP. Il valore MONITOR\_TIME è una data/ora che indica quando il programma Capture ha inserito una riga nella tabella di controllo Capture. È possibile visualizzare la latenza media per secondo, minuto, ora, giorno o settimana. Ad esempio, dalla finestra Latenza Capture, è possibile visualizzare la latenza media per un programma Capture per ora, durante l'ultima settimana.

## Analisi dei messaggi del programma Apply

Utilizzare la finestra Messaggi Apply per analizzare i messaggi inseriti nella tabella IBMSNAP\_APPLYTRACE in un determinato periodo di tempo. La tabella IBMSNAP\_APPLYTRACE contiene righe per eventi significativi come l'inizializzazione, le avvertenze e gli errori che potrebbero essere emessi dal programma Apply.

Ad esempio, dalla finestra Messaggi Apply, è possibile analizzare tutti i messaggi di errore e avvertenza che potrebbero essere stati registrati dal programma Apply in una settimana. Da tale finestra è inoltre possibile stampare o salvare tali dati in un file.

Utilizzare la finestra Prospetto Apply per verificare l'esito di un determinato programma Apply in uno specifico periodo di tempo analizzando i dati inseriti nella tabella IBMSNAP\_APPLYTRAIL. La tabella IBMSNAP\_APPLYTRAIL contiene dati relativi all'esecuzione di serie di richieste, comprendenti il relativo stato, messaggi di errore e il numero di righe elaborate.

Nella finestra Prospetto Apply, è possibile visualizzare i seguenti dati:

- Tutte le serie di richieste
- Le serie di richieste con esito negativo
- Le serie di richieste con esito positivo
- Riepiloghi di errori relativi a serie di richieste con esito negativo

Ad esempio, dalla finestra Prospetto Apply, è possibile determinare se il programma Apply ha elaborato con esito positivo le serie di richieste durante l'ultima settimana. È possibile visualizzare i messaggi di errore emessi dal programma Apply per qualsiasi serie di richieste di cui potrebbe non essere eseguita la replica. Inoltre, è possibile utilizzare la finestra Prospetto Apply insieme alla finestra di analisi della velocità di trasmissione di Apply. Una volta utilizzata la finestra Prospetto Apply per individuare le serie di cui è stata eseguita la replica

con esito positivo, è possibile utilizzare la finestra di analisi della velocità di trasmissione di Apply per determinare il numero di righe replicate e il tempo necessario all'esecuzione della replica.

È possibile inoltre utilizzare la finestra Prospetto Apply per visualizzare tutti i dati di una particolare riga della tabella IBMSNAP\_APPLYTRAIL.

## Esame della velocità di trasmissione del programma Apply

Utilizzare la finestra di analisi della velocità di trasmissione di Apply per esaminare le statistiche delle prestazioni per un determinato qualificatore Apply. È possibile eseguire il filtro e raggruppare i dati senza scrivere istruzioni SQL.

Utilizzare la finestra di analisi della velocità di trasmissione di Apply per esaminare le statistiche delle prestazioni per un determinato qualificatore Apply. È possibile eseguire il filtro e raggruppare i dati senza scrivere istruzioni SQL.

Ad esempio, è possibile visualizzare il numero di righe inserite, aggiornate, eliminate e rielaborate nelle tabelle di destinazione nella serie di richieste elaborata da un determinato qualificatore Apply. È possibile inoltre determinare il tempo necessario al programma Apply per elaborare le serie di richieste relative ad un determinato qualificatore Apply.

## Visualizzazione della durata media della replica di transazioni

Utilizzare la finestra Latenza end-to-end per visualizzare un valore approssimato della durata media della replica di transazioni in una determinata serie di richieste.

Dalla finestra Latenza end-to-end, ad esempio, è possibile visualizzare la latenza approssimata di una serie di richieste per ogni ciclo Apply durante un periodo di tempo. È possibile inoltre dividere il periodo di tempo in intervalli e visualizzare la latenza media per ciascun intervallo.

Il Centro di replica utilizza la seguente formula per calcolare la latenza end-to-end:  $(\text{ENDTIME} - \text{LASTRUN}) + (\text{SOURCE\_CONN\_TIME} - \text{SYNCHTIME})$

- ENDTIME è l'ora in cui il programma Apply termina l'elaborazione della serie di richieste.
- LASTRUN è l'ora in cui il programma Apply avvia l'elaborazione di una serie di richieste.
- SOURCE\_CONN\_TIME è l'ora in cui il programma Apply si collega al server di controllo Capture per estrarre i dati.
- SYNCHTIME è l'ora in cui è stato eseguito il commit dei dati più recente sulle tabelle CD dal programma Capture.

Tabella 28. Esempio di valori per il calcolo della latenza end-to-end

Parametro	Valore colonna
ENDTIME	2006-10-20-10:01:00
LASTRUN	2006-10-20-10:00:30
SOURCE_CONN_TIME	2006-10-20-10:00:32
SYNCHTIME	2006-10-20-10:00:00

Ad esempio, si assuma che una determinata serie di richieste ha i valori mostrati in Tabella 28. Mediante l'uso della precedente equazione, la latenza end-to-end media per questa serie di richieste è 62 secondi:

---

## Checking the status of the Capture and Apply journal jobs (System i)

### System i

On DB2 for System i, use the Work with Subsystem Jobs (WRKSBSJOB) system command to check the status of the journal jobs for the Capture and Apply programs.

#### Procedure

To check the status of the journal jobs for the Capture and Apply programs:

Use the Work with Subsystem Jobs (WRKSBSJOB) system command as follows:

1. Enter the command:

```
WRKSBSJOB subsystem
```

Where *subsystem* is the subsystem name. In most cases, the subsystem is QZSNDPR, unless you created your own subsystem description

2. Identify jobs of interest from among those listed as running.

The journal job is named after the journal to which it was assigned. If no job is listed there, use the Work with Submitted Jobs (WRKSBJOB) system command or the Work with Job (WRKJOB) system command to locate the job. Find the job's joblog to verify that it completed successfully or to identify why it failed.

---

## Monitoring the progress of the Capture program (System i)

### System i

If the Capture program has terminated, you can inspect the IBMSNAP\_RESTART table to determine how much progress the Capture program made before termination. There is one row for each journal used by the source tables. The LOGMARKER column provides the timestamp of the last journal entry processed successfully. The SEQNBR column provides the sequence number of that journal entry.

#### Informazioni su questa attività

If the Capture program has terminated, you can inspect the IBMSNAP\_RESTART table to determine how much progress the Capture program made before termination. There is one row for each journal used by the source tables. The LOGMARKER column provides the timestamp of the last journal entry processed successfully. The SEQNBR column provides the sequence number of that journal entry.

#### Procedure

To determine progress of the Capture program while it is running:

1. Open the CD table for each source table being captured.
2. In the last row of each CD table, note the hex value in the COMMITSEQ column.

3. Identify a row in the IBMSNAP\_UOW table with the same COMMITSEQ hex value. If no matching COMMITSEQ value exists in the IBMSNAP\_UOW table, repeat the process with the second-to-last row in the CD table. Proceed backward through the CD table until you identify a matching hex value.
4. When you find a matching COMMITSEQ hex value, note the value in the LOGMARKER column of the UOW row. This is the timestamp of the last journal entry processed. All changes to the source table up to that time are ready to be applied.
5. Use the Display Journal (DSPJRN) system command to determine how many journal entries remain to be processed by the Capture program. Direct the output to an output file (or printer) to preserve the report, as shown in the following example:

```
DSPJRN FILE(JRNLIB/DJRN1)
       RCVRNG(*CURCHAIN)
       FROMTIME(timestamp)
       TOTIME(*LAST)
       JRNCDE(J F R C)
       OUTPUT(*OUTFILE)
       ENTDTALEN(1) OUTFILE(library/outfile)
```

where *timestamp* is the timestamp that you identified in 4.

The number of records in the output file is the approximate number of journal entries that remain to be processed by the Capture program.

---

## Capitolo 20. Personalizzazione ed esecuzione di script SQL per la replica

Per creare tabelle di controllo, registrare tabelle di origine e creare membri e serie di richieste, è necessario eseguire script SQL generati da Centro di replica e programma della riga comandi ASNCLP. È possibile eseguire gli script SQL utilizzando il Centro di replica oppure da una riga comandi DB2. Se necessario, è possibile modificare gli script SQL per soddisfare le proprie necessità.

### Prima di iniziare

Se si eseguono gli script SQL da una riga comandi DB2, è necessario collegarsi manualmente ai server quando si esegue lo script SQL e modificare le istruzioni SQL per specificare l'ID utente e la password per il server a cui si sta eseguendo la connessione. Ad esempio, ricercare una riga simile all'esempio riportato di seguito ed aggiungere le proprie informazioni sostituendo i caratteri XXXX:

```
CONNECT TO srcdb USER XXXX USING XXXX ;
```

### Informazioni su questa attività

In ASNCLP e nel Centro di replica è possibile eseguire immediatamente uno script SQL generato o salvarlo per eseguirlo successivamente. Anche se si sceglie di eseguire ora l'SQL, si potrebbe anche volerlo salvare come riferimento futuro. Ad esempio, se si salvano le definizioni di una serie di richieste di replica di grandi dimensioni in un file SQL, è possibile eseguire nuovamente le definizioni come necessario.

Quando si modificano gli script SQL creati, non modificare i caratteri finali. Inoltre, non modificare i separatori degli script se in un file sono salvati più script.

È possibile che si desideri personalizzare gli script SQL per il proprio ambiente per effettuare le attività riportate di seguito:

- Creare più copie della stessa azione di replica, personalizzata per più server.
- Impostare la dimensione di tablespace o database delle tabelle CD.
- Definire standard specifici del sito.
- Combinare le definizioni tra loro ed eseguire come lavoro batch.
- Ritardare la replica fino ad un'ora specificata.
- Creare librerie di script SQL per il backup, la personalizzazione specifica del sito o l'esecuzione in modalità autonoma su siti distribuiti, come per un ambiente collegato occasionalmente.
- Modificare le istruzioni di creazione tabella e indice per rappresentare gli oggetti di database.
- Relativamente a Informix e altri database relazionali non DB2, accertarsi che le tabelle vengano create nei dbspace o tablespace desiderati.
- Relativamente a Microsoft SQL Server, creare tabelle di controllo su un segmento esistente.
- Analizzare e modificare predicati di membri di serie di richieste come modalità di definizione contemporanea di più serie di richieste. È possibile utilizzare variabili di sostituzione nei predicati in uso e risolvere le variabili con la logica di programmazione.

## Procedure

Utilizzare uno dei metodi riportati di seguito per eseguire i file che contengono script SQL da una riga comandi DB2:

- Utilizzare questo comando se lo script SQL ha il punto e virgola ( ; ) come carattere finale: `db2 -tvf filename`
- Utilizzare questo comando se lo script SQL ha un altro carattere come delimitatore (in questo esempio, come nella replica eterogenea, il cancelletto ( # ) è il carattere finale): `db2 -td# -vf filename`

---

## Capitolo 21. Regole di denominazione per gli oggetti della replica SQL

Nella seguente tabella sono elencati i limiti per i nomi degli oggetti della replica.

Tabella 29. Limiti per il nome degli oggetti della replica

Oggetto	Limiti per il nome
Tabella di origine e destinazione	<p><b>Linux UNIX Windows</b> Seguire le regole di denominazione per il sistema di gestione di database in uso.</p> <p><b>System i</b> I nomi non possono comprendere spazi bianchi, asterischi (*), punti di domanda (?), apici ('), virgolette (") o una barra (/).</p>
Colonne di origine e destinazione	Seguire le regole di denominazione per il sistema di gestione di database in uso. (Tenere presente che tutte le colonne pre-immagine hanno un prefisso di un carattere. Per evitare nomi di colonna pre-immagine ambigui, accertarsi che i nomi di colonna di origine siano univoci di 29 caratteri e che i nomi di colonna pre-immagine non siano in conflitto con i nomi di colonna esistenti quando il prefisso di un carattere pre-immagine viene aggiunto al nome di colonna).
Serie di richieste	Un nome di serie di richieste può includere qualsiasi carattere consentito da DB2 per le colonne VARCHAR (Varying-Character). <b>suggerimento:</b> seguire le regole di denominazione relative ai nomi delle colonne e delle tabelle DB2. Poiché la replica DB2 memorizza il nome della serie di richieste in ciascun server di controllo della replica, accertarsi che il nome sia compatibile con le pagine di codice di tutti e tre i server.
Schema Capture	<p><b>Linux UNIX Windows</b> Lo schema Capture può essere una stringa di 30 caratteri o un numero inferiore<sup>1</sup>.</p> <p><b>z/OS</b> Lo schema Capture può essere una stringa di 18 caratteri o un numero inferiore; sui sottosistemi in modalità nuova funzione DB2 UDB per z/OS Versione 8 può essere di 128 caratteri<sup>1</sup>.</p> <p><b>System i</b> Lo schema Capture (CAPCTLLIB) può essere una stringa di 10 caratteri alfanumerici o un numero inferiore<sup>1</sup>.</p>
Qualificatore Apply	<p><b>z/OS</b> <b>Linux UNIX Windows</b> Il qualificatore Apply può essere una stringa di 18 caratteri o un numero inferiore<sup>1</sup>.</p> <p><b>System i</b> Il qualificatore Apply può essere una stringa di 18 caratteri o un numero inferiore ma, poiché i lavori di Apply possono essere lunghi fino ad un massimo di 10 caratteri, i primi 10 caratteri devono essere univoci per un dato qualificatore Apply<sup>1</sup>.</p>

Tabella 29. Limiti per il nome degli oggetti della replica (Continua)

Oggetto	Limiti per il nome
Qualificatore Monitor	<div style="display: flex; justify-content: space-around; align-items: center;"> <div style="background-color: #800000; color: white; padding: 2px 5px; font-weight: bold;">z/OS</div> <div style="background-color: #800000; color: white; padding: 2px 5px; font-weight: bold;">Linux UNIX Windows</div> </div> Il qualificatore Monitor può essere una stringa di 18 caratteri o un numero inferiore <sup>1</sup> .

**Nota:**

1. Relativamente agli schemi Capture, i qualificatori Apply e Monitor assicurano che si utilizzi solo i seguenti caratteri validi nei nomi di tali oggetti:

- dalla A alla Z (lettere maiuscole)
- dalla a alla z (lettere minuscole)
- Numerali (da 0 a 9)
- Il carattere di sottolineatura " \_ "

Gli spazi non sono consentiti; né altri caratteri speciali quali i due punti ":" e il segno più "+".

I comandi di sistema della replica e il Centro di replica, per impostazione predefinita, convertono tutti i nomi forniti in maiuscolo. Racchiudere tra virgolette un nome con caratteri maiuscoli e minuscoli (o qualsiasi carattere per cui il sistema di destinazione è configurato per l'uso) per conservare il maiuscolo/minuscolo e salvare il nome esattamente come è stato immesso. Ad esempio, immettendo `myqual` o `MyQual` o `MYQUAL`, il nome viene salvato come `MYQUAL`. Se questi stessi nomi vengono immessi e racchiusi tra virgolette, essi sono salvati come `myqual` o `MyQual` o `MYQUAL`, rispettivamente. Alcuni sistemi operativi non riconoscono le virgolette e potrebbe essere necessario utilizzare un carattere escape, di solito una barra retroversa (\).

**Windows** Sui sistemi operativi Windows, è necessario utilizzare un percorso univoco per differenziare i nomi che sono identici. Ad esempio, supporre di disporre di tre qualificatori Apply: `myqual`, `MyQual` e `MYQUAL`. I tre nomi utilizzano gli stessi caratteri ma con lettere maiuscole/minuscole differenti. Se questi tre qualificatori sono contenuti nella stesso percorso Apply, essi provocheranno conflitti di nomi.

**Importante:** quando vengono impostati i servizi Windows per Capture, Apply o Replication Alert Monitor, è necessario utilizzare nomi univoci per lo schema Capture, il qualificatore Apply e il qualificatore Monitor. Non è possibile utilizzare il maiuscolo/minuscolo per differenziare i nomi.



---

## Capitolo 22. Comandi di sistema per la replica SQL (Linux, UNIX, Windows, z/OS)

z/OS

Linux UNIX Windows

Questa sezione descrive i comandi per Linux, UNIX, Windows e UNIX System Services (USS) su z/OS che permettono l'avvio, l'utilizzo, la modifica e il controllo dei programmi di replica SQL.

Tutti questi comandi hanno un prefisso asn e sono immessi in una richiesta comandi del sistema operativo o in uno script shell. Uno dei comandi, asnanalyze, operano anche con dati remoti che risiedono sui sistemi operativi System i.

---

### asncap: Avvio Capture

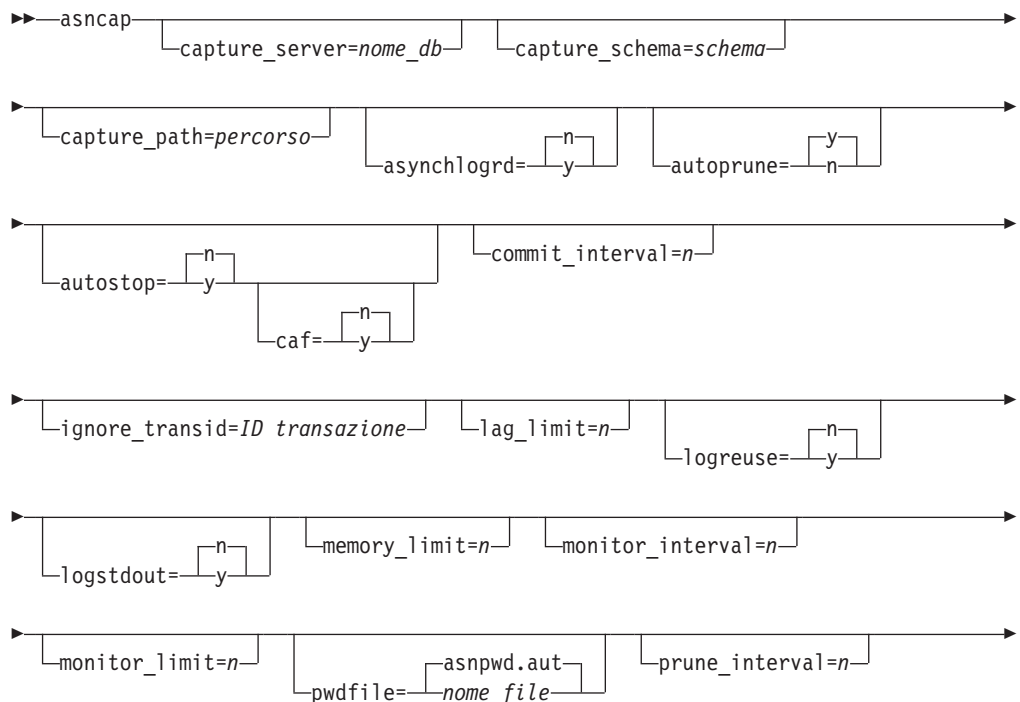
z/OS

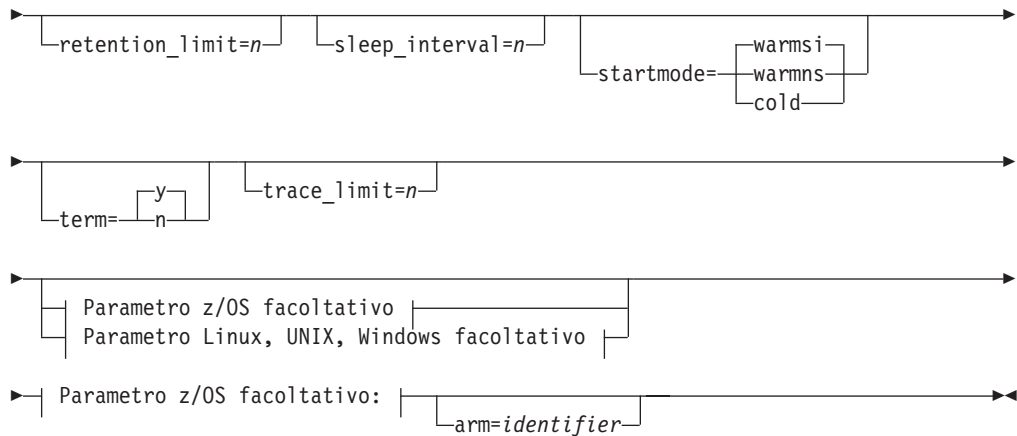
Linux UNIX Windows

Utilizzare il comando asncap per avviare il programma Capture su Linux, UNIX, Windows e UNIX System Services (USS) su z/OS. Eseguire questo comando alla richiesta del sistema operativo o in uno script shell.

Dopo aver avviato il programma Capture, esso rimane in esecuzione fino a quando non viene arrestato o quando rileva un errore non ripristinabile.

#### Sintassi





### Parametro Linux, UNIX, Windows facoltativo:



## Parametri

Tabella 30 definisce i parametri di richiamo.

Tabella 30. Definizione del parametro di richiamo `asncap` per i sistemi operativi Linux, UNIX, Windows e z/OS

Parametri	Definizione
<code>capture_server=nome_db</code>	<p>Specifica il nome del server di controllo Capture.</p> <p><b>z/OS</b> Specifica il nome del sottosistema DB2 dove sarà eseguito il programma Capture. Per la condivisione dei file, non utilizzare il nome di collegamento del gruppo. Specificare invece un nome membro del sottosistema.</p> <p><b>Linux UNIX Windows</b> Se non viene specificato un server di controllo Capture, questo parametro fa riferimento per impostazione predefinita al valore della variabile d'ambiente DB2DBDFT.</p>
<code>add_partition=y/n</code>	<p><b>Linux UNIX Windows</b> Specifica se il programma Capture viene avviato leggendo il file di log per le partizioni appena aggiunte dall'ultima volta che il programma Capture è stato riavviato.</p> <p><b>n (predefinito)</b> Nessuna nuova partizione è stata aggiunto dall'ultimo avvio del programma Capture.</p> <p><b>y</b> Il programma Capture avvia la lettura del file di log su una o più delle nuove partizioni. In ogni partizione, il programma Capture avvia la lettura della registrazione dall'LSN (log sequence number) utilizzato inizialmente l'ultima volta che è stato avviato il database.</p>

Tabella 30. Definizione del parametro di richiamo *asncap* per i sistemi operativi Linux, UNIX, Windows e z/OS (Continua)

Parametri	Definizione
<i>arm=identififer</i>	<p><b>z/OS</b> Specifica una stringa alfanumerica di tre caratteri utilizzata per identificare una singola istanza del programma Capture all'Automatic Restart Manager. Il valore fornito viene aggiunto al nome dell'elemento ARM che Capture genera per se stesso: <i>ASNTCxxxxyyyy</i> (dove <i>xxxx</i> è il nome allegato del gruppo di condivisione dati e <i>yyyy</i> è il nome del membro DB2). È possibile specificare qualsiasi lunghezza di stringa per il parametro <b>arm</b>, ma il programma Capture concatenerà al nome attuale solo un massimo di tre caratteri. Se necessario, il programma Capture aggiungerà spazi bianchi al nome per renderlo un nome univoco di 16 byte.</p>
<i>asynchlogrd=y/n</i>	<p><b>n (predefinito)</b>            Specifica che si desidera che il programma Capture utilizzi lo stesso thread per leggere la registrazione di recupero DB2 e per elaborare le transazioni che sono state catturate dal log.</p> <p><b>y</b>            Specifica che si desidera che il programma Capture utilizzi un thread dedicato per catturare le transazioni dalla registrazione di recupero DB2. Il thread del programma di lettura delle transazioni effettua una prelettura delle transazioni di cui si è eseguito il commit in un buffer di memoria, da un altro thread riceve le transazioni e le elabora in istruzioni SQL per inserirle nella tabella CD. Questa modalità asincrona può migliorare la produttività di Capture in tutti gli ambienti, con particolari benefici per i database partizionati e la condivisione di dati z/OS. In sistemi con livelli di attività molto elevati, questa prelettura potrebbe portare un maggiore utilizzo di memoria. Regolare il parametro <b>memory_limit</b> di conseguenza. Se si dispone di un basso volume di modifiche, si può optare per il valore predefinito di N, per ridurre il consumo della CPU.</p>
<i>capture_schema=schema</i>	<p>Specifica il nome dello schema Capture utilizzato per identificare un particolare programma Capture. È necessario che il nome dello schema abbia un numero di caratteri compreso fra 1 e 30. Il valore predefinito è ASN.</p>
<i>capture_path=percorso</i>	<p>Specifica l'ubicazione dei file di lavoro utilizzati dal programma Capture. Il valore predefinito è la directory dove è stato richiamato il comando <i>asncap</i>.</p>

Tabella 30. Definizione del parametro di richiamo *asncap* per i sistemi operativi Linux, UNIX, Windows e z/OS (Continua)

Parametri	Definizione
<b>autoprune=y/n</b>	<p>Specifica se è abilitata l'eliminazione automatica delle righe nelle tabelle CD (change-data), UOW (unit-of-work), IBMSNAP_CAPMON, IBMSNAP_CAPTRACE e IBMSNAP_SIGNAL.</p> <p><b>y (predefinito)</b>                      Il programma Capture elimina automaticamente le righe idonee nell'intervallo specificato nella tabella IBMSNAP_CAPPARMS. Il programma Capture elimina le righe CD, UOW e IBMSNAP_SIGNAL più vecchie del limite di conservazione, indipendentemente dalle righe che sono state replicate.</p> <p><b>n</b>                      L'eliminazione automatica è disabilitata.</p>
<b>autostop=y/n</b>	<p>Specifica se il programma Capture si arresta dopo avere richiamato tutte le transazioni registrate prima dell'avvio del programma Capture.</p> <p><b>n (predefinito)</b>                      Il programma Capture non si arresta dopo aver richiamato le transazioni.</p> <p><b>y</b>                      Il programma Capture si arresta dopo aver richiamato le transazioni.</p>
<b>caf=n/y</b>	<p><b>z/OS</b> Il programma Apply viene eseguito con RRS (Recoverable Resource Manager Services) di connessione predefinito (CAF=n). È possibile sovrascrivere questo valore predefinito e indicare al programma Capture di utilizzare CAF (Call Attach Facility) specificando l'opzione <b>caf =y</b>. L'opzione <b>caf =y</b> specifica che il programma di replica sovrascriverà l'RRS di connessione predefinito e verrà eseguita con CAF di connessione.</p> <p><b>n (predefinito)</b>                      Il programma Capture utilizza l'RRS (Recoverable Resource Manager Services) di connessione (CAF=n).</p> <p><b>y</b>                      Specifica che il programma di replica sovrascriverà l'RRS di connessione predefinito e verrà eseguito con CAF di connessione.                      Se l'RRS non è disponibile, l'utente riceverà un messaggio e il programma di replica passerà a CAF. Il messaggio avvisa che il programma non è stato in grado di effettuare la connessione poiché l'RRS non è stato avviato. Il programma eseguirà pertanto un tentativo di utilizzo di CAF. Il programma verrà eseguito correttamente con CAF di connessione.</p>
<b>commit_interval=n</b>	<p>Specifica il numero di secondi che il programma Capture attende prima di eseguire il commit delle righe delle tabelle UOW (unit-of-work) e CD (change-data). Il valore predefinito è 30 secondi.</p>

Tabella 30. Definizione del parametro di richiamo *asncap* per i sistemi operativi Linux, UNIX, Windows e z/OS (Continua)

Parametri	Definizione
<b>ignore_transid=transaction_ID</b>	<p>Specifica che il programma Capture non catturerà la transazione identificata dal <i>transaction_ID</i>.</p> <p>Il valore per <i>transaction_ID</i> è un identificatore esadecimale a 10-byte nel seguente formato:</p> <p><b>z/OS</b></p> <p>0000:xxxx:xxxx:xxxx:mmmm</p> <p>Dove <i>xxxx:xxxx:xxxx</i> è l'ID transazione e <i>mmmm</i> è l'ID membro di dati condivisi. È possibile individuare l'ID membro negli ultimi 2 byte del dell'intestazione del record della registrazione nell'output LOGP. L'ID membro è 0000 se la condivisione dei dati non è abilitata.</p> <p><b>Linux UNIX Windows</b></p> <p>mmm:0000:xxxx:xxxx:xxx</p> <p>Dove <i>xxxx:xxxx:xxx</i> è l'ID della transazione e <i>mmm</i> è l'identificatore di partizione per il database partizionato (questo valore è 0000 se per database non partizionati).</p>
<b>lag_limit=n</b>	<p>Specifica il numero di minuti per cui è permesso al programma Capture di ritardare l'elaborazione dei record di registrazione. Il valore predefinito è 10080 minuti (sette giorni). Il programma Capture controlla il valore di questo parametro solamente durante un avvio a caldo. Se il limite viene superato, il programma Capture non si avvierà.</p>
<b>logreuse=y/n</b>	<p>Specifica se il programma Capture riutilizza o aggiunge i messaggi al file di log.</p> <p><b>n (predefinito)</b></p> <p>Il programma Capture aggiunge messaggi al file di log, anche dopo che il programma Capture è stato riavviato.</p> <p><b>y</b></p> <p>Il programma Capture riutilizza il file di log innanzitutto eliminando il file di log corrente e poi avviando una nuova registrazione quando il programma Capture viene riavviato.</p> <p><b>z/OS</b> Il nome del file di log non contiene il nome dell'istanza DB2 :  <i>capture_server.capture_schema.CAP.log.</i></p> <p><b>Linux UNIX Windows</b> Il nome del file di log comprende il nome dell'istanza DB2:  <i>db2instance.capture_server.capture_schema.CAP.log.</i></p>

Tabella 30. Definizione del parametro di richiamo *asncap* per i sistemi operativi Linux, UNIX, Windows e z/OS (Continua)

Parametri	Definizione
<b>logstdout=y/n</b>	<p>Specifica dove il programma Capture indirizza il file di log dei messaggi:</p> <p><b>n (predefinito)</b>                      Il programma Capture indirizza la maggior parte dei messaggi del file di log solamente al file di log. I messaggi di inizializzazione vengono memorizzati sia nel file di log che nell'output standard (STDOUT).</p> <p><b>y</b>                      Il programma Capture indirizza i messaggi sia al file di log che all'output standard (STDOUT).</p>
<b>memory_limit=n</b>	<p>Specifica la dimensione massima (in megabyte) della memoria che il programma Capture può utilizzare per creare una transazione. Dopo aver raggiunto questo limite di memoria, il programma Capture trasferisce le transazioni in un file. Il valore predefinito è 32 megabyte.</p> <p><b>z/OS</b> Se si specifica <b>memory_limit=0</b>, il programma Capture determina la quantità di memoria da utilizzare dal parametro della dimensione della regione del lavoro Capture. L'allocazione di memoria è l'80 per cento della dimensione della regione.</p>
<b>monitor_interval=n</b>	<p>Specifica la frequenza (in secondi) con cui il programma Capture inserisce righe nella tabella IBMSNAP_CAPMON. Il valore predefinito è 300 secondi (cinque minuti).</p>
<b>monitor_limit=n</b>	<p>Specifica quanto tempo (in minuti) una riga può rimanere nella tabella IBMSNAP_CAPMON prima di essere idonea per l'eliminazione. Tutte le righe IBMSNAP_CAPMON più vecchie del valore del parametro <b>monitor_limit</b> vengono eliminate nel successivo ciclo di eliminazione. Il valore predefinito è 10 080 minuti (sette giorni).</p>
<b>pwdfile=nomefile</b>	<p>Specifica il nome del file di password. Se non viene specificato un file di password, il valore predefinito è <code>asnpwd.aut</code>.</p> <p>Questo comando ricerca il file di password nella directory specificata dal parametro <b>capture_path</b>. Se non è specificato alcun parametro <b>capture_path</b>, questo comando ricerca il file di password nella directory da cui è stato richiamato il comando.</p>
<b>prune_interval=n</b>	<p>Specifica con che frequenza (in secondi) le tabelle CD (change-data), UOW (unit-of-work), IBMSNAP_CAPMON, IBMSNAP_CAPTRACE e IBMSNAP_SIGNAL vengono eliminate. Questo parametro è ignorato se il parametro <code>autoprunes</code> è impostato su <code>n</code>. Il valore predefinito è 300 secondi (cinque minuti).</p>
<b>retention_limit=n</b>	<p>Specifica per quanto tempo (in minuti) una riga può rimanere nelle tabelle CD (change-data), UOW (unit-of-work), o IBMSNAP_SIGNAL prima di diventare idonea per l'eliminazione. Tutte le righe più vecchie del parametro <b>retention_limit</b> vengono eliminate durante il successivo ciclo di eliminazione. Il valore predefinito è di 10 080 minuti (sette giorni).</p>

Tabella 30. Definizione del parametro di richiamo *asncap* per i sistemi operativi Linux, UNIX, Windows e z/OS (Continua)

Parametri	Definizione
<b>sleep_interval=<i>n</i></b>	<p>Specifica il numero di secondi per i quali il programma Capture si disattiva quando termina l'elaborazione di una registrazione attiva e determina che il buffer è vuoto. Il valore predefinito è cinque secondi.</p> <p><b>z/OS</b> Specifica il numero di secondi che il programma Capture si disattiva dopo che il buffer è pieno per meno della metà.</p>
<b>startmode=<i>modo</i></b>	<p>Specifica il processo di elaborazione utilizzata dal programma Capture durante un avvio di Capture.</p> <p><b>warmsi (predefinito)</b>                      Se l'informazione di avvio a caldo è disponibile, il programma Capture riprende l'elaborazione dove era stata terminata nel precedente avvio. Se questa è la prima volta che si avvia il programma Capture, esso passerà automaticamente ad un avvio a freddo.</p> <p>Durante un avvio a caldo, il programma Capture lascia intatte le tabelle IBMSNAP_CAPTRACE, CD (change-data), UOW (unit-of-work) e IBMSNAP_RESTART. In caso di errori dopo l'avvio del programma Capture è stato avviato, il programma Capture si arresta.</p> <p><b>warmns</b>                      Se l'informazione di avvio a caldo è disponibile, il programma Capture riprende l'elaborazione dove era stata terminata nel precedente avvio. In caso di errori dopo l'avvio del programma Capture è stato avviato, il programma Capture si arresta. Se il programma Capture non può essere avviato a caldo, esso non passa all'avvio a freddo.</p> <p><b>a freddo</b>                      Il programma Capture avvia l'eliminazione di tutte le righe nelle proprie tabelle CD e UOW. Tutte le richieste a queste origini di replica vengono completamente aggiornate durante il nuovo ciclo di elaborazione Apply. Le registrazioni per le CCD esterne e quelle richieste le quali destinazioni sono CCD incomplete non vengono aggiornate completamente.</p>

Tabella 30. Definizione del parametro di richiamo *asncap* per i sistemi operativi Linux, UNIX, Windows e z/OS (Continua)

Parametri	Definizione
<b>term=y/n</b>	<p>Specifica se il programma Capture termina se DB2 è terminato o è in stato di sospensione.</p> <p><b>y (predefinito)</b> Il programma Capture termina se DB2 termina o è in stato di sospensione.</p> <p><b>n</b> Il programma Capture continua l'esecuzione se DB2 termina o è in stato di sospensione. Quando DB2 si inizializza, il programma Capture si avvia a caldo e inizia l'acquisizione dal punto in cui si è interrotto quando DB2 è terminato o era in stato di sospensione.</p> <p>Se DB2 viene terminato mediante FORCE o a causa di una fine anomala, il programma Capture verrà terminato anche se questo parametro è stato impostato su n.</p> <p>Se questo parametro è stato impostato su n e DB2 viene avviato con accesso limitato (ACCESS MAINT), il programma Capture non può eseguire il collegamento e di conseguenza viene terminato.</p>
<b>trace_limit=n</b>	<p>Specifica per quanto tempo (in minuti) una riga può rimanere nella tabella IBMSNAP_CAPTRACE prima di diventare idonea per l'eliminazione. Tutte le righe IBMSNAP_CAPTRACE più vecchie del valore del parametro <b>trace_limit</b> vengono eliminate durante il ciclo di eliminazione successivo. Il valore predefinito è 10 080 minuti (sette giorni).</p>

## Codici di restituzione

Il comando *asncap* restituisce un codice di ritorno zero in caso di completamento con esito positivo. In caso di comando con esito negativo, viene restituito un codice di restituzione diverso da zero.

## Esempi per *asncap*

I seguenti esempi illustrano come utilizzare il comando *asncap*.

### Esempio 1

Per avviare per la prima volta un programma Capture che utilizza un server di controllo Capture denominato *db* e uno schema Capture ASN con file di lavoro ubicati nella directory `/home/files/capture/logs/`:

```
asncap capture_server=db capture_schema=ASN
capture_path=/home/files/capture/logs/ startmode=cold
```

### Esempio 2

Per riavviare un programma Capture senza che si verifichi l'eliminazione dopo che il programma è stato arrestato:

```
asncap capture_server=db autoprune=n sleep_interval=10 startmode=warmsi
```



Il programma Capture in questo esempio trattiene tutte le righe corrispondenti alle tabelle di controllo e si disattiva per dieci secondi dopo aver terminato l'elaborazione della registrazione attiva e determina che il buffer è vuoto. Il programma Capture riprende l'elaborazione dove era stata terminata nel precedente avvio e passa ad un avvio a freddo se l'informazione di avvio a caldo non è disponibile.

### Esempio 3

Per riavviare un programma Capture con una modalità di avvio warmns e con impostazioni modificate:

```
asncap capture_server=db autoprune=y prune_interval=60 retention_limit=1440
startmode=warmns
```

Questo comando riavvia il programma Capture e utilizza nuove impostazioni del parametro per diminuire la quantità di tempo prima che le tabelle CD, UOW e IBMSNAP\_SIGNAL diventino idonee per l'eliminazione e per aumentare la frequenza di eliminazione dalle impostazioni predefinite del parametro. Il programma Capture riprende l'elaborazione dove era stata terminata nel precedente avvio e passa ad un avvio a freddo se l'informazione di avvio a caldo non è disponibile.

### Esempio 4

Per avviare un programma Capture che effettui l'invio di tutti i file di lavoro ad una nuova sottodirectory chiamata capture\_files:

1. Andare alla directory appropriata e creare una nuova sottodirectory chiamata capture\_files:

```
cd /home/db2inst
mkdir capture_files
```

2. Avviare il programma Capture e specificare un percorso di Capture ubicato nella sottodirectory appena creata:

```
asncap capture_server=db capture_schema=ASN
capture_path=/home/db2inst/capture_files startmode=warmns
```

---

## asncmd: Funzionamento Capture

z/OS

Linux UNIX Windows

Utilizzare il comando asncmd per inviare un comando ad un programma Capture in esecuzione su Linux, UNIX, Windows e UNIX System Services (USS) su z/OS. Eseguire questo comando alla richiesta del sistema operativo o in uno script shell.

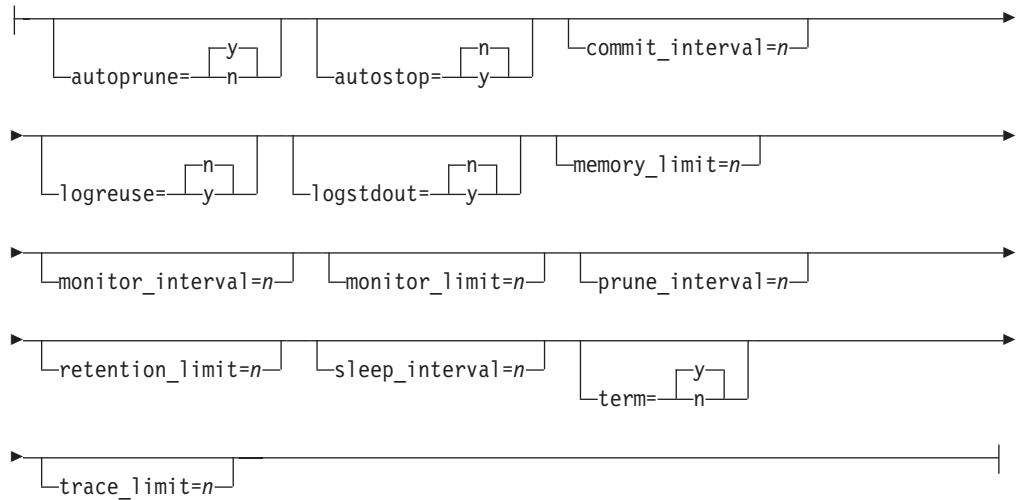
### Sintassi

Per informazioni sull'utilizzo del comando MVS MODIFY per l'invio di comandi a un programma Capture in esecuzione su z/OS, fare riferimento a Lavorare con i programmi di replica SQL utilizzando il comando MVS MODIFY.

```
asncmd [capture_server=nome_db] [capture_schema=schema]
```



**Parametri:**



**Parametri**

Tabella 31 definisce i parametri di richiamo per il comando asncmd.

Tabella 31. Definizioni per i parametri di richiamo asncmd

Parametri	Definizione
capture_server=y/n	<p>Specifica il nome del server di controllo Capture.</p> <p><b>z/OS</b></p> <p>Il nome del server database che si connette al server di controllo. Per la condivisione dati, utilizzare il nome di collegamento del gruppo oppure il nome di un membro del sottosistema.</p> <p><b>Linux UNIX Windows</b></p> <p>Se non viene specificato un server di controllo Capture, questo parametro fa riferimento per impostazione predefinita al valore della variabile d'ambiente <i>DB2DBDFT</i>.</p>
capture_schema=schema	<p>Specifica il nome dello schema Capture utilizzato per identificare un particolare programma Capture. È necessario che il nome dello schema abbia un numero di caratteri compreso fra 1 e 30. Il valore predefinito è ASN.</p>

Tabella 31. Definizioni per i parametri di richiamo `asncmd` (Continua)

Parametri	Definizione
<b>chgparms</b>	<p>Specifica la modifica di uno o più parametri di funzionamento di un programma Capture mentre è in esecuzione:</p> <ul style="list-style-type: none"> <li>• <b>autostop</b></li> <li>• <b>commit_interval</b></li> <li>• <b>logreuse</b></li> <li>• <b>logstdout</b></li> <li>• <b>memory_limit</b></li> <li>• <b>monitor_interval</b></li> <li>• <b>monitor_limit</b></li> <li>• <b>prune_interval</b></li> <li>• <b>retention_limit</b></li> <li>• <b>signal_limit</b></li> <li>• <b>sleep_interval</b></li> <li>• <b>term</b></li> <li>• <b>trace_limit</b></li> </ul> <p><b>Limitazione:</b> <code>z/OS</code> Il valore di <b>memory_limit</b> non può essere alterato con il programma Capture in esecuzione. Per modificare il valore, per prima cosa è necessario arrestare il programma Capture.</p> <p>È possibile specificare più parametri all'interno di un comando <code>asncmd chgparms</code>, ed è possibile modificare i valori di tali parametri ogni volta che si desidera. Le modifiche sovrascrivono temporaneamente i valori nella tabella <code>IBMSNAP_CAPPARMS</code>, ma non vengono scritte nella tabella. Quando si arresta e si riavvia, il programma Capture esso utilizza i valori in <code>IBMSNAP_CAPPARMS</code>. "asncap: Avvio Capture" a pagina 263 include le descrizioni dei parametri che è possibile sovrascrivere con questo comando.</p>
<b>prune</b>	<p>Specificare questo parametro se si desidera eliminare le tabelle CD (change-data) (CD), UOW (unit-of-work), <code>IBMSNAP_CAPMON</code>, <code>IBMSNAP_CAPTRACE</code> e <code>IBMSNAP_SIGNAL</code> una volta. Il programma Capture immette un messaggio quando il comando viene accodato.</p>
<b>qryparms</b>	<p>Specificare se si desidera che i valori dei parametri di funzionamento correnti siano scritti nell'output standard (stdout).</p>
<b>reinit</b>	<p>Specificare in modo che il programma Capture acquisisca le origini di replica appena aggiunte dalla tabella <code>IBMSNAP_REGISTER</code>. Ad esempio, utilizzare questo parametro se viene aggiunta una nuova origine di replica o se viene utilizzata un'istruzione <code>ALTER ADD</code> per aggiungere una colonna alle tabelle origine di replica e CD (change-data) mentre il programma Capture è in esecuzione.</p>

Tabella 31. Definizioni per i parametri di richiamo `asnccmd` (Continua)

Parametri	Definizione
<b>suspend</b>	Specificare in modo da abbandonare le risorse del sistema operativo in transazioni operative durante i periodi di picco senza distruggere l'ambiente del programma Capture.  <b>Attenzione:</b> Non sospendere Capture per cancellare un'origine di replica. Arrestare invece il programma Capture.
<b>resume</b>	Specificare in modo che un programma Capture sospeso riprenda l'acquisizione dei dati.
<b>status</b>	Specificare in modo da ricevere messaggi che indicano lo stato di ogni thread Capture (gestione, eliminazione, serializzazione e lavoro).
<b>stop</b>	Specificare in modo da arrestare il programma Capture in un modo ordinato ed eseguire il commit dei record di registrazione elaborati fino a quel punto.

## Esempi per `asnccmd`

I seguenti esempi illustrano come utilizzare il comando `asnccmd`.

### Esempio 1

Per abilitare un programma Capture in esecuzione al riconoscimento delle origini di replica appena aggiunte:

```
asnccmd capture_server=db capture_schema=ASN reinit
```

### Esempio 2

Per eliminare le tabelle CD, UOW, IBMSNAP\_CAPMON, IBMSNAP\_CAPTRACE e IBMSNAP\_SIGNAL una volta:

```
asnccmd capture_server=db capture_schema=ASN prune
```

### Esempio 3

Per ricevere i messaggi sullo stato di ogni thread Capture:

```
asnccmd capture_server=db capture_schema=ASN status
```

### Esempio 4

Per inviare i correnti valori operativi di un programma Capture all'output standard:

```
asnccmd capture_server=db capture_schema=ASN qryparms
```

### Esempio 5

Per disabilitare l'eliminazione automatica nel programma Capture in esecuzione:

```
asnccmd capture_server=db capture_schema=ASN chgparms autoprune=n
```

### Esempio 6

Per arrestare un programma Capture in esecuzione:

## asnapply: avvio di Apply

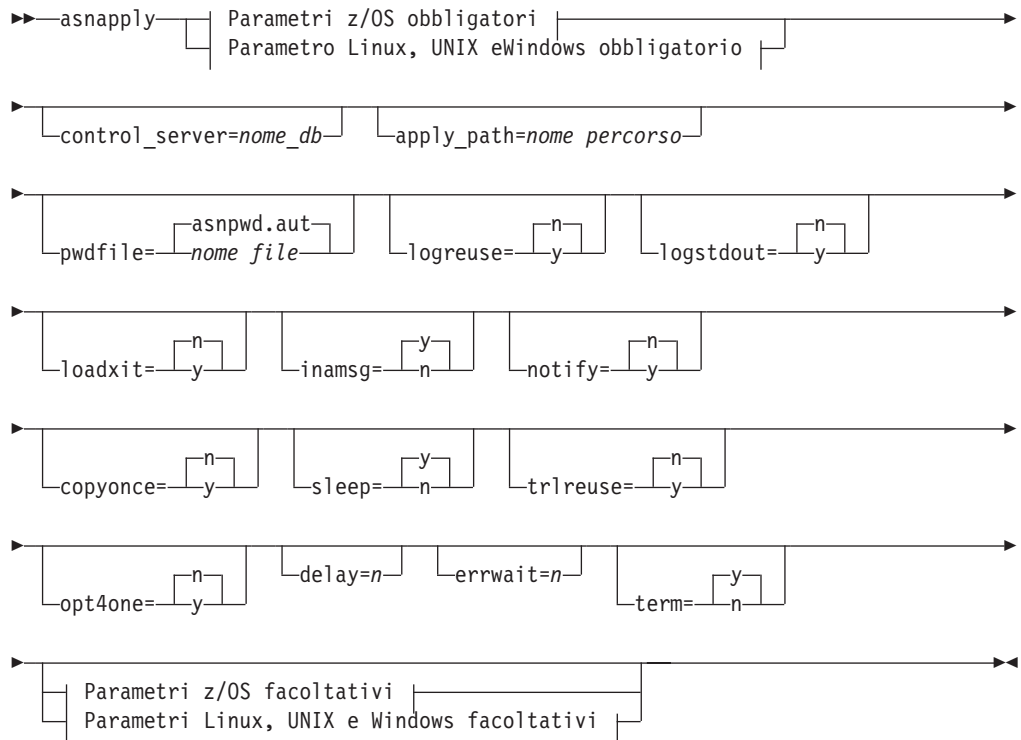
z/OS

Linux UNIX Windows

Utilizzare il comando `asnapply` per avviare il programma Apply su Linux, UNIX, Windows, e UNIX System Services (USS) su z/OS. Eseguire questo comando alla richiesta del sistema operativo o in uno script shell.

Dopo aver avviato il programma Apply, esso rimane in esecuzione fino a quando non viene arrestato o quando rileva un errore non ripristinabile.

### Sintassi



#### Parametri z/OS obbligatori:

|—apply\_qual=qualificatore\_apply—db2\_subsystem=nome—|

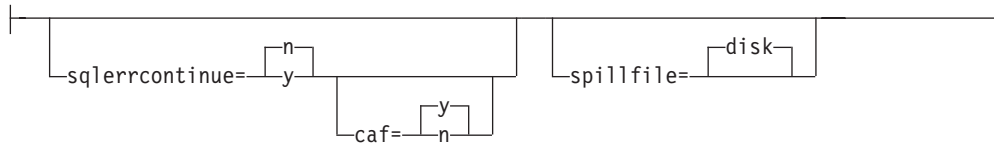
#### Parametro Linux, UNIX e Windows obbligatorio:

|—apply\_qual=qualificatore\_apply—|

#### Parametri z/OS facoltativi:

| spillfile=mem | disk | arm=identifier |

## Parametri Linux, UNIX e Windows facoltativi:



## Parametri

Tabella 32 definisce i parametri di richiamo.

Tabella 32. Definizioni dei parametri di richiamo *asnapply* per i sistemi operativi Linux, UNIX, Windows e z/OS

Parametri	Definizione
<code>apply_qual=qualificatore_apply</code>	<p>Specifica il qualificatore Apply che il programma Apply utilizza per identificare le serie di richieste che saranno disponibili.</p> <p>Il valore che viene inserito deve corrispondere al valore della colonna <code>APPLY_QUAL</code> nella tabella <code>IBMSNAP_SUBS_SET</code>. Il nome del qualificatore Apply è sensibile al maiuscolo/minuscolo e può essere composto da un massimo di 18.</p>
<code>db2_subsystem=nome</code>	<p><b>z/OS</b> Specifica il nome del sottosistema DB2 dove sarà eseguito il programma Apply. Il nome del sottosistema che si immette può avere un massimo di quattro caratteri. Non esiste alcun valore predefinito per questo parametro. Questo parametro è obbligatorio.</p>
<code>control_server=nome_db</code>	<p>Specifica il nome del server di controllo Apply in cui risiedono le definizioni delle richieste e le tabelle di controllo del programma Apply.</p> <p><b>z/OS</b> Specifica il nome dell'ubicazione del server di controllo Apply.</p> <p><b>Linux UNIX Windows</b> Se non viene specificato un server di controllo Apply, questo parametro fa riferimento per impostazione predefinita al valore della variabile d'ambiente <code>DB2DBDFT</code>.</p>
<code>apply_path=nome percorso</code>	<p>Specifica l'ubicazione dei file di lavoro utilizzati dal programma Apply. Il valore predefinito è la directory in cui il comando <b>asnapply</b> è stato richiamato.</p>
<code>pwdfile=nomefile</code>	<p>Specifica il nome del file di password. Se non viene specificato un file di password, il valore predefinito è <code>asnpwd.aut</code>.</p> <p>Questo comando ricerca il file di password nella directory specificata dal parametro <b>apply_path</b>. Se non viene specificato il parametro <b>apply_path</b>, questo comando ricerca il file di password nella directory da cui è stato richiamato il comando.</p>

Tabella 32. Definizioni dei parametri di richiamo *asnapply* per i sistemi operativi Linux, UNIX, Windows e z/OS (Continua)

Parametri	Definizione
<b>logreuse=y/n</b>	<p>Specifica se il programma Apply riutilizza o aggiunge i messaggi al file di log.</p> <p><b>n (predefinito)</b> Il programma Apply aggiunge i messaggi al file di log, anche dopo il riavvio del programma Apply.</p> <p><b>y</b> Il programma Apply riutilizza il file di log, eliminandolo e poi ricreandolo, al momento del riavvio.</p> <p><b>z/OS</b> Il nome del file di log non contiene il nome dell'istanza DB2: <i>control_server.apply_qualifier.APP.log</i>.</p> <p><b>Linux UNIX Windows</b> Il nome del file di log contiene il nome dell'istanza DB2: <i>db2instance.control_server.apply_qualifier.APP.log</i>.</p>
<b>logstdout=y/n</b>	<p>Specifica dove il programma Apply indirizza i messaggi dei file di log:</p> <p><b>n (predefinito)</b> Il programma Apply indirizza la maggior parte dei messaggi dei file di log solo al file di log. I messaggi di inizializzazione vengono memorizzati sia nel file di log che nell'output standard (STDOUT).</p> <p><b>y</b> Il programma Apply indirizza i messaggi dei file di log sia nel file di log che nell'output standard (STDOUT).</p>
<b>loadxit=y/n</b>	<p>Specifica se il programma Apply richiama ASNLOAD. ASNLOAD è una routine di chiusura fornita dall'IBM che utilizza i programmi di utilità per l'esportazione e il caricamento per aggiornare le tabelle di destinazione.</p> <p><b>n (predefinito)</b> Il programma Apply non richiama ASNLOAD.</p> <p><b>y</b> Il programma Apply richiama ASNLOAD.</p>
<b>inamsg=y/n</b>	<p>Specifica se il programma Apply emette un messaggio quando è inattivo.</p> <p><b>y (predefinito)</b> Il programma Apply emette un messaggio quando è inattivo.</p> <p><b>n</b> Il programma Apply non emette un messaggio quando è inattivo.</p>
<b>notify=y/n</b>	<p>Specifica se il programma Apply deve richiamare ASNDONE. ASNDONE è una routine di chiusura che restituisce il controllo all'utente quando il programma Apply termina di copiare una serie di richieste.</p> <p><b>n (predefinito)</b> Il programma Apply non richiama ASNDONE.</p> <p><b>y</b> Il programma Apply richiama ASNDONE.</p>

Tabella 32. Definizioni dei parametri di richiamo *asnapply* per i sistemi operativi Linux, UNIX, Windows e z/OS (Continua)

Parametri	Definizione
<b>copyonce=y/n</b>	<p>Specifica se il programma Apply esegue un ciclo di copia per ogni serie di richieste idonea nel momento in cui il programma Apply viene richiamato. Successivamente, il programma Apply termina. Una serie di richieste idonea subscription set soddisfa i seguenti criteri:</p> <ul style="list-style-type: none"> <li>• (ACTIVATE &gt; 0) nella tabella IBMSNAP_SUBS_SET. Quando il valore della colonna ACTIVATE è maggiore di zero, la serie di richieste è attiva illimitatamente o viene utilizzata solo per l'elaborazione di richieste singola.</li> <li>• (REFRESH_TYPE = R o B) o (REFRESH_TYPE = E e si è verificato l'evento specificato). Il valore della colonna REFRESH_TYPE è memorizzato nella tabella IBMSNAP_SUBS_SET.</li> </ul> <p>Il limite MAX_SYNCH_MINUTES dalla tabella delle serie di richieste e la data/ora END_OF_PERIOD dalla tabella IBMSNAP_SUBS_EVENT sono rispettati se specificati.</p> <p><b>n (predefinito)</b> Il programma Apply non esegue un ciclo di copia per ogni serie di richieste idonea.</p> <p><b>y</b> Il programma Apply esegue un ciclo di copia per ogni serie di richieste idonea.</p>
<b>sleep=y/n</b>	<p>Specifica come deve procedere il programma Apply se non esistono nuove serie di richieste idonee per l'elaborazione.</p> <p><b>y (predefinito)</b> Il programma Apply si disattiva.</p> <p><b>n</b> Il programma Apply si arresta.</p>
<b>trlreuse=y/n</b>	<p>Specifica se il programma Apply svuota la tabella IBMSNAP_APPLYTRAIL all'avvio.</p> <p><b>n (predefinito)</b> Il programma Apply aggiunge voci alla tabella IBMSNAP_APPLYTRAIL. Il programma Apply non svuota la tabella.</p> <p><b>y</b> Il programma Apply svuota la tabella IBMSNAP_APPLYTRAIL durante l'avvio del programma.</p>
<b>opt4one=y/n</b>	<p>Specifica se le prestazioni del programma Apply sono ottimizzate se viene definita solo una serie di richieste per il programma Apply.</p> <p><b>n (predefinito)</b> Le prestazioni del programma Apply non sono ottimizzate per una serie di richieste.</p> <p><b>y</b> Le prestazioni del programma Apply sono ottimizzate per una serie di richieste. Se l'ottimizzazione viene impostata su y, il programma Apply memorizza nella cache e riutilizza le informazioni sui membri della serie di richieste. Questo riutilizzo delle informazione sui membri della serie di richieste riduce l'utilizzo CPU e migliora la velocità di prestazione.</p>



Tabella 32. Definizioni dei parametri di richiamo *asnapply* per i sistemi operativi Linux, UNIX, Windows e z/OS (Continua)

Parametri	Definizione
<b>delay</b> = <i>n</i>	Specifica il tempo di ritardo (in secondi) alla fine di ogni ciclo Apply quando viene utilizzata la replica continua, in cui <i>n</i> =0, 1, 2, 3, 4, 5, o 6. Il valore predefinito è 6 e viene usato durante la replica continua (ovvero, quando la serie di richieste utilizza <i>sleep</i> =0 minuti). Questo parametro viene ignorato se viene specificato <b>copyonce</b> .
<b>errwait</b> = <i>n</i>	Specifica il numero di secondi (da 1 a 65535) che il programma Apply attende prima di effettuare un nuovo tentativo dopo che il programma rileva una condizione di errore. Il valore predefinito è 300 secondi (cinque minuti). <b>Nota:</b> non specificare un numero troppo piccolo, perché il programma Apply rimane in esecuzione a ciclo continuo e crea molte righe nella tabella IBMSNAP_APPLYTRAIL.
<b>term</b> = <i>y/n</i>	Specifica se il programma Apply continua ad essere eseguito se non riesce a connettersi al proprio server di controllo. <b>y (predefinito)</b> Per impostazione predefinita, il programma Apply termina se non riesce a connettersi al proprio server di controllo. <b>n</b> Il programma Apply non termina. Apply invece registra un errore, attende per la quantità di tempo impostata dal parametro <b>errwait</b> , quindi ritenta la connessione.  Questo parametro viene ignorato se viene specificato <b>copyonce</b> .
<b>spillfile</b> = <i>filetype</i>	Specifica dove è memorizzata la serie di risposte estratte.  <b>z/OS</b> I valori validi sono:  <b>mem (predefinito)</b> Un file di memoria. Il programma Apply ha esito negativo se la memoria è insufficiente per la serie di risposte. <b>disk</b> Un file di disco.  <b>Linux UNIX Windows</b> I valori validi sono:  <b>disk (predefinito)</b> Un file di disco.
<b>arm</b> = <i>identifier</i>	<b>z/OS</b> Specifica una stringa alfanumerica di tre caratteri utilizzata per identificare una singola istanza del programma Apply all'Automatic Restart Manager. Il valore fornito viene aggiunto al nome dell'elemento ARM che Apply genera per se stesso: ASNTAxxxxyyyy (dove <i>xxxx</i> è il nome allegato del gruppo di condivisione dati e <i>yyyy</i> è il nome del membro DB2). È possibile specificare qualsiasi lunghezza di stringa per il parametro <b>arm</b> , ma il programma Apply concatenerà al nome attuale solo un massimo di tre caratteri. Se necessario, il programma Apply aggiungerà spazi bianchi al nome per renderlo un nome univoco di 16 byte.

Tabella 32. Definizioni dei parametri di richiamo `asnapply` per i sistemi operativi Linux, UNIX, Windows e z/OS (Continua)

Parametri	Definizione
<code>caf=y/n</code>	<p><b>z/OS</b> Specifica se il programma Apply viene eseguito con RRS (Recoverable Resource Manager Services) di connessione (CAF=n). L'opzione del parametro di runtime CAF (Call Attach Facility) <code>caf =y</code> specifica se il programma di replica sovrascrive l'RRS di connessione e viene eseguito con CAF di connessione. L'opzione <code>caf =y</code> è l'impostazione predefinita per il programma Apply.</p> <p><b>y (predefinito)</b> Specifica che il programma Apply viene eseguito con il CAF di connessione.</p> <p><b>n</b> Il programma Apply utilizza l'RRS (Recoverable Resource Manager Services) di connessione (<code>caf =n</code>).</p>
<code>sqlerrcontinue=y/n</code>	<p>Specifica se il programma Apply continua l'elaborazione quando rileva alcuni errori SQL.</p> <p>Il programma Apply verifica SQLSTATE con esito negativo con i valori specificati nel file SQLSTATE file, che è stato creato prima dell'esecuzione del programma Apply. Se viene rilevata una corrispondenza, il programma Apply scrive le informazioni sulle righe con esito negativo in un file di errore (<code>apply_qualifier.ERR</code>) e continua l'elaborazione. Il file SQLSTATE può contenere più di 20 valori a cinque byte.</p> <p><b>n (predefinito)</b> Il programma Apply non verifica il file SQLSTATE.</p> <p><b>y</b> Il programma Apply verifica il file SQLSTATE durante l'elaborazione.</p>

## Codici di restituzione

Il comando `asnapply` restituisce un codice di restituzione zero in caso di completamento con esito positivo. In caso di comando con esito negativo, viene restituito un codice di restituzione diverso da zero.

## Esempi per `asnapply`

I seguenti esempi illustrano come utilizzare il comando `asnapply`.

### Esempio 1

Per avviare un programma Apply con un qualificatore Apply denominato AQ1, un server di controllo denominato dbx con file di lavoro ubicati nella directory `/home/files/apply/`:

```
asnapply apply_qual=AQ1 control_server=dbx apply_path=/home/files/apply/
  pwdfile=pass1.txt
```

Il programma Apply ricerca la directory `/home/files/apply/` per il file di password denominato `pass1.txt`.

### Esempio 2



Tabella 33. Definizioni dei parametri di richiamo `asnacmd` per i sistemi operativi Linux, UNIX, Windows e z/OS (Continua)

Parametri	Definizione
<code>control_server=nome_db</code>	<p>Specifica il nome del server di controllo Apply in cui risiedono le definizioni delle richieste e le tabelle di controllo Apply.</p> <p><b>z/OS</b> Il parametro del server di controllo è il nome del server di database che si collega al server di controllo.</p> <p><b>Linux UNIX Windows</b> Se non viene specificato un server di controllo Apply, questo parametro fa riferimento per impostazione predefinita al valore della variabile d'ambiente DB2DBDFT.</p>
<code>status</code>	Specificare di ricevere messaggi che indicano lo stato di ogni thread (gestione e lavoro) in Apply.
<code>stop</code>	Specificare di arrestare il programma Apply in un modo ordinato.

## Esempi per `asnacmd`

Gli esempi seguenti illustrano come utilizzare il comando `asnacmd`.

### Esempio 1

Per ricevere i messaggi sullo stato di ogni thread Apply:

```
asnacmd apply_qual=AQ1 control_server=dbx status
```

### Esempio 2

Per arrestare il programma Apply:

```
asnacmd apply_qual=AQ1 control_server=dbx stop
```

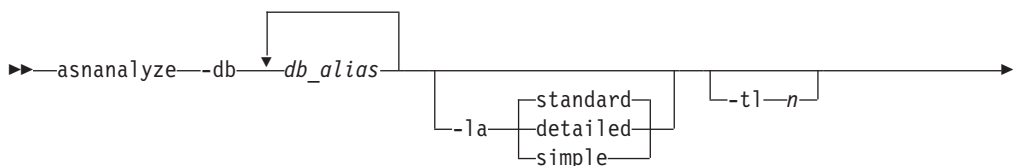
## asnanalyze: Funzionamento di Analyzer

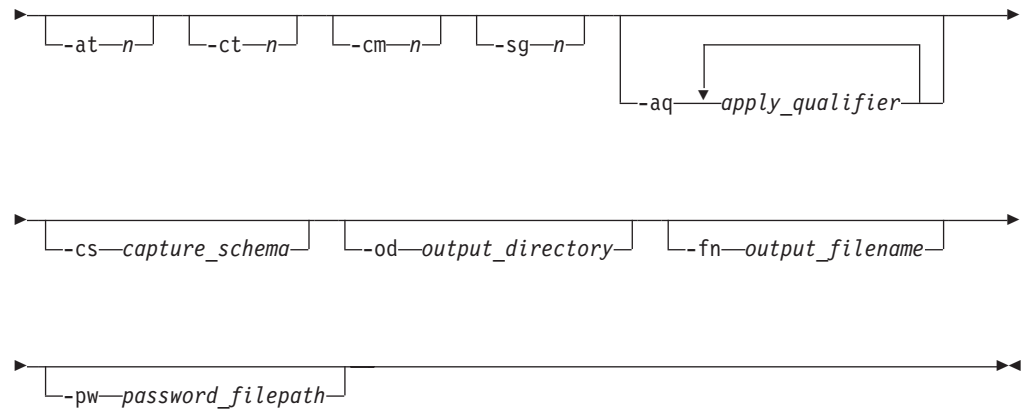
### Linux UNIX Windows

Utilizzare il comando `asnanalyze` per generare prospetti riguardo lo stato di replica delle tabelle di controllo. Questo comando analizza le tabelle di controllo di replica che risiedono su qualsiasi sistema operativo, inclusi i System i; tuttavia, è necessario richiamare il comando da Linux, UNIX o Windows.

Per richiamare il comando è necessario inserire uno spazio tra il comando `asnanalyze` e il primo parametro. Se il comando viene immesso senza alcun parametro, si riceverà l'aiuto comando sullo schermo.

### Sintassi





## Parametri

Tabella 34 definisce i parametri di richiamo.

Tabella 34. Definizioni del parametro di richiamo *asnanalyze* per i sistemi operativi Linux, UNIX and Windows

Parametri	Definizione
<b>-db</b> <i>db_alias</i>	<p>Specifica il server di controllo Capture, il server di destinazione e il server di controllo Apply.</p> <p>È necessario fornire almeno un alias database. Se esiste più di un database alias, utilizzare spazi vuoti per separare i valori.</p>
<b>-la</b> <i>level_of_analysis</i>	<p>Specifica il livello di analisi da notificare:</p> <p><b>standard (predefinito)</b> Crea una notifica che include i contenuti della tabella di controllo e l'informazione di stato dai programmi Capture e Apply.</p> <p><b>dettagliato</b> Crea l'informazione nella notifica standard e:</p> <ul style="list-style-type: none"> <li>• Informazioni sull'eliminazione delle tabelle Change-data (CD) e unit-of-work (UOW)</li> <li>• DB2 per z/OS partizionamento del tablespace e informazione di compressione</li> <li>• Analisi degli indici di destinazione per le chiavi di sottoscrizione</li> </ul> <p><b>semplice</b> Crea le informazioni nel prospetto standard, ma esclude le informazioni di dettaglio dalla tabella IBMSNAP_SUBS_COLS.</p>
<b>-tl</b> <i>n</i>	<p>Specifica l'intervallo di data (da 0 a 30 giorni) delle voci da recuperare dalla tabella IBMSNAP_APPLYTRAIL. Il valore predefinito è 3 giorni.</p>
<b>-at</b> <i>n</i>	<p>Specifica l'intervallo di data (da 0 a 30 giorni) delle voci da recuperare dalla tabella della traccia Apply IBMSNAP_APPLYTRACE. Il valore predefinito è 3 giorni.</p>

Tabella 34. Definizioni del parametro di richiamo *asnanalyze* per i sistemi operativi Linux, UNIX and Windows (Continua)

Parametri	Definizione
<b>-ct</b> <i>n</i>	Specifica l'intervallo di data (da 0 a 30 giorni) di voci da recuperare dalla tabella IBMSNAP_CAPTRACE. Il valore predefinito è 3 giorni.
<b>-cm</b> <i>n</i>	Specifica l'intervallo di data (da 0 a 30 giorni) di voci da recuperare dalla tabella IBMSNAP_CAPMON. Il valore predefinito è 3 giorni.
<b>-sg</b> <i>n</i>	Specifica l'intervallo di data (da 0 a 30 giorni) delle voci da recuperare dalla tabella IBMSNAP_SIGNAL. Il valore predefinito è 3 giorni.
<b>-aq</b> <i>apply_qualifier</i>	<p>Specifica il qualificatore Apply che identifica serie specifiche di sottoscrizione da analizzare.</p> <p>È possibile specificare più di un qualificatore Apply. Se esiste più di un qualificatore Apply, utilizzare spazi vuoti per separare i valori. Se non è stato specificato alcun qualificatore Apply, tutte le serie di richieste per gli alias database specificati vengono analizzate.</p>
<b>-cs</b> <i>capture_schema</i>	<p>Specifica il nome dello schema Capture che si desidera analizzare.</p> <p>Se si utilizza questo parametro, è possibile specificare solo uno schema Capture.</p>
<b>-od</b> <i>output_directory</i>	Specifica la directory nella quale si desidera memorizzare la notifica dell'Analyzer. Il valore predefinito è la directory corrente.
<b>-fn</b> <i>output_filename</i>	<p>Specifica il nome del file che contiene l'output della notifica dell'Analyzer.</p> <p>Utilizzare le convenzioni di denominazione del sistema operativo utilizzato per eseguire l'Analyzer. Se il nome del file è già esistente, il file viene sovrascritto. Il nome predefinito del file è <i>asnanalyze.htm</i>.</p>
<b>-pw</b> <i>password_filepath</i>	Specifica il nome e il percorso del file password. Se non viene specificato questo parametro, l'Analyzer ricerca il file <i>asnpwd.aut</i> all'interno della directory corrente.

## Esempi per *asnanalyze*

Gli esempi seguenti illustrano come utilizzare il comando *asnanalyze*.

### Esempio 1

Per analizzare le tabelle di controllo replica sul database denominato *proddb1*:

```
asnanalyze -db proddb1
```

### Esempio 2

Per ottenere un livello di analisi dettagliato riguardo le tabelle di controllo di replica sui database *proddb1* e *proddb2*:

```
asnanalyze -db proddb1 proddb2 -la detailed
```

### Esempio 3

Per analizzare gli ultimi due giorni di informazione dalle tabelle IBMSNAP\_APPLYTRAIL, IBMSNAP\_APPLYTRACE, IBMSNAP\_CAPTRACE, IBMSNAP\_CAPMON e IBMSNAP\_SIGNAL sui database proddb1 e proddb2:

```
asnanalyze -db proddb1 proddb2 -t1 2 -at 2 -ct 2 -cm 2 -sg 2
```

### Esempio 4

Per ottenere un'analisi di livello semplice riguardo gli ultimi due giorni di informazione dalle tabelle IBMSNAP\_APPLYTRAIL, IBMSNAP\_APPLYTRACE, IBMSNAP\_CAPTRACE, IBMSNAP\_CAPMON e IBMSNAP\_SIGNAL sui database proddb1 e proddb2 solo per i qualificatori Apply qual1 e qual2:

```
asnanalyze -db proddb1 proddb2 -la simple -t1 2 -at 2 -ct 2 -cm 2 -sg 2  
-aq qual1 qual2 -od c:\mydir -fn anzout -pw c:\SQLLIB
```

Questo esempio di comando scrive l'output dell'analyzer in un file denominato anzout nella directory c:\mydir e utilizza l'informazione password nella directory c:\SQLLIB.

### Esempio 5

Per analizzare uno schema di Capture particolare:

```
asnanalyze -db proddb1 proddb2 -cs BSN
```

### Esempio 6

Per visualizzare l'aiuto riga di comando:

```
asnanalyze
```

---

## asnmon: Starting a Replication Alert Monitor

z/OS

Linux UNIX Windows

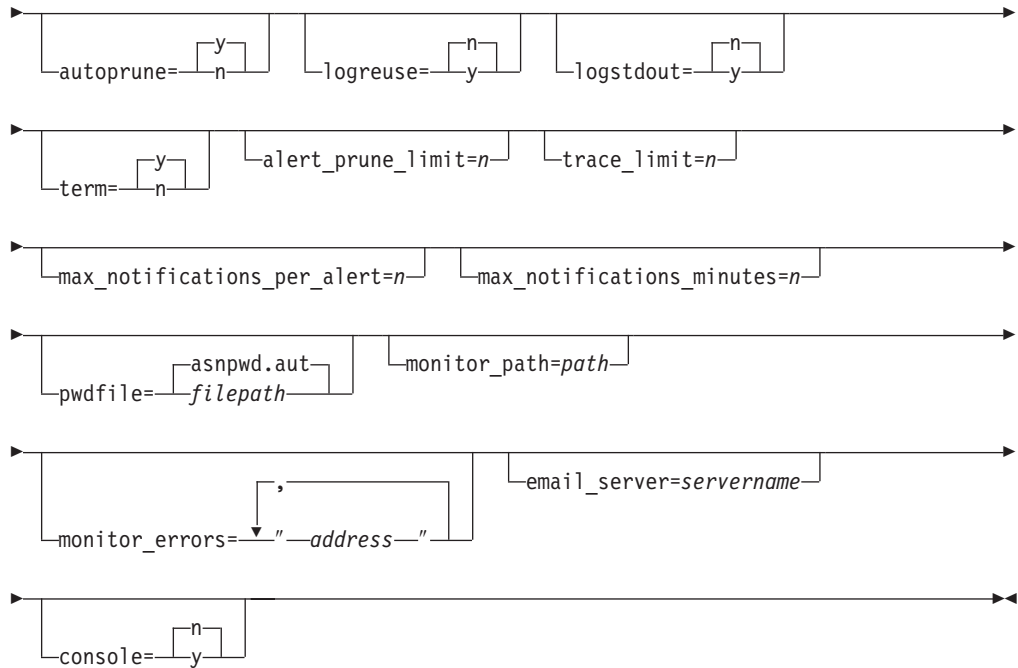
Use the `asnmon` command to start a Replication Alert Monitor on Linux, UNIX, Windows, and UNIX System Services (USS) on z/OS. Run this command at an operating system prompt or in a shell script.

The Replication Alert Monitor records the following information:

- The status of Q Capture, Q Apply, Capture, and Apply programs
- Error messages written to the control tables
- Threshold values

### Syntax

```
▶ asnmon [monitor_server=server] [monitor_qual=mon_qual] [monitor_interval=n] [runonce=y|n] [arm=identifier]
```



## Parameters

Tabella 35 defines the invocation parameters for the asnmon command.

Tabella 35. asnmon invocation parameter definitions for Linux, UNIX, Windows, and z/OS operating systems

Parameter	Definition
<b>monitor_server=server</b>	<p>Specifies the name of the Monitor control server where the Replication Alert Monitor program runs and the monitor control tables reside. This must be the first parameter if entered.</p> <p><b>Linux UNIX Windows</b> If you do not specify a Monitor control server, this parameter defaults to the value from the DB2DBDFT environment variable.</p> <p><b>z/OS</b> The default is DSN.</p>
<b>monitor_qual=mon_qual</b>	<p>Specifies the monitor qualifier that the Replication Alert Monitor program uses. The monitor qualifier identifies the server to be monitored and the associated monitoring conditions.</p> <p>You must specify a monitor qualifier. The monitor qualifier name is case sensitive and can be a maximum of 18 characters.</p>



Tabella 35. *asnmon* invocation parameter definitions for Linux, UNIX, Windows, and z/OS operating systems (Continua)

Parameter	Definition
<b>monitor_interval</b> = <i>n</i>	<p>Specifies how frequently (in seconds) the Replication Alert Monitor program runs for this monitor qualifier. The default is 300 seconds (five minutes).</p> <p>This parameter is ignored by the Replication Alert Monitor if you set the <b>runonce</b> parameter to <i>y</i>.  <b>Important:</b> This <b>monitor_interval</b> parameter affects the Replication Alert Monitor program only. This parameter does not affect Q Capture, Q Apply, Capture, and Apply programs.</p>
<b>runonce</b> = <i>y/n</i>	<p>Specifies whether the Replication Alert Monitor program runs only one time for this monitor qualifier.</p> <p><b>n (default)</b>  The Replication Alert Monitor program runs at the frequency indicated by the <b>monitor_interval</b> parameter.</p> <p><b>y</b>  The Replication Alert Monitor program runs only one monitor cycle.</p> <p>If you set the <b>runonce</b> parameter to <i>y</i>, the <b>monitor_interval</b> parameter is ignored by the Replication Alert Monitor.</p>
<b>arm</b> = <i>identifier</i>	<p><b>z/OS</b> Specifies a three-character alphanumeric string that is used to identify a single instance of the Replication Alert Monitor program to the Automatic Restart Manager. The value that you supply is appended to the ARM element name that the monitor program generates for itself: <i>ASNAMxxxxyyyy</i> (where <i>xxx</i> is the data-sharing group attach name, and <i>yyy</i> is the DB2 member name). You can specify any length of string for the <b>arm</b> parameter, but the monitor program will concatenate only up to three characters to the current name. If necessary, the monitor program will pad the name with blanks to make a unique 16-byte name.</p>
<b>autoprune</b> = <i>y/n</i>	<p>Specifies whether automatic pruning of the rows in the Replication Alert Monitor alerts (IBMSNAP_ALERTS) table is enabled.</p> <p><b>y (default)</b>  The Replication Alert Monitor program automatically prunes the rows in the IBMSNAP_ALERTS table that are older than the value of the <b>alert_prune_limit</b> parameter.</p> <p><b>n</b>  Automatic pruning is disabled.</p>

Tabella 35. *asnmon* invocation parameter definitions for Linux, UNIX, Windows, and z/OS operating systems (Continua)

Parameter	Definition
<b>logreuse=y/n</b>	<p>Specifies whether the Replication Alert Monitor program reuses or appends messages to its diagnostic log file ( db2instance.monitor_server.mon_qual.MON.log ).</p> <p><b>n (default)</b> The Replication Alert Monitor program appends messages to the log file.</p> <p><b>y</b> The Replication Alert Monitor program reuses the log file by deleting it and then recreating it when the Replication Alert Monitor program is restarted.</p>
<b>logstdout=y/n</b>	<p>Specifies where messages are sent by the Replication Alert Monitor program.</p> <p><b>n (default)</b> The Replication Alert Monitor program sends messages to the log file only.</p> <p><b>y</b> The Replication Alert Monitor program sends messages to both the log file and the standard output (stdout).</p>
<b>term=y/n</b>	<p>Specifies whether a monitor program keeps running when DB2 is quiesced.</p> <p><b>y (default)</b> The monitor program stops when DB2 is quiesced.</p> <p><b>n</b> The monitor program keeps running while DB2 is in quiesce mode and has forced all applications to disconnect (including the monitor program). When DB2 is taken out of quiesce mode, the monitor program goes back to monitoring replication.</p> <p>Regardless of the setting for the <b>term</b> parameter, a monitor program stops when DB2 shuts down. When DB2 starts again, you must restart the monitor program.</p>
<b>alert_prune_limit=n</b>	<p>Specifies how long (in minutes) rows are kept in the Replication Alert Monitor alerts (IBMSNAP_ALERTS) table. Any rows older than this value are pruned. The default is 10 080 minutes (seven days).</p>
<b>trace_limit=n</b>	<p>Specifies how long (in minutes) a row can remain in the Replication Alert Monitor trace (IBMSNAP_MONTRACE) table before it becomes eligible for pruning. All IBMSNAP_MONTRACE rows that are older than the value of this <b>trace_limit</b> parameter are pruned at the next pruning cycle. The default is 10 080 minutes (seven days).</p>
<b>max_notifications_per_alert=n</b>	<p>Specifies the maximum number of the same alerts that are sent to a user when the alerts occurred during the time period specified by the <b>max_notifications_minutes</b> parameter value. Use this parameter to avoid re-sending the same alerts to a user. The default is 3.</p>

Tabella 35. *asnmon* invocation parameter definitions for Linux, UNIX, Windows, and z/OS operating systems (Continua)

Parameter	Definition
<b>max_notifications_minutes</b> = <i>n</i>	This parameter works with the <b>max_notifications_per_alert</b> parameter to indicate the time period when alert conditions occurred. The default is 60 minutes.
<b>pwdfile</b> = <i>filepath</i>	Specifies the fully qualified name of the password file. You define this file by using the <code>asnpwd</code> command. The default file name is <code>asnpwd.aut</code> .
<b>monitor_path</b> = <i>path</i>	Specifies the location of the log files used by the Replication Alert Monitor program. The default is the directory where the <code>asnmon</code> command was invoked.
<b>monitor_errors</b> = <i>address</i>	<p>Specifies the e-mail address to which notifications are sent if a fatal error is detected before the alert monitor connects to the Monitor control server. Use this parameter to send a notification that the Monitor control server connection failed because of invalid start parameters, an incorrect monitor qualifier, a down database, or other error.</p> <p>Type double quotation marks around the e-mail address text.</p> <p>You can enter multiple e-mail addresses. Separate the e-mail addresses with commas. You can type spaces before or after the commas.</p>
<b>email_server</b> = <i>servername</i>	Specifies the e-mail server address. Enter this parameter <i>only</i> if you use the <code>ASNMAIL</code> exit routine with SMTP (Simple Mail Transfer Protocol).
<b>console</b> = <i>y/n</i>	<p><b>z/OS</b> Specifies whether the Replication Alert Monitor program sends alert notifications to the z/OS console. If you set this parameter to Y (yes) and an e-mail server was already configured, alerts are sent to both the z/OS console and the e-mail server.</p> <p><b>n (default)</b> The Replication Alert Monitor program does not send alert notifications to the z/OS console.</p> <p><b>y</b> The Replication Alert Monitor program sends alert notifications to the z/OS console.</p>

## Return codes

The `asnmon` command returns a zero return code upon successful completion. A nonzero return code is returned if the command is unsuccessful.

## Examples for `asnmon`

The following examples illustrate how to use the `asnmon` command.

### Example 1

To start the Replication Alert Monitor with the default parameters:

```
asnmon monitor_server=wsdb monitor_qual=monqual
```

### Example 2

To start a Replication Alert Monitor that runs every 120 seconds (two minutes) for the specified monitor qualifier:

```
asnmn monitor_server=wsdb monitor_qual=monqual monitor_interval=120
```

### Example 3

To start a Replication Alert Monitor and specify that it run only once for the specified monitor qualifier:

```
asnmn monitor_server=wsdb monitor_qual=monqual runonce=y
```

### Example 4

To start a Replication Alert Monitor that sends e-mail notifications if it detects monitoring errors:

```
asnmn monitor_server=wsdb monitor_qual=monqual  
monitor_errors="repladm@company.com, dbadmin@company.com"
```

### Example 5

To start a Replication Alert Monitor that runs every 120 seconds (two minutes) and waits 1440 minutes (24 hours) before sending alerts:

```
asnmn monitor_server=wsdb monitor_qual=monqual monitor_interval=120  
max_notifications_per_alert=2 max_notifications_minutes=1440
```

This Replication Alert Monitor program sends a maximum of two alerts when the alerts occurred during the time period specified by the **max\_notifications\_minutes** parameter value (1440 minutes).

---

## asnmcmd: Working with a running Replication Alert Monitor

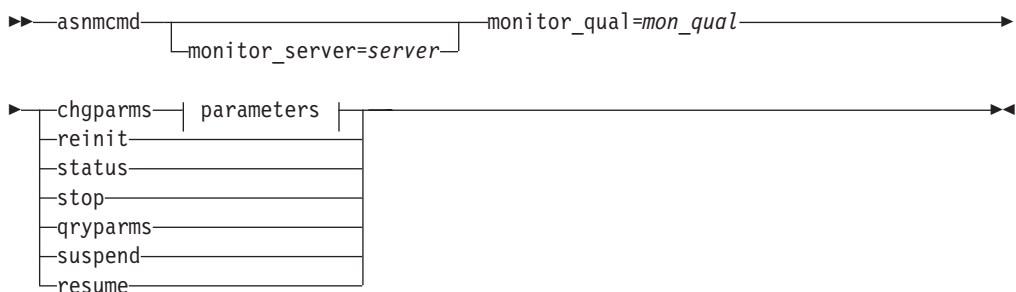
z/OS

Linux UNIX Windows

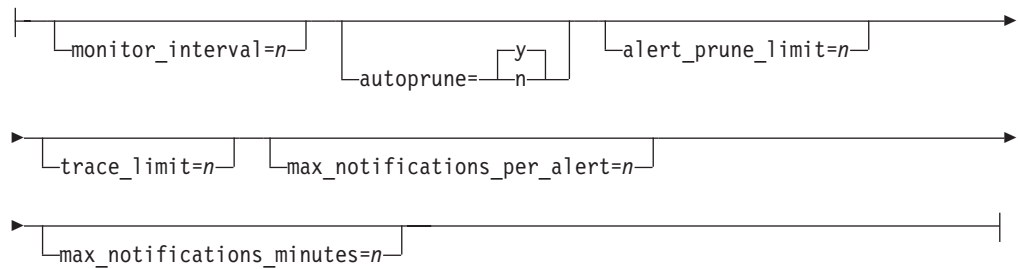
Use `asnmcmd` to send commands to a running Replication Alert Monitor on Linux, UNIX, Windows, and UNIX System Services (USS) on z/OS. Run this command at an operating system prompt or in a shell script.

For information on using the MVS `MODIFY` command to send commands to a running Replication Alert Monitor program on z/OS, see [Working with running SQL replication programs by using the MVS `MODIFY` command](#).

### Syntax



## Parameters:



## Parameters

Tabella 36 defines the invocation parameters for the asnmcmd command.

Tabella 36. *asnmcmd* invocation parameter definitions for Linux, UNIX, Windows, and z/OS operating systems

Parameter	Definition
<b>monitor_server</b> = <i>server</i>	<p>Specifies the name of the Monitor control server where the Replication Alert Monitor program runs and the monitor control tables reside. This must be the first parameter if entered.</p> <p><b>Linux UNIX Windows</b> If you do not specify a Monitor control server, this parameter defaults to the value from the DB2DBDFT environment variable.</p> <p><b>z/OS</b> The default is DSN.</p>
<b>monitor_qual</b> = <i>mon_qual</i>	<p>Specifies the monitor qualifier that the Replication Alert Monitor program uses. The monitor qualifier identifies the server to be monitored and the associated monitoring conditions.</p> <p>You must specify a monitor qualifier. The monitor qualifier name is case sensitive and can be a maximum of 18 characters.</p>

Tabella 36. *asnmcmd* invocation parameter definitions for Linux, UNIX, Windows, and z/OS operating systems (Continua)

Parameter	Definition
<b>chgparms</b>	<p>Specify to change one or more of the following operational parameters of the Replication Alert Monitor while it is running:</p> <ul style="list-style-type: none"> <li>• <b>monitor_interval</b></li> <li>• <b>autoprun</b></li> <li>• <b>alert_prune_limit</b></li> <li>• <b>trace_limit</b></li> <li>• <b>max_notifications_per_alert</b></li> <li>• <b>max_notifications_minutes</b></li> </ul> <p>You can specify multiple parameters in one <code>chgparms</code> subcommand, and you can change these parameter values as often as you want. The changes temporarily override the values in the <code>IBMSNAP_MONPARMS</code> table, but they are not saved in the table. When you stop and restart the Replication Alert Monitor, it uses the values in <code>IBMSNAP_MONPARMS</code>. “<i>asnmcmd: Starting a Replication Alert Monitor</i>” a pagina 285 includes descriptions of the parameters that you can override with this subcommand. <b>Important:</b> The parameter that you are changing must immediately follow the <code>chgparms</code> subcommand.</p>
<b>reinit</b>	<p>Specify to have the Replication Alert Monitor program read its control tables to refresh the data that it has for contacts, alert conditions, and parameters in its memory. When all values are read, the Monitor program begins its cycle of checking conditions on the servers. After this cycle is complete, the next monitor cycle begins after the time specified in <b>monitor_interval</b> has elapsed.</p>
<b>status</b>	<p>Specify to receive messages that indicate the state of each thread (administration, serialization, and worker) in the Replication Alert Monitor.</p>
<b>qryparms</b>	<p>Specify if you want the current operational parameter values for the Replication Alert Monitor written to the standard output (stdout).</p>
<b>suspend</b>	<p>Specify if you want the Replication Alert Monitor to stop checking conditions on servers temporarily until you issue the resume command.</p>
<b>resume</b>	<p>Specify if the Replication Alert Monitor has been suspended and you want the Monitor program to begin checking conditions on servers again.</p>
<b>stop</b>	<p>Specify to stop the Replication Alert Monitor in an orderly way.</p>

## Examples for *asnmcmd*

The following examples illustrate how to use the *asnmcmd* command.

### Example 1

To stop the Replication Alert Monitor for the specified monitor qualifier:

```
asnmcmd monitor_server=wsdb monitor_qual=monqual stop
```

### Example 2

To receive messages that indicate the status of the Replication Alert Monitor threads:

```
asnmcmd monitor_server=wsdb monitor_qual=monqual status
```

### Example 3

To refresh the Replication Alert Monitor with current values from the monitor control tables:

```
asnmcmd monitor_server=wsdb monitor_qual=monqual reinit
```

### Example 4

To reduce the maximum number of notifications that the Replication Alert Monitor sends during a specified time period from the default of 3:

```
asnmcmd monitor_server=wsdb monitor_qual=monqual  
chgparms max_notifications_per_alert=2
```

### Example 5

To send the current operational parameters of the Replication Alert Monitor to the standard output:

```
asnmcmd monitor_server=wsdb monitor_qual=monqual qryparms
```

---

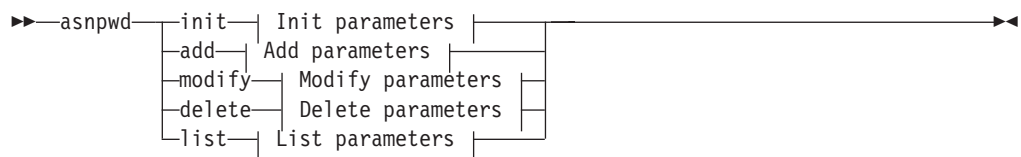
## asnpwd: Creating and maintaining password files

Linux UNIX Windows

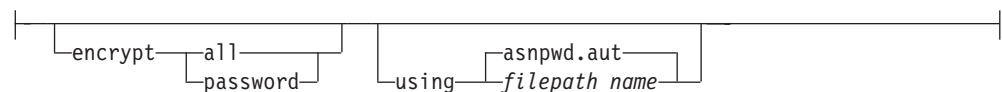
Use the `asnpwd` command to create and change password files on Linux, UNIX, and Windows. Run this command at the command line or in a shell script.

Command help appears if you enter the `asnpwd` command without any parameters, followed by a `?`, or followed by incorrect parameters.

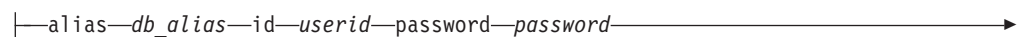
### Syntax

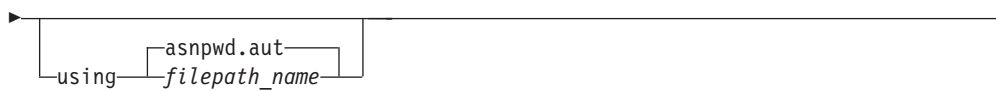


### Init parameters:

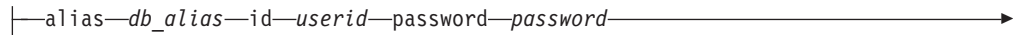


### Add parameters:

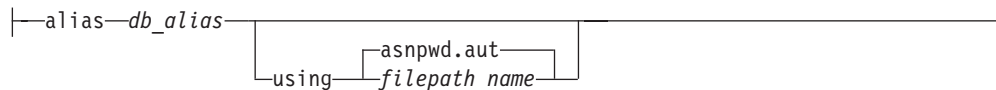




**Modify parameters:**



**Delete parameters:**



**List parameters:**



**Parameters**

Tabella 37 defines the invocation parameters for the asnpwd command.

**Important note about compatibility of password files:** Password files that are created by the asnpwd command starting with Version 9.5 Fix Pack 2 use a new encryption method and cannot be read by older versions of the replication programs and utilities. If you share a password file among programs and utilities that are at mixed level, with some older than these fix packs, do not recreate the password file by using an asnpwd utility that is at these fix packs or newer. Replication programs and utilities at these fix packs or newer can continue to work with older password files. Also, you cannot change an older password file to use the new encryption method; you must create a new password file.

**Usage note:** On 64-bit Windows operating systems, the ADD, MODIFY, DELETE, and LIST options are not supported for password files that were created by using the asnpwd command before Version 9.5 Fix Pack 2.

*Tabella 37. asnpwd invocation parameter definitions for Linux, UNIX, and Windows operating systems*

Parameter	Definition
init	Specify to create an empty password file. This command will fail if you specify the <b>init</b> parameter with a password file that already exists.



Tabella 37. *asnpwd* invocation parameter definitions for Linux, UNIX, and Windows operating systems (Continua)

Parameter	Definition
<b>add</b>	Specify to add an entry to the password file. There can only be one entry in the password file per <i>db_alias</i> . This command will fail if you specify the <b>add</b> parameter with an entry that already exists in the password file. Use the <b>modify</b> parameter to change an existing entry in the password file.
<b>modify</b>	Specify to modify the password or user ID for an entry in the password file.
<b>delete</b>	Specify to delete an entry from the password file.
<b>list</b>	Specify to list the aliases and user ID entries in a password file. This parameter can be used only if the password file was created by using the <b>encrypt password</b> parameter. Passwords are never displayed by the <b>list</b> command.
<b>encrypt</b>	Specifies which entries in a file to encrypt. <p><b>all (default)</b> Encrypt all entries in the specified file such that you cannot list the database aliases, user names, and passwords that are in the file. This option reduces the exposure of information in password files.</p> <p><b>password</b> Encrypt the password entry in the specified file. This option allows users to list the database aliases and user names stored in their password file. Passwords can never be displayed.</p>
<b>using</b> <i>filepath</i>	Specifies the path and name of the password file. Follow the file naming conventions of your operating system. An example of a valid password file on Windows is C:\sqllib\mypwd.aut. <p>If you specify the path and name of the password file, the path and the password file must already exist. If you are using the <b>init</b> parameter and you specify the path and name of the password file, the path must already exist and the command will create the password file for you.</p> <p>If you do not specify this parameter, the default file name is <i>asnpwd.aut</i> and the default file path is the current directory.</p>
<b>alias</b> <i>db_alias</i>	Specifies the alias of the database to which the user ID has access. The alias is always folded to uppercase, regardless of how it is entered.
<b>id</b> <i>userid</i>	Specifies the user ID that has access to the database.
<b>password</b> <i>password</i>	Specifies the password for the specified user ID. This password is case sensitive and is encrypted in the password file.

## Return Codes

The *asnpwd* command returns a zero return code upon successful completion. A nonzero return code is returned if the command is unsuccessful.

## Examples for asnpwd

The following examples illustrate how to use the asnpwd command.

### Example 1

To create a password file with the default name of asnpwd.aut in the current directory:

```
asnpwd INIT
```

### Example 2

To create a password file named pass1.aut in the c:\myfiles directory:

```
asnpwd INIT USING c:\myfiles\pass1.aut
```

### Example 3

To create a password file named mypwd.aut with the encrypt all parameter:

```
asnpwd INIT ENCRYPT ALL USING mypwd.aut
```

### Example 4

To create a password file named mypwd.aut with the encrypt password parameter:

```
asnpwd INIT ENCRYPT PASSWORD USING mypwd.aut
```

### Example 5

To create a default password file with the encrypt password parameter:

```
asnpwd INIT ENCRYPT PASSWORD
```

### Example 6

To add a user ID called oneuser and its password to the password file named pass1.aut in the c:\myfiles directory and to grant this user ID access to the db1 database:

```
asnpwd ADD ALIAS db1 ID oneuser PASSWORD mypwd using c:\myfiles\pass1.aut
```

### Example 7

To modify the user ID or password of an entry in the password file named pass1.aut in the c:\myfiles directory:

```
asnpwd MODIFY Alias sample ID chglocalid PASSWORD chgmajorpwd  
USING c:\myfiles\pass1.aut
```

### Example 8

To delete the database alias called sample from the password file named pass1.aut in the c:\myfiles directory:

```
asnpwd delete alias sample USING c:\myfiles\pass1.aut
```

### Example 9

To see command help:

```
asnpwd
```

### Example 10

To list the entries in a default password file:

```
asnpwd LIST
```

### Example 11

To list the entries in a password file named pass1.aut:

```
asnpwd LIST USING pass1.aut
```

The output from this command depends on how the password file was initialized:

- If it was initialized by using the **encrypt all** parameter, the following message is issued:

```
ASN1986E "Asnpwd" : "". The password file "pass1.aut" contains encrypted information that cannot be listed.
```

- If it was not initialized by using the **encrypt all** parameter, the following details are listed:

```
asnpwd LIST USING pass1.aut  
Alias: SAMPLE ID: chglocalid  
Number of Entries: 1
```

---

## asnsct: Creating a replication service

### Windows

Use the asnsct command to create a replication service in the Windows Service Control Manager (SCM) and invoke the asnqcap, asnqapp, asnmon, asncap, and asnapply commands. Run the asnsct command on the Windows operating system.

### Syntax

```
▶▶ asnsct [-QC] db2_instance [-account] [-password] [asnqcap_command | asnqapp_command | asnmon_command | asncap_command | asnapply_command]
```

### Parameters

Tabella 38 defines the invocation parameters for the asnsct command.

Tabella 38. asnsct invocation parameter definitions for Windows operating systems

Parameter	Definition
-QC	Specifies that you are starting a Q Capture program.
-QA	Specifies that you are starting a Q Apply program.
-M	Specifies that you are starting a Replication Alert Monitor program.
-C	Specifies that you are starting a Capture program.
-A	Specifies that you are starting an Apply program.
db2_instance	Specifies the DB2 instance used to identify a unique DB2 replication service. The DB2 instance can be a maximum of eight characters.

Tabella 38. *asnsrct invocation parameter definitions for Windows operating systems (Continua)*

Parameter	Definition
<b>account</b>	Specifies the account name that you use to log on to Windows. If the account is local it must begin with a period and a backslash (.\ <i>).</i> Otherwise the domain or machine name must be specified (for example, <i>domain_name\account_name</i> ).
<b>password</b>	Specifies the password used with the account name. If the password contains special characters, type a backslash (\) before each special character.
<b>asnsrct_command</b>	<p>Specifies the complete <i>asnsrct</i> command to start a Q capture program. Use the documented <i>asnsrct</i> command syntax with the appropriate <i>asnsrct</i> parameters.</p> <p>If the <i>DB2PATH</i> environment variable is not defined, you must specify a location for the work files by including the <b>capture_path</b> parameter with the <i>asnsrct</i> command. If the <i>DB2PATH</i> variable is defined and you specify a <b>capture_path</b>, the <b>capture_path</b> parameter overrides the <i>DB2PATH</i> variable.</p> <p>The <i>asnsrct</i> command does not validate the syntax of the <i>asnsrct</i> parameters that you enter.</p>
<b>asnsrct_command</b>	<p>Specifies the complete <i>asnsrct</i> command to start a Q apply program. Use the documented <i>asnsrct</i> command syntax with the appropriate <i>asnsrct</i> parameters.</p> <p>If the <i>DB2PATH</i> environment variable is not defined, you must specify the location for the work files by including the <b>apply_path</b> parameter with the <i>asnsrct</i> command. If the <i>DB2PATH</i> variable is defined and you specify an <b>apply_path</b>, the <b>apply_path</b> parameter overrides the <i>DB2PATH</i> variable. The <i>asnsrct</i> command does not validate the syntax of the <i>asnsrct</i> parameters that you enter.</p>
<b>asnmmon_command</b>	<p>Specifies the complete <i>asnmmon</i> command to start a Replication Alert Monitor program. Use the documented <i>asnmmon</i> command syntax with the appropriate <i>asnmmon</i> parameters.</p> <p>If the <i>DB2PATH</i> environment variable is not defined, you must specify a location for the log files by including the <b>monitor_path</b> parameter with the <i>asnmmon</i> command. If the <i>DB2PATH</i> variable is defined and you specify a <b>monitor_path</b>, the <b>monitor_path</b> parameter overrides the <i>DB2PATH</i> variable.</p> <p>The <i>asnsrct</i> command does not validate the syntax of the <i>asnmmon</i> parameters that you enter.</p>

Tabella 38. *asnscri* invocation parameter definitions for Windows operating systems (Continua)

Parameter	Definition
<b>asncap_command</b>	<p>Specifies the complete asncap command to start a Capture program. Use the documented asncap command syntax with the appropriate asncap parameters.</p> <p>If the DB2PATH environment variable is not defined, you must specify a location for the work files by including the <b>capture_path</b> parameter with the asncap command. If the DB2PATH variable is defined and you specify a <b>capture_path</b>, the <b>capture_path</b> parameter overrides the DB2PATH variable.</p> <p>The asnscri command does not validate the syntax of the asncap parameters that you enter.</p>
<b>asnapply_command</b>	<p>Specifies the complete asnapply command to start an Apply program. Use the documented asnapply command syntax with the appropriate asnapply parameters.</p> <p>If the DB2PATH environment variable is not defined, you must specify the location for the work files by including the <b>apply_path</b> parameter with the asnapply command. If the DB2PATH variable is defined and you specify an <b>apply_path</b>, the <b>apply_path</b> parameter overrides the DB2PATH variable.</p> <p>The asnscri command does not validate the syntax of the asnapply parameters that you enter.</p>

## Examples for asnscri

The following examples illustrate how to use the asnscri command.

### Example 1

To create a DB2 replication service that invokes a Q Apply program under a DB2 instance called inst2 and uses a logon account of .\joesmith and a password of my\$pwd:

```
asnscri -QA inst2 .\joesmith my\my$pwd asnqapp apply_server=mydb2 apply_schema =as2
  apply_path=X:\sql11b
```

### Example 2

To create a DB2 replication service that invokes a Capture program under a DB2 instance called inst1:

```
asnscri -C inst1 .\joesmith password asncap capture_server=sampled
  capture_schema=ASN capture_path=X:\logfiles
```

### Example 3

To create a DB2 replication service that invokes an Apply program under a DB2 instance called inst2 and uses a logon account of .\joesmith and a password of my\$pwd:

```
asnscri -A inst2 .\joesmith my\my$pwd asnapply control_server=db2 apply_qual=aq2
  apply_path=X:\sql11b
```

### Example 4

To create a DB2 replication service that invokes a Replication Alert Monitor program under a DB2 instance called inst3:

```
asnsrct -M inst3 .\joesmith password asnmon monitor_server=db3 monitor_qual=mq3
monitor_path=X:\logfiles
```

### Example 5

To create a DB2 replication service that invokes a Capture program under a DB2 instance called inst4 and overrides the default work file directory with a fully qualified **capture\_path**:

```
asnsrct -C inst4 .\joesmith password X:\sqllib\bin\asnrcap capture_server=scdb
capture_schema=ASN capture_path=X:\logfiles
```

### Example 6

To create a DB2 replication service that invokes a Q capture program under a DB2 instance called inst1:

```
asnsrct -QC inst1 .\joesmith password asnqcap capture_server=mydb1
capture_schema=QC1 capture_path=X:\logfiles
```

---

## asnsdrop: Dropping a replication service

### Windows

Use the `asnsdrop` command to drop replication services from the Windows Service Control Manager (SCM) on the Windows operating system.

### Syntax

```
asnsdrop service_name
        ALL
```

### Parameters

Tabella 39 defines the invocation parameters for the `asnsdrop` command.

*Tabella 39. asnsdrop invocation parameter definitions for Windows operating systems*

Parameter	Definition
<code>service_name</code>	Specifies the fully qualified name of the DB2 replication service. Enter the Windows SCM to obtain the DB2 replication service name. On Windows operating systems, you can obtain the service name by opening the Properties window of the DB2 replication service.  If the DB2 replication service name contains spaces, enclose the entire service name in double quotation marks.
<code>ALL</code>	Specifies that you want to drop all DB2 replication services.

### Examples for asnsdrop

The following examples illustrate how to use the `asnsdrop` command.

#### Example 1

To drop a DB2 replication service:  
asnsdrop DB2.SAMPLEDB.SAMPLEDB.CAP.ASN

### Example 2

To drop a DB2 replication service with a schema named A S N (with embedded blanks), use double quotation marks around the service name:

```
asnsdrop "DB2.SAMPLEDB.SAMPLEDB.CAP.A S N"
```

### Example 3

To drop all DB2 replication services:

```
asnsdrop ALL
```

---

## asnslist: Listing replication services

### Windows

Use the asnslist command to list replication services in the Windows Service Control Manager (SCM). You can optionally use the command to list details about each service. Run the asnslist command on the Windows operating system.

### Syntax

```
asnslist [DETAILS]
```

### Parameters

Tabella 40 defines the invocation parameter for the asnslist command.

Tabella 40. asnslist invocation parameter definition for Windows operating systems

Parameter	Definition
details	Specifies that you want to list detailed data about all DB2 replication services on a system.

### Examples for asnslist

The following examples illustrate how to use the asnslist command.

#### Example 1

To list the names of DB2 replication services on a system:

```
asnslist
```

Here is an example of the command output:

```
DB2.DB2.SAMPLE.QAPP.ASN  
DB2.DB4.SAMPLE.QCAP.ASN
```

#### Example 2

To list details about all services on a system:

```
asnslist details
```

Here is an example of the command output:

```
DB2.DB2.SAMPLE.QAPP.ASN
Display Name: DB2 DB2 SAMPLE QAPPLY ASN
Image Path:  ASNSERV DB2.DB2.SAMPLE.APP.AQ1 -ASNQAPPLY QAPPLY_SERVER=SAMPLE AP
             PLY_SCHEMA=ASN QAPPLY_PATH=C:\PROGRA~1\SQLLIB
Dependency:  DB2-0

DB2.DB4.SAMPLE.QCAP.ASN
Display Name: DB2 DB4 SAMPLE QAPPLY ASN
Image Path:  ASNSERV DB2.DB4.SAMPLE.APP.AQ1 -ASNQCAP QCAPTURE_SERVER=SAMPLE CA
             PTURE_SCHEMA=ASN QCAPTURE_PATH=C:\PROGRA~1\SQLLIB
Dependency:  DB4-0
```

## asntdiff: Comparing data in source and target tables

z/OS

Linux UNIX Windows

Use the `asntdiff` command to compare a source table with a target table and generate a list of differences between the two. Run the `asntdiff` command on Linux, UNIX, Windows, or z/OS at an operating system prompt or in a shell script.

The `asntdiff` command compares DB2 tables on Linux, UNIX, Windows, z/OS, and System i operating systems.

For information on the `asntdiff -f` command option, which enables you to compare tables that are not part of replication by using an input file, see “`asntdiff -f` (input file) command option” a pagina 307.

### Syntax

```

▶▶ asntdiff—DB=server—DB2_SUBSYSTEM=subsystem—SCHEMA=schema
▶
▶ [DIFF_SCHEMA=difference_table_schema] [DIFF_TABLESPACE=tablespace]
▶ [DIFF_DROP=n—y] [MAXDIFF=difference_limit] WHERE=WHERE_clause
▶ [DIFF_PATH=log_path] [PWDFILE=filename] [DIFF=table_name]
▶ [RANGECOL=range_clause_option] [SQLID=authorization_ID]

```

#### range\_clause\_option:

```

| src_colname—FROM:date-time_lower-bound—TO:date-time_upper-bound
| src_colname—FROM:date-time
| src_colname—TO:date-time

```

### Parameters

Tabella 41 a pagina 303 defines the invocation parameters for the `asntdiff` command.



Tabella 41. *asntdiff* invocation parameter definitions for Linux, UNIX, Windows, and z/OS operating systems

Parameter	Definition
<b>DB=server</b>	<p>Specifies the DB2 alias of the database that stores information about the source and target tables that will be compared. The value differs depending on whether you are using Q replication or SQL replication:</p> <p><b>Q replication</b> The name of the Q Capture server, which contains the IBMQREP_SUBS table.</p> <p><b>z/OS</b> The location name of the Q Capture server, which contains the IBMQREP_SUBS table.</p> <p><b>SQL replication</b> The name of the Apply control server, which contains the IBMSNAP_SUBS_MEMBR table.</p> <p><b>z/OS</b> The location name of the Apply control server, which contains the IBMSNAP_SUBS_MEMBR table.</p>
<b>DB2_SUBSYSTEM=Subsystem</b>	<b>z/OS</b> Specifies the name of the subsystem where you run the <i>asntdiff</i> utility.
<b>SCHEMA=Schema</b>	Specifies the schema of the Q Capture control tables for Q replication, or the schema of the Apply control tables for SQL replication. The default is ASN.
<b>DIFF_SCHEMA=</b> <i>difference_table_schema</i>	Specifies the schema that qualifies the difference table. The default is ASN.
<b>DIFF_TABLESPACE=tablespace</b>	<p>Specifies the table space where the difference table will be placed. If this parameter is not specified, the table will be created in the default table space in the database or subsystem where the <i>asntdiff</i> command was run.</p> <p><b>z/OS</b> This is a two-part name, <i>dbname.tablespace</i>, where <i>dbname</i> is the logical database name and <i>tablespace</i> is the table space name.</p>
<b>DIFF_DROP=y/n</b>	<p>Specifies whether an existing difference table will be dropped and recreated before it is used to record differences. If the table does not exist, the <i>asntdiff</i> command creates it.</p> <p><b>n (default)</b> The difference table will be used as is and the existing rows will be deleted.</p> <p><b>y</b> The difference table will be dropped and recreated.</p>
<b>MAXDIFF=limit</b>	Specifies the maximum number of differences that you want the <i>asntdiff</i> command to process before it stops. The default value is 10000.

Tabella 41. *asntdiff* invocation parameter definitions for Linux, UNIX, Windows, and z/OS operating systems (Continua)

Parameter	Definition
<b>WHERE=WHERE_clause</b>	<p>Specifies an SQL WHERE clause that uniquely identifies one row of the control table that stores information about the source and target tables that will be compared. The WHERE clause must be in double quotation marks. The value of this parameter differs depending on whether you are using Q replication or SQL replication:</p> <p><b>Q replication</b> The WHERE clause specifies a row in the IBMQREP_SUBS table and uses the SUBNAME column to identify the Q subscription that contains the source and target tables.</p> <p><b>SQL replication</b> The WHERE clause specifies a row in the IBMSNAP_SUBS_MEMBR table and uses the SET_NAME, APPLY_QUAL, TARGET_SCHEMA, and TARGET_TABLE columns to identify the subscription set member that contains the source and target tables.</p>
<b>DIFF_PATH=log_path</b>	<p>Specifies the location where you want the <i>asntdiff</i> utility to write its log. The default value is the directory where you ran the command. The value must be an absolute path name. Use double quotation marks (") to preserve case.</p>
<b>PWDFILE=filename</b>	<p>Specifies the name of the password file that is used to connect to databases. If you do not specify a password file, the default value is <i>asnpwd.aut</i> (the name of the password file that is created by the <i>asnpwd</i> command). The <i>asntdiff</i> command searches for the password file in the directory that is specified by the <b>DIFF_PATH</b> parameter. If no value for the <b>DIFF_PATH</b> parameter is specified, the command searches for the password file in the directory where the command was run.</p>
<b>DIFF=table_name</b>	<p>Specifies the name of the table that will be created in the source database to store differences between the source and target tables. The table will have one row for each difference that is detected. If you do not include this parameter or the <b>DIFF_SCHEMA</b> parameter, the difference table will be named <b>ASN.ASNTDIFF</b>.</p>

Tabella 41. *asntdiff* invocation parameter definitions for Linux, UNIX, Windows, and z/OS operating systems (Continua)

Parameter	Definition
RANGECOL clause	<p>Specifies a range of rows from the source table that you want to compare. You provide the name of a DATE, TIME, or TIMESTAMP column in the source table, and then use one three different clauses for specifying the range. The column name must be enclosed in single quotation marks. The clause must be enclosed in double quotation marks.</p> <p>The timestamp uses the following format: YYYY-MM-DD-HH.MM.SS.mmmmm. For example, 2008-03-10-10.35.30.55555 is the GMT timestamp for March 10, 2008, 10:35 AM, 30 seconds, and 55555 microseconds.</p> <p>Use one of the following clauses:</p> <p><i>src_colname</i> <b>FROM:</b> <i>date-time_lower-bound</i> <b>TO:</b> <i>date-time_upper-bound</i></p> <p>Specifies a lower and upper bound for the range of rows to compare.</p> <p>The following example uses a TIMESTAMP column:</p> <pre>" 'SALETIME' FROM: 2008-02-08-03.00.00.00000 TO: 2008-02-15-03.00.00.00000"</pre> <p><b>Remember:</b> Both the <b>FROM:</b> and <b>TO:</b> keywords are required and both keywords must be followed by a colon (:).</p> <p><i>src_colname</i> <b>FROM:</b> <i>date-time</i></p> <p>Specifies that you want to compare all rows with timestamps that are greater than or equal to <i>date-time</i>.</p> <p>For example:</p> <pre>" 'SALE_TIME' FROM: 2008-03-10-10.35.30.55555"</pre> <p><i>src_colname</i> <b>TO:</b> <i>date-time</i></p> <p>Specifies that you want to compare all rows with timestamps that are less than or equal to the <i>date-time</i>.</p> <p>For example:</p> <pre>" 'SALETIME' TO: 2008-03-20-12.00.00.00000"</pre> <p><b>Recommendation:</b> For better performance, ensure that you have an index on the source column that is specified in the range clause. When you compare tables that are involved in peer-to-peer replication, you can use the IBM-generated IBMQREPVERTIME column for the source column in the range clause.</p> <p><b>Restriction:</b> The RANGECOL parameter is not valid for the <i>asntdiff</i> -f (input file) option. You can use a SQL WHERE clause in the input file to achieve similar results.</p>

Tabella 41. *asntdiff* invocation parameter definitions for Linux, UNIX, Windows, and z/OS operating systems (Continua)

Parameter	Definition
<code>SQLID=authorization_ID</code>	<b>z/OS</b> Specifies an authorization ID that can be used on z/OS to create the difference table. Use this parameter if the ID that is used to run the <code>asntdiff</code> command does not have authorization to create tables. The value of the <b>SQLID</b> parameter is used as the schema for the difference table if you do not explicitly specify a schema by using the <b>DIFF_SCHEMA</b> parameter.

## Examples for `asntdiff`

The following examples show how to use the `asntdiff` command.

**z/OS** See the `ASNTDIFF` sample program for sample JCL to run the `asntdiff` command.

### Example 1

In Q replication, to find the differences between a source and target table that are specified in a Q subscription named `my_qsub`, on a Q Capture server named `source_db`, with a Q Capture schema of `asn`:

```
asntdiff db=source_db schema=asn where="subname = 'my_qsub'"
```

### Example 2

In SQL replication, to find the differences between a source and target table that are specified in a subscription set called `my_set`, with a target table named `trg_table`, on an Apply control server named `apply_db`, with an Apply schema of `asn`, and to name the difference table `diff_table`:

```
asntdiff DB=apply_db schema=asn where="set_name = 'my_set'
and target_table = 'trg_table'" diff=diff_table
```

### Example 3

In Q replication, to find the differences between a range of rows in the source and target tables that are specified in a peer-to-peer Q subscription named `my_qsub`, on a Q Capture server named `source_db`, with a Q Capture schema of `asn`:

```
asntdiff db=source_db schema=asn where="subname = 'my_qsub'"
RANGECOL="'IBMQREPVERTIME' FROM: '2008-03-10-0.00.00.00000'
TO: '2007-04-12-00.00.00.00000'"
```

### Example 4

In SQL replication, to find the differences between a range of rows in the source and target table that are specified in a subscription set called `my_set`, with a target table named `trg_table`, on an Apply control server named `apply_db`, with an Apply schema of `asn`, and to name the difference table `diff_table`:

```
asntdiff DB=apply_db schema=asn where="set_name = 'my_set'
and target_table = 'trg_table'" diff=diff_table
RANGECOL="'CREDIT_TIME' FROM: '2008-03-10-12.00.00.00000'
TO: '2008-03-11-12.00.00.00000'"
```

---

## asntdiff -f (input file) command option

z/OS

Linux UNIX Windows

With the `asntdiff -f` command option, you use an input file to specify information about any two tables that you want to compare, whether or not they are being replicated.

The input file contains SQL `SELECT` statements for the source and target tables that specify the rows that you want to compare. The standard `asntdiff` command compares tables that are involved in replication by using subscription information from the replication control tables.

The `asntdiff -f` option can compare any tables on z/OS, Linux, UNIX, or Windows. You can run `asntdiff -f` from a Linux, UNIX, or Windows command prompt, from z/OS as a batch job that uses JCL, or from z/OS under the UNIX System Services (USS) environment.

In addition to the `SELECT` statements, the input file contains the source and target database information, the difference table information, and optional parameters that specify methods for processing the differences. You can use a password file that is created by the `asnpwd` command to specify a user ID and password for connecting to the source and target databases.

**Nota:** The `asntrep` command for repairing table differences does not support the input file option.

The format of the input file contents is as follows:

```
* Optional comment line
# Optional comment line
SOURCE_SERVER=server_name
SOURCE_SELECT="SQL_SELECT_STATEMENT"
TARGET_SERVER=server_name
TARGET_SELECT="SQL_SELECT_STATEMENT"
PARAMETER=value
...
```

Follow these guidelines:

- Each parameter must follow the *parameter=value* format.
- Multiple parameter-value pairs can be specified on a single line, separated by a blank. The parameter-value pairs also can be specified on a new line.
- To preserve blanks, surround parameter values with double quotation marks ("). Double quotation marks are also required for the source and target `SELECT` statements.
- If you want to preserve mixed case or blanks in the names of single DB2 objects (column or table names, `DIFF_SCHEMA`, `DIFF_TABLESPACE`) mask them with `\ " \`, for example `\ "MY NAME\ "` or `\ "ColumnName\ "` or `\ "name\ "`.
- Comments must be prefixed with an asterisk (\*) or pound sign (#). This line is ignored. Comments must be on their own line and cannot be added to a line that contains parameters.
- Surround the `DIFF_PATH` and `PWDFILE` parameters with double quotation marks ("). A final path delimiter for `DIFF_PATH` is not required.

## Syntax

▶▶ asntdiff -f *input\_filename* ▶▶

## Parameters

Tabella 42 defines the mandatory parameters to include in the input file for the asntdiff -f command.

For descriptions of optional parameters that you can include in the input file (and which are shared by the standard asntdiff command) see “asntdiff: Comparing data in source and target tables” a pagina 302.

*Tabella 42. asntdiff -f invocation parameter definitions for Linux, UNIX, Windows, and z/OS*

Parameter	Definition
<i>input_filename</i>	Specifies the name of the file that contains the source and target database information and SELECT statements. Specify a directory path if the file is located somewhere other than the directory from which you run the asntdiff -f command.
<b>SOURCE_SERVER=</b> <i>source_server_name</i>	Specifies the alias of the database where the source table exists.
<b>TARGET_SERVER=</b> <i>target_server_name</i>	Specifies the alias of the database where the target table exists.
<b>SOURCE_SELECT=</b> <i>source_select_statement</i>	Any valid SQL SELECT statement.
<b>TARGET_SELECT=</b> <i>target_select_statement</i>	<p>The result sets from the SQL statement at each table must contain columns with matching data types and lengths. The asntdiff command describes the queries and compares the data from the two result sets. The command does not explicitly check the system catalog for type and length information. The SELECT can be an open select as in (*), or a SELECT statement that contains column names, SQL expressions, and WHERE clauses that are permitted.</p> <p>An ORDER BY clause is mandatory. The clause must contain the numeric values of the positions of the columns in the SQL statement.</p> <p>Ensure that the column or columns in the ORDER BY clause reference a unique key or unique composite key. Otherwise the results are incorrect. An index on the columns in the ORDER BY clause might improve performance by eliminating the need for a sort.</p> <p>The entire statement must be enclosed in double quotes to mark the beginning and the end.</p>

The following examples show the mandatory parameters, SQL statements, and optional parameters that you put in the input file.

z/OS

## Example 1

This example shows the use of an open SELECT statement on DB2 for z/OS. Note the use of the \ " to preserve mixed case in the table owner, and the use of optional parameters in the input file. Also note the use of the DB2\_SUBSYSTEM parameter.

```
SOURCE_SERVER=STPLEX4A_DSN7
SOURCE_SELECT="select * from CXAIMS.ALDEC order by 1"
TARGET_SERVER=STPLEX4A_DSN7
TARGET_SELECT="select * from \"Cxaims\".TARG_ALDEC order by 1"
DIFF_DROP=Y
DB2_SUBSYSTEM=DSN7
MAXDIFF=10000
DEBUG=YES
```

z/OS

## Example 2

This example demonstrates the use of SUBSTR and CAST functions in the SELECT statements.

```
SOURCE_SERVER=D7DP
SOURCE_SELECT="select HIST_CHAR12,HIST_DATE,HIST_CHAR6,HIST_INT1,HIST_INT2,
HIST_INT3,SUBSTR(CHAR1,1,5) AS CHAR1,SUBSTR(CHAR2,1,10) AS CHAR2,HIST_INT3,
HIST_DEC1,HIST_DEC2,HIST_DEC3,CAST(INT1 AS SMALLINT) AS INT1
FROM BISVT.THIST17 ORDER BY 4"
TARGET_SERVER=STPLEX4A_DSN7
TARGET_SELECT="select HIST_CHAR12,HIST_DATE,HIST_CHAR6,HIST_INT1,HIST_INT2,
HIST_INT3,CHAR1,CHAR2,HIST_INT3,HIST_DEC1,HIST_DEC2,HIST_DEC3,SML1
FROM BISVT.THIST17 ORDER BY 4"
DB2_SUBSYSTEM=DSN7
DIFF_DROP=Y
DEBUG=YES
MAXDIFF=10000
```

Windows

## Example 3

This example compares the EMPLOYEE tables on SOURCEDB and TARGETDB and includes several optional parameters.

```
SOURCE_SERVER=SOURCEDB
SOURCE_SELECT="select FIRSTNME, LASTNAME, substr(WORKDEPT,1,1)
as WORKDEPT, EMPNO from EMPLOYEE order by 4"
TARGET_SERVER=TARGETDB
TARGET_SELECT="select FIRSTNME, LASTNAME, substr(WORKDEPT,1,1)
as WORKDEPT, EMPNO from EMPLOYEE order by 4"
DIFF_DROP=Y
DIFF =\"diffTable\"
DEBUG=YES
MAXDIFF=10000
PWDFILE="asnpwd.aut"
DIFF_PATH="C:\utils\"
```

Linux UNIX

## Example 4

This example compares the EMPLOYEE tables in a Linux or UNIX environment and uses a casting function.

```
SOURCE_SERVER=SOURCEDB
SOURCE_SELECT="select EMPNO, FIRSTNME, LASTNAME, cast(SALARY as INT)
as SALARY from EMPLOYEE order by 1"
TARGET_SERVER=TARGETDB
```

```
TARGET_SELECT="select EMPNO, FIRSTNME, LASTNAME, cast(SALARY as INT)
as SALARY from EMPLOYEE order by 1"
DIFF_DROP=Y
DIFF_="\diffTable\"
DEBUG=YES
MAXDIFF=10000
PWDFILE="asnpwd.aut"
DIFF_PATH="home/laxmi/utlils"
```

---

## asntrc: Operating the replication trace facility

z/OS

Linux UNIX Windows

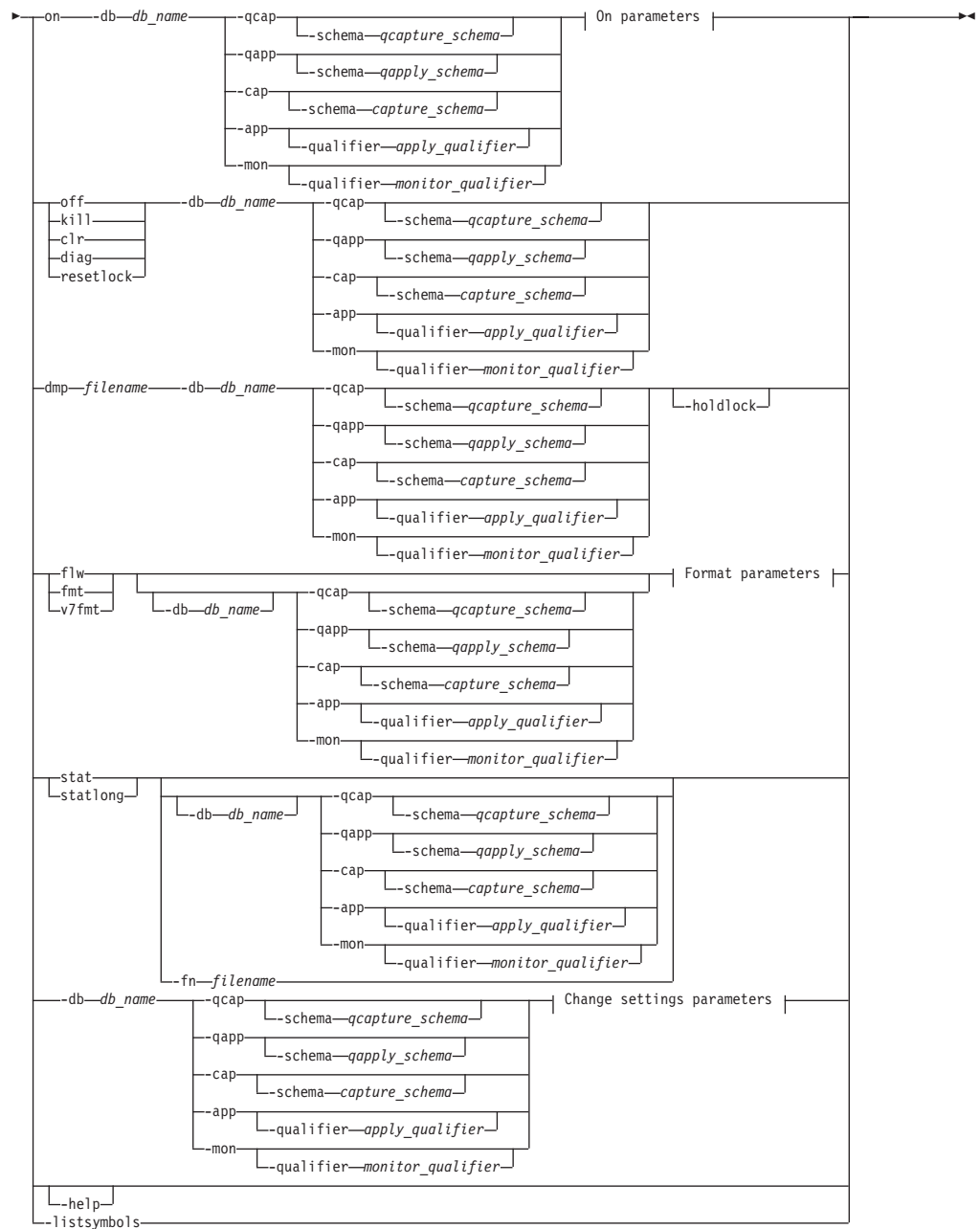
Use the `asntrc` command to run the trace facility on Linux, UNIX, Windows, and UNIX System Services (USS) on z/OS. The trace facility logs program flow information from Q Capture, Q Apply, Capture, Apply, and Replication Alert Monitor programs. You can provide this trace information to IBM Software Support for troubleshooting assistance. Run this command at an operating system prompt or in a shell script.

You run this command at an operating system prompt or in a shell script.

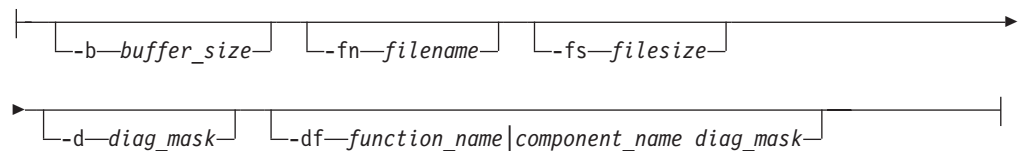
### Syntax

▶▶ `asntrc` →

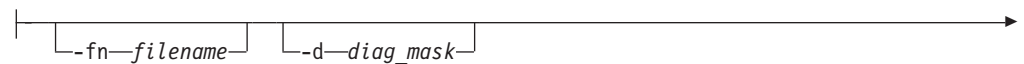




**On parameters:**



**Format parameters:**



```

┌-df--function_name|component_name diag_mask┐ ┌-holdlock┐

```

**Change settings parameters:**

```

┌-d--diag_mask┐ ┌-df--function_name|component_name diag_mask┐

```

**Parameters**

Tabella 43 defines the invocation parameters for the asntrc command.

*Tabella 43. asntrc invocation parameter definitions for Linux, UNIX, Windows, and z/OS operating systems*

Parameter	Definition
<b>on</b>	Specify to turn on the trace facility for a specific Q Capture, Q Apply, Capture, Apply, or Replication Alert Monitor program. The trace facility creates a shared memory segment used during the tracing process.
<b>-db</b> <i>db_name</i>	Specifies the name of the database to be traced: <ul style="list-style-type: none"> <li>• Specifies the name of the Q Capture server for the Q Capture program to be traced.</li> <li>• Specifies the name of the Q Apply server for the Q Apply program to be traced.</li> <li>• Specifies the name of the Capture control server for the Capture program to be traced.</li> <li>• Specifies the name of the Apply control server for the Apply program to be traced.</li> <li>• Specifies the name of the Monitor control server for the Replication Alert Monitor program to be traced.</li> </ul>
<b>-qcap</b>	Specifies that a Q Capture program is to be traced. The Q Capture program is identified by the <b>-schema</b> parameter.
<b>-schema</b> <i>qcapture_schema</i>	Specifies the name of the Q Capture program to be traced. The Q Capture program is identified by the Q Capture schema that you enter. Use this parameter with the <b>-qcap</b> parameter.
<b>-qapp</b>	Specifies that a Q Apply program is to be traced. The Q Apply program is identified by the <b>-schema</b> parameter.
<b>-schema</b> <i>qapply_schema</i>	Specifies the name of the Q Apply program to be traced. The Q Apply program is identified by the Q Apply schema that you enter. Use this parameter with the <b>-qapp</b> parameter.
<b>-cap</b>	Specifies that a Capture program is to be traced. The Capture program is identified by the <b>-schema</b> parameter.
<b>-schema</b> <i>capture_schema</i>	Specifies the name of the Capture program to be traced. The Capture program is identified by the Capture schema that you enter. Use this parameter with the <b>-cap</b> parameter.

Tabella 43. *asnlrc* invocation parameter definitions for Linux, UNIX, Windows, and z/OS operating systems (Continua)

Parameter	Definition
<b>-app</b>	Specifies that an Apply program is to be traced. The Apply program is identified by the <b>-qualifier</b> parameter.
<b>-qualifier</b> <i>apply_qualifier</i>	Specifies the name of Apply program to be traced. This Apply program is identified by the Apply qualifier that you enter. Use this parameter with the <b>-app</b> parameter.
<b>-mon</b>	Specifies that a Replication Alert Monitor program is to be traced. The Replication Alert Monitor program is identified by the <b>-qualifier</b> parameter.
<b>-qualifier</b> <i>monitor_qualifier</i>	Specifies the name of Replication Alert Monitor program to be traced. This Replication Alert Monitor program is identified by the monitor qualifier that you enter. Use this parameter with the <b>-mon</b> parameter.
<b>off</b>	Specify to turn off the trace facility for a specific Q Capture, Q Apply, Capture, Apply, or Replication Alert Monitor program and free the shared memory segment in use.
<b>kill</b>	Specify to force an abnormal termination of the trace facility.  Use this parameter only if you encounter a problem and are unable to turn the trace facility off with the <b>off</b> parameter.
<b>clr</b>	Specify to clear a trace buffer. This parameter erases the contents of the trace buffer but leaves the buffer active.
<b>diag</b>	Specify to view the filter settings while the trace facility is running.
<b>resetlock</b>	Specify to release the buffer latch of a trace facility. This parameter enables the buffer latch to recover from an error condition in which the trace program terminated while holding the buffer latch.
<b>dmp</b> <i>filename</i>	Specify to write the current contents of the trace buffer to a file.
<b>-holdlock</b>	Specifies that the trace facility can complete a file dump or output command while holding a lock, even if the trace facility finds insufficient memory to copy the buffer.
<b>flw</b>	Specify to display summary information produced by the trace facility and stored in shared memory or in a file. This information includes the program flow and is displayed with indentations that show the function and call stack structures for each process and thread.
<b>fmt</b>	Specify to display detailed information produced by the trace facility and stored in shared memory or in a file. This parameter displays the entire contents of the traced data structures in chronological order.
<b>v7fmt</b>	Specify to display information produced by the trace facility and stored in shared memory or in a file. This trace information appears in Version 7 format.

Tabella 43. *asnlrc* invocation parameter definitions for Linux, UNIX, Windows, and z/OS operating systems (Continua)

Parameter	Definition
<b>stat</b>	Specify to display the status of a trace facility. This status information includes the trace version, application version, number of entries, buffer size, amount of buffer used, status code, and program timestamp.
<b>statlong</b>	Specify to display the status of a trace facility with additional z/OS version level information. This additional information includes the service levels of each module in the application and appears as long strings of text.
<b>-fn filename</b>	Specifies the file name containing the mirrored trace information, which includes all the output from the trace facility.
<b>-help</b>	Displays the valid command parameters with descriptions.
<b>-listsymbols</b>	Displays the valid function and component identifiers to use with the <b>-df</b> parameter.
<b>-b buffer_size</b>	Specifies the size of the trace buffer (in bytes). You can enter a K or an M after the number to indicate kilobytes or megabytes, respectively; these letters are not case sensitive.
<b>-fs filesize</b>	Specifies the size limit (in bytes) of the mirrored trace information file.
<b>-d diag_mask</b>	<p>Specifies the types of trace records to be recorded by the trace facility. Trace records are categorized by a diagnostic mask number:</p> <ul style="list-style-type: none"> <li>1 Flow data, which includes the entry and exit points of functions.</li> <li>2 Basic data, which includes all major events encountered by the trace facility.</li> <li>3 Detailed data, which includes the major events with descriptions.</li> <li>4 Performance data.</li> </ul> <p><b>Important:</b> The higher diagnostic mask numbers are <i>not</i> inclusive of the lower diagnostic mask numbers.</p> <p>You can enter one or more of these numbers to construct a diagnostic mask that includes only the trace records that you need. For example, specify <b>-d 4</b> to record only performance data; specify <b>-d 1,4</b> to record only flow and performance data; specify <b>-d 1,2,3,4</b> (the default) to record all trace records. Separate the numbers with commas.</p> <p>Enter a diagnostic mask number of 0 (zero) to specify that no global trace records are to be recorded by the trace facility. Type <b>-d 0</b> to reset the diagnostic level before specifying new diagnostic mask numbers for a tracing facility.</p>

Tabella 43. *asntrc* invocation parameter definitions for Linux, UNIX, Windows, and z/OS operating systems (Continua)

Parameter	Definition
<code>-df function_name   component_name diag_mask</code>	Specifies that a particular function or component identifier is to be traced.  Type the diagnostic mask number (1,2,3,4) after the function or component identifier name. You can enter one or more of these numbers. Separate the numbers with commas.

## Examples for *asntrc*

The following examples illustrate how to use the *asntrc* command. These examples can be run on Linux, UNIX, Windows, or z/OS operating systems.

### Example 1

To trace a running Capture program:

1. Start the trace facility, specifying a trace file name with a maximum buffer and file size:

```
asntrc on -db mydb -cap -schema myschema -b 256k -fn myfile.trc -fs 500m
```

2. Start the Capture program, and let it run for an appropriate length of time.

3. While the trace facility is on, display the data directly from shared memory.

To display the summary process and thread information from the trace facility:

```
asntrc flw -db mydb -cap -schema myschema
```

To view the flow, basic, detailed, and performance data records only from the Capture log reader:

```
asntrc fmt -db mydb -cap -schema myschema -d 0  
-df "Capture Log Read" 1,2,3,4
```

4. Stop the trace facility:

```
asntrc off -db mydb -cap -schema myschema
```

The trace file contains all of the Capture program trace data that was generated from the start of the Capture program until the trace facility was turned off.

5. After you stop the trace facility, format the data from the generated binary file:

```
asntrc flw -fn myfile.trc
```

and

```
asntrc fmt -fn myfile.trc -d 0 -df "Capture Log Read" 1,2,3,4
```

### Example 2

To start a trace facility of a Replication Alert Monitor program:

```
asntrc on -db mydb -mon -qualifier monq
```

### Example 3

To trace only performance data of an Apply program:

```
asntrc on -db mydb -app -qualifier aql -b 256k -fn myfile.trc -d 4
```

#### Example 4

To trace all flow and performance data of a Capture program:

```
asntrc on dbserve1 -cap -schema myschema -b 256k
      -fn myfile.trc -d 1,4
```

#### Example 5

To trace all global performance data and the specific Capture log reader flow data of a Capture program:

```
asntrc on -db mydb -cap -schema myschema -b 256k -fn myfile.trc -d 4
      -df "Capture Log Read" 1
```

#### Example 6

To trace a running Capture program and then display and save a point-in-time image of the trace facility:

1. Start the trace command, specifying a buffer size large enough to hold the latest records:

```
asntrc on -db mydb -cap -schema myschema -b 4m
```

2. Start the Capture program, and let it run for an appropriate length of time.

3. View the detailed point-in-time trace information that is stored in shared memory:

```
asntrc fmt -db mydb -cap -schema myschema
```

4. Save the point-in-time trace information to a file:

```
asntrc dmp myfile.trc -db mydb -cap -schema myschema
```

5. Stop the trace facility:

```
asntrc off -db mydb -cap -schema myschema
```

### Examples for asntrc with shared segments

The standalone trace facility, `asntrc`, uses a shared segment to communicate with the respective Q Capture, Q Apply, Capture, Apply or Replication Alert Monitor programs to be traced. The shared segment will also be used to hold the trace entries if a file is not specified. Otherwise, matching options must be specified for both the `asntrc` command and for the respective programs to be traced to match the correct shared segment to control traces. The following examples show the options that need to be specified when the trace facility is used in conjunction with Q Capture, Q Apply, Capture, Apply or Alert Monitor programs.

With the Q Capture program, the database specified by the `-db` parameter with the `asntrc` command needs to match the database specified by the `capture_server` parameter with the `asnqcap` command:

```
asntrc -db ASN6 -schema EMI -qcap
asnqcap capture_server=ASN6 capture_schema=EMI
```

With the Q Apply program, the database specified by the `-db` parameter with the `asntrc` command needs to match the database specified by the `apply_server` parameter with the `asnqapp` command:

```
asntrc -db TSN3 -schema ELB -qapp
asnqapp apply_server=TSN3 apply_schema=ELB
```

With the Capture program, the database specified by the **-db** parameter with the `asntrc` command needs to match the database specified by the **capture\_server** parameter with the `asncap` command:

```
asntrc -db DSN6 -schema JAY -cap
asncap capture_server=DSN6 capture_schema=JAY
```

With the Apply program, the database specified by the **-db** parameter with the `asntrc` command needs to match the database specified by the **control\_server** parameter with the `asnapply` command:

```
asntrc -db SVL_LAB_DSN6 -qualifier MYQUAL -app
asnapply control_server=SVL_LAB_DSN6 apply_qual=MYQUAL
```

With the Replication Alert Monitor program, the database specified by the **-db** parameter with the `asntrc` command needs to match the database specified by the **monitor\_server** parameter with the `asnmon` command:

```
asntrc -db DSN6 -qualifier MONQUAL -mon
asnmon monitor_server=DSN6 monitor_qual=MONQUAL
```

---

## asntrep: Repairing differences between source and target tables

z/OS

Linux UNIX Windows

Use the `asntrep` command to synchronize a source and target table by repairing differences between the two tables. Run the `asntrep` command on Linux, UNIX, and Windows at an operating system prompt or in a shell script.

### Syntax

```
▶▶ asntrep—DB=server—DB2_SUBSYSTEM=subsystem—SCHEMA=schema—
  DIFF_SCHEMA=difference_table_schema—DIFF_TABLESPACE=tablespace—
  WHERE=WHERE_clause—DIFF_PATH=log_path—PWDFILE=filename—
  DIFF=table_name—▶▶
```

### Parameters

Tabella 44 a pagina 318 defines the invocation parameters for the `asntrep` command.

Tabella 44. *asntrep* invocation parameter definitions for Linux, UNIX, Windows, and z/OS operating systems

Parameter	Definition
<b>DB=server</b>	<p>Specifies the DB2 alias of the database that stores information about the source and target tables that you want to synchronize. The value differs depending on whether you are using Q replication or SQL replication:</p> <p><b>Q replication</b> The value is the name of the Q Capture server, which contains the IBMQREP_SUBS table.</p> <p><b>SQL replication</b> The value is the name of the Apply control server, which contains the IBMSNAP_SUBS_MEMBR table.</p> <p><b>z/OS</b> The value of this parameter is a location name.</p>
<b>DB2_SUBSYSTEM=subsystem</b>	<p><b>z/OS</b> Specifies the name of the subsystem where you run the <i>asntrep</i> utility.</p>
<b>SCHEMA=schema</b>	<p>Specifies the schema of the Q Capture control tables for Q replication, or the Apply control tables for SQL replication.</p>
<b>DIFF_SCHEMA=</b> <i>difference_table_schema</i>	<p>Specifies the schema that qualifies the difference table. The default is ASN.</p>
<b>DIFF_TABLESPACE=tablespace</b>	<p>Specifies the table space where a copy of the difference table is placed in the target database or subsystem. The copy is then used to repair the target table. If this parameter is not specified, the table will be created in the default table space in the database or subsystem in which the <i>asntrep</i> command was run.</p>
<b>WHERE=WHERE_clause</b>	<p>Specifies a SQL WHERE clause that uniquely identifies one row of the control table that stores information about the source and target tables that you are synchronizing. The WHERE clause must be in double quotation marks. The value of this parameter differs depending on whether you are using Q replication or SQL replication:</p> <p><b>Q replication</b> The WHERE clause specifies a row in the IBMQREP_SUBS table and uses the SUBNAME column to identify the Q subscription that contains the source and target tables.</p> <p><b>SQL replication</b> The WHERE clause specifies a row in the IBMSNAP_SUBS_MEMBR table and uses the SET_NAME, APPLY_QUAL, TARGET_SCHEMA, and TARGET_TABLE columns to identify the subscription set member that contains the source and target tables.</p>



Tabella 44. *asntrep* invocation parameter definitions for Linux, UNIX, Windows, and z/OS operating systems (Continua)

Parameter	Definition
<code>DIFF_PATH=log_path</code>	Specifies the location where you want the <i>asntrep</i> utility to write its log. The default value is the directory where you ran the command. The value must be an absolute path name. Use double quotation marks (") to preserve case.
<code>PWDFILE=filename</code>	Specifies the name of the password file that is used to connect to databases. If you do not specify a password file, the default value is <code>asnpwd.aut</code> (the name of the password file that is created by the <code>asnpwd</code> command). The <i>asntrep</i> utility searches for the password file in the directory that is specified by the <code>DIFF_PATH</code> parameter. If no value for the <code>DIFF_PATH</code> parameter is specified, the command searches for the password file in the directory where the command was run.
<code>DIFF=table_name</code>	Specifies the name of the table that was created in the source database by the <code>asntdiff</code> command to store differences between the source and target tables. The information that is stored in this table is used to synchronize the source and target tables.

## Examples for *asntrep*

The following examples illustrate how to use the *asntrep* command.

### Example 1

In Q replication, to synchronize a source and target table that are specified in a Q subscription named `my_qsub`, on a Q Capture server named `source_db`, with a Q Capture schema of `asn`, and whose differences are stored in a table called `q_diff_table`:

```
asntrep db=source_db schema=asn where="subname = 'my_qsub'" diff=q_diff_table
```

### Example 2

In SQL replication, to synchronize a source and target table that are specified in a subscription set called `my_set`, with a target table named `trg_table`, on an Apply control server named `apply_db`, with an Apply schema of `asn`, and whose differences are stored in a table called `sql_diff_table`:

```
asntrep DB=apply_db SCHEMA=asn WHERE="set_name = 'my_set'
and target_table = 'trg_table'" diff=sql_diff_table
```



---

## Capitolo 23. System commands for SQL replication (System i)

### System i

Some replication commands are specific to the System i operating system on System i servers. You can enter these commands at an operating system command prompt or through a command line program.

The following topics describe these commands.

---

### ADDDPRREG: Adding a DPR registration (System i)

#### System i

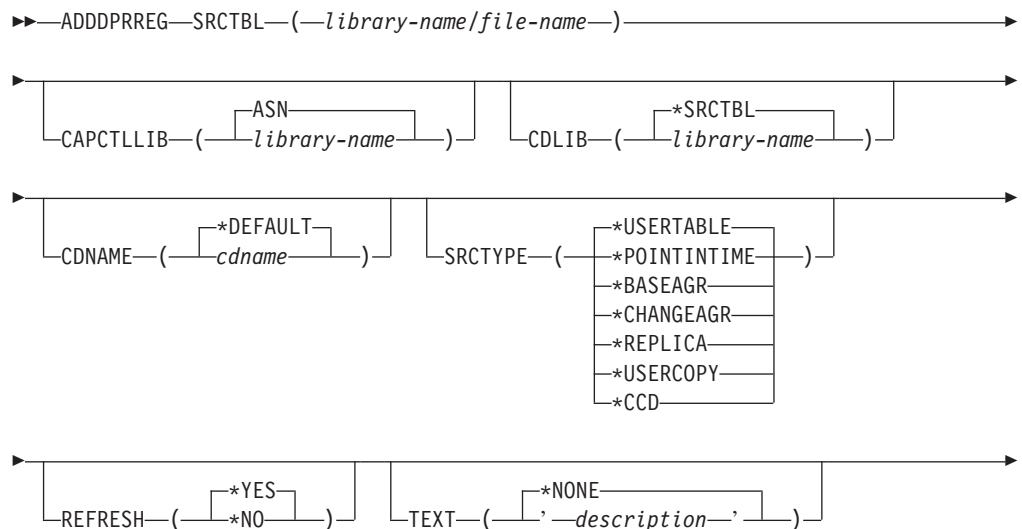
Use the Add DPR registration (ADDDPRREG) command to register a table as a source table for DB2 DataPropagator for iSeries.

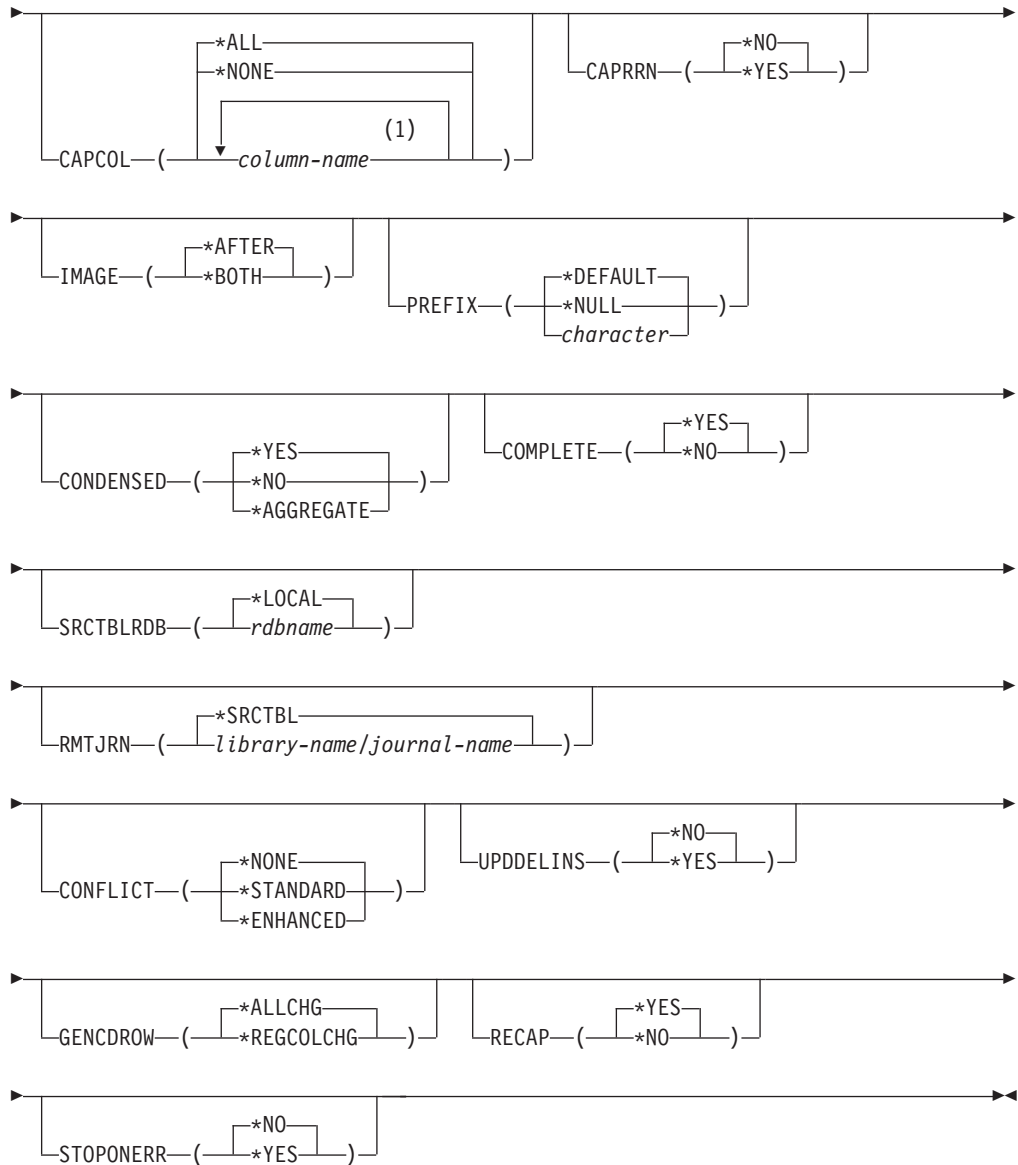
**Restriction:** You can register a table only if the ASN (Capture schema) library is in the same Auxiliary Pool (either base or independent ASP) where the ASN library is located.

After you type the command name on the command line, you can press the F4 key to display the command syntax.

To display a complete description of this command and all of its parameters, move the cursor to the command at the top of the screen and press the F1 key. To display a description of a specific parameter, place the cursor on that parameter and press the F1 key.

#### Syntax





**Note:**

1 You can specify up to 300 column names.

Tabella 45 lists the invocation parameters.

*Tabella 45. ADDPPREG command parameter definitions for System i*

Parameter	Definition and prompts
<b>SRCTBL</b>	Specifies the table that you want to register as a source table. The Capture program supports any physical file in a System i library or collection that is externally defined and in single format. This parameter is required.  <i>library-name/file-name</i> Represents the qualified name of the table that you want to register.

Tabella 45. ADDDPREG command parameter definitions for System i (Continua)

Parameter	Definition and prompts
<b>CAPCTLLIB</b>	<p>Specifies the Capture schema, which is the name of the library in which the Capture control tables reside.</p> <p><b>ASN (default)</b> The Capture control tables reside in the ASN library.</p> <p><i>library-name</i> The name of the library that contains the Capture control tables. You can create this library by using the CRTDPRTBL command with the CAPCTLLIB parameter.</p>
<b>CDLIB</b>	<p>Specifies the library in which the change-data (CD) table for this registered source is created.</p> <p><b>*SRCTBL (default)</b> Creates the CD table in the library in which the source table resides.</p> <p><i>library-name</i> Creates the CD table in this specified library name.</p>
<b>CDNAME</b>	<p>Specifies the name of the change-data (CD) table.</p> <p><b>*DEFAULT (default)</b> Creates the CD table with the default name, which is based on the current timestamp. For example, if the current timestamp is January 23, 2002 at 09:58:26, the default name is ASN020123095826CD.</p> <p><i>cdname</i> Creates the CD table with this specified name.</p>

Tabella 45. ADDDPREG command parameter definitions for System i (Continua)

Parameter	Definition and prompts
SRCTYPE	<p>Specifies the type of source table that you are registering. Choose a source type based on your replication configuration:</p> <ul style="list-style-type: none"> <li>• Use the default of USERTABLE for a basic data distribution or a data consolidation configuration.</li> <li>• Use REPLICA for an update-anywhere configuration.</li> <li>• Use POINTINTIME, BASEAGR, CHANGEAGR, USERCOPY, or CCD if you have a multi-tier configuration and want the target table to be a source for a subsequent tier in your replication configuration.</li> </ul> <p>If you are registering an existing target table as a source, the registration fails if the target table does not contain the IBMSNAP table columns indicated by the specified source type.</p> <p><b>*USERTABLE (default)</b>  A user database table, which is the most common type of registered table. The table cannot contain any columns that start with a DB2 DataPropagator for System i column identifier of either IBMSNAP or IBMQSQ.</p> <p><b>*POINTINTIME</b>  A point-in-time copy table, which includes content that matches all or part of the content of a source table and a DB2 DataPropagator for System i system column that identifies the time when a particular row was last inserted or updated at the source system. The table must contain the IBMSNAP_LOGMARKER timestamp column and can optionally contain an INTEGER column called IBMQSQ_RRN.</p> <p><b>*BASEAGR</b>  A base aggregate copy, which contains data aggregated at intervals from a user table or from a point-in-time table. The base aggregate table must contain the IBMSNAP_HLOGMARKER and IBMSNAP_LLOGMARKER timestamp columns.</p> <p><b>*CHANGEAGR</b>  A change aggregate copy table, which contains data aggregations that are based on changes recorded for a source table. The table must contain the IBMSNAP_HLOGMARKER and IBMSNAP_LLOGMARKER timestamp columns.</p> <p><b>*REPLICA</b>  A target table for a replica subscription. Register this type of table so that changes from the target table are replicated back to the original source table. This table cannot contain any DB2 DataPropagator for System i system columns or any columns that start with the DB2 DataPropagator for System i column identifier of either IBMSNAP or IBMQSQ. The table contains all of the columns from the original source table.</p> <p><b>*USERCOPY</b>  A target table with content that matches all or part of the content of a source table. The user copy table contains only user data columns.</p>

Tabella 45. ADDDPREG command parameter definitions for System i (Continua)

Parameter	Definition and prompts
SRCTYPE (Continued)	<p><b>*CCD</b> A consistent-change data (CCD) table, which contains transaction-consistent data from the source table. The table must contain columns that are defined as follows:</p> <ul style="list-style-type: none"> <li>• IBMSNAP_INTENTSEQ CHAR(10) FOR BIT DATA NOT NULL</li> <li>• IBMSNAP_OPERATION CHAR(1) NOT NULL</li> <li>• IBMSNAP_COMMITSEQ CHAR(10) FOR BIT DATA NOT NULL</li> <li>• IBMSNAP_LOGMARKER TIMESTAMP NOT NULL</li> </ul>
REFRESH	<p>Specifies whether the full-refresh capability is enabled. You can use this value to turn off the capability of the Apply program to perform a full refresh from the source database.</p> <p><b>*YES (default)</b> Full refreshes are allowed.</p> <p><b>*NO</b> Full refreshes are not allowed.</p> <p>If the target table is a base aggregate or change aggregate, you should set this parameter to *No.</p>
TEXT	<p>Specifies the textual description that is associated with this registration.</p> <p><b>*NONE (default)</b> No description is associated with the entry.</p> <p><i>description</i> The textual description of this registration. You can enter a maximum of 50 characters and must enclose the text in single quotation marks.</p>
CAPCOL	<p>Specifies the columns for which changes are captured for this registered table.</p> <p><b>*ALL (default)</b> Changes are captured for all columns.</p> <p><b>*NONE</b> Changes are not captured for this table. Use this value to specify that you want this table registered for full refresh only. The change-data (CD) table is not created with this registered table, and the Capture program will not capture changes for the table.</p> <p><i>column-name</i> The column names for which changes are captured. You can type up to 300 column names. Separate the column names with spaces.</p>
CAPRRN	<p>Specifies whether the relative record number (RRN) of each changed record is captured.</p> <p><b>*NO (default)</b> The relative record number is not captured.</p> <p><b>*YES</b> The relative record number is captured. An additional column called IBMQSQ_RRN is created in the change-data (CD) table.</p> <p>Set this parameter to *YES only if there are no unique keys in the source table.</p>

Tabella 45. ADDDPREG command parameter definitions for System i (Continua)

Parameter	Definition and prompts
<b>IMAGE</b>	<p>Specifies whether the change-data (CD) table contains both before and after images of the changes to the source table. This applies globally to all columns specified on the Capture columns (<b>CAPCOL</b>) parameter.</p> <p>This <b>IMAGE</b> parameter is not valid when the <b>CAPCOL</b> parameter is set to *NONE.</p> <p>The source table must be journaled with *BOTH images even if you specify *AFTER on this parameter.</p> <p><b>*AFTER (default)</b> The Capture program records only after images of the source table in the CD table.</p> <p><b>*BOTH</b> The Capture program records both before images and after images of the source table in the CD table.</p>
<b>PREFIX</b>	<p>Specifies the prefix character identifying before-image column names in the change-data (CD) table. You must ensure that none of the registered column names of the source table begins with this prefix character.</p> <p><b>*DEFAULT (default)</b> The default prefix (@) is used.</p> <p><b>*NULL</b> No before images are captured. This value is not valid if the <b>IMAGE</b> parameter is set to *BOTH.</p> <p><i>character</i> Any single alphabetic character that is valid in an object name.</p>
<b>CONDENSED</b>	<p>Specifies whether the source table is condensed. A condensed table contains current data with no more than one row for each primary key value in the table.</p> <p><b>*YES (default)</b> The source table is condensed.</p> <p><b>*NO</b> The source table is not condensed.</p> <p><b>*AGGREGATE</b> The source table type is either *BASEAGR (base aggregate) or *CHANGEAGR (change aggregate). If this value is used, you must set the <b>COMPLETE</b> parameter to *No</p>
<b>COMPLETE</b>	<p>Specifies whether the source table is complete, which means that the table contains a row for every primary key value of interest.</p> <p><b>*YES (default)</b> The source table is complete.</p> <p><b>*NO</b> The source table is not complete.</p>



Tabella 45. ADDDPREG command parameter definitions for System i (Continua)

Parameter	Definition and prompts
<b>SRCTBLRDB</b>	<p>Specifies whether you want to use remote journaling, in which the source table and the remote journal reside on different systems. Use this parameter to specify the location of the source table.</p> <p><b>*LOCAL (default)</b> The source table resides locally (on the machine where you are running the ADDDPREG command).</p> <p><i>rdbname</i> The name of the relational database where the source table exists. You can use the Work with RDB Directory Entries (WRKRDBDIRE) command to find this relational database name.</p>
<b>RMTJRN</b>	<p>Specifies the name of the remote journal when the name of this journal and the name of the journal on the source system are different. You must issue this command from the system where the remote journal resides.</p> <p><b>*SRCTBL (default)</b> The remote journal name is the same as the journal name of the source table.</p> <p><i>library-name/journal-name</i> The qualified library and journal name that reside on this system and are used for journaling the remote source table.</p> <p>You can specify a remote journal name only if you specified a remote source table location by using the <b>SRCTBLRDB</b> parameter.</p>
<b>CONFLICT</b>	<p>Specifies the conflict level that is used by the Apply program when detecting conflicts in a replica subscription.</p> <p><b>*NONE (default)</b> No conflict detection.</p> <p><b>*STANDARD</b> Moderate conflict detection. The Apply program searches for conflicts in rows that are already captured in the replica change-data (CD) tables.</p> <p><b>*ENHANCED</b> Enhanced conflict detection. This option provides the best data integrity among all replicas and source tables.</p>
<b>UPDDELINS</b>	<p>Determines how the Capture program stores updated source data in the change-data (CD) table.</p> <p><b>*NO (default)</b> The Capture program stores each source change in a single row in the CD table.</p> <p><b>*YES</b> The Capture program stores each source change by using two rows in the CD table, one for the delete and one for the insert. The Apply program processes the delete row first and the insert row second.</p>

Tabella 45. ADDDPREG command parameter definitions for System i (Continua)

Parameter	Definition and prompts
GENCDROW	<p>Specifies whether the Capture program captures changes from all rows in the source table.</p> <p><b>*ALLCHG (default)</b> The Capture program captures changes from all rows in the source table (including changes in unregistered columns) and adds these changes to the change-data (CD) table.</p> <p><b>*REGCOLCHG</b> The Capture program captures changes only if the changes occur in registered columns. The Capture program then adds these rows to the CD table.</p> <p>You cannot specify *REGCOLCHG if the CAPCOL parameter is set to *ALL or *NONE.</p>
RECAP	<p>Specifies whether the changes made by the Apply program are recaptured by the Capture program.</p> <p><b>*YES (default)</b> Changes made to the source table by the Apply program are captured and entered into the change-data (CD) table.</p> <p><b>*NO</b> Changes that were made to the source table by the Apply program are not captured and, therefore, do not appear in the CD table. You should use this option when registering REPLICA type tables.</p>
STOPONERR	<p>Specifies whether the Capture program stops when it encounters an error.<sup>1</sup></p> <p><b>*NO (default)</b> The Capture program does not stop when it encounter an error. The Capture program issues messages, deactivates the registration that caused the error, and then continues processing.</p> <p><b>*YES</b> The Capture program issues messages and then stops when it encounters an error.</p>

**Nota:**

1. If this parameter is set to Yes (Y), the Capture journal job stops while other journal jobs continue to run. If this parameter is set to No (N), the Capture program stops the registration file that contains the error.  
This parameter also sets the columns in the register table rows. The STATE column is set to 'S' and the STATE\_INFO column to is set 200Axxx where xxx is the reason code. To set the registration back to the Action ('A') state, perform the following steps:
  - Correct the ASN200A message. Refer to the appropriate System i documentation for the corrected action.
  - Use the Replication Center or the System i command STRSQL to set the columns in the IBMSNAP\_REGISTER table row. Set the STATE column to 'A', and the STATE\_INFO column to null.
  - If Capture is running, issue the INZDPRCAP command to reinitialize data replication for that journal.

## Examples for ADDDPREG

The following examples illustrate how to use the ADDDPREG command.

### Example 1:

To register a source table named EMPLOYEE from the HR library under the default Capture schema:

```
ADDDPRREG SRCTBL(HR/EMPLOYEE)
```

#### Example 2:

To register a source table named EMPLOYEE from the HR library under the BSN Capture schema and to create a CD table named CDEMPLOYEE under the HRCDLIB library:

```
ADDDPRREG SRCTBL(HR/EMPLOYEE) CAPCTLLIB(BSN) CDLIB(HRCDLIB) CDNAME(CDEMPLOYEE)
```

#### Example 3:

To register a source table with a source type of point-in-time that is named SALES from the DEPT library under the BSN Capture schema:

```
ADDDPRREG SRCTBL(DEPT/SALES) CAPCTLLIB(BSN) SRCTYPE(*POINTINTIME)
```

#### Example 4:

To register a source table named SALES from the DEPT library and to specify that the CD table contains both before and after images of source table changes:

```
ADDDPRREG SRCTBL(DEPT/SALES) IMAGE(*BOTH)
```

#### Example 5:

To register a source table named SALES from the DEPT library of the relational database named RMTRDB1 using remote journals:

```
ADDDPRREG SRCTBL(DEPT/SALES) SRCTBLRDB(RMTRDB1) RMTJRN(RMTJRNLIB/RMTJRN)
```

#### Example 6:

To register the EMPLOYEE source table from the HR library and to capture changes only for the EMPNO, NAME, DEPT, and NETPAY columns:

```
ADDDPRREG SRCTBL(HR/EMPLOYEE) CAPCOL(EMPNO NAME DEPT NETPAY)
```

---

## ADDDPRSUB: Adding a DPR subscription set (System i)

### System i

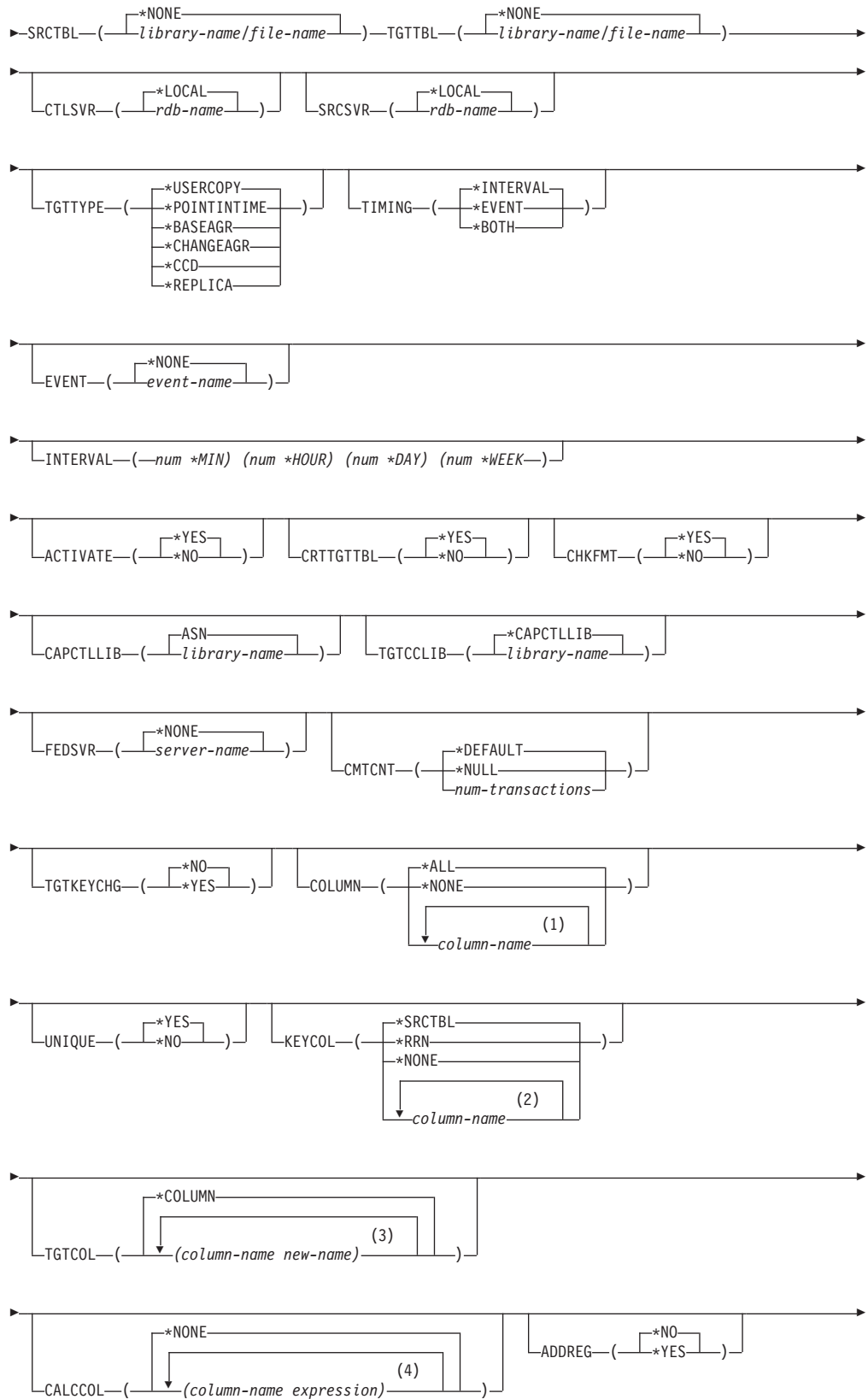
Use the Add DPR subscription set (ADDDPRSUB) command to create a subscription set with either one member or no members.

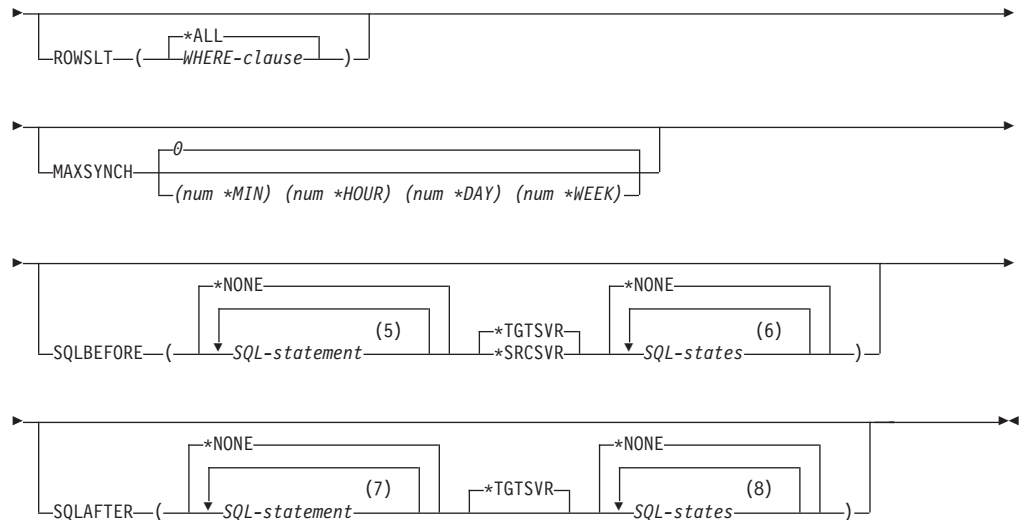
After you type the command name on the command line, you can press the F4 key to display the command syntax.

To display a complete description of this command and all of its parameters, move the cursor to the command at the top of the screen and press the F1 key. To display a description of a specific parameter, place the cursor on that parameter and press the F1 key.

### Syntax

```
►►—ADDDPRSUB—APYQUAL—(—apply-qualifier—)—SETNAME—(—set-name—)—►►
```





**Note:**

- 1 You can specify up to 300 column names.
- 2 You can specify up to 120 column names.
- 3 You can specify up to 300 column names.
- 4 You can specify up to 100 column names and expressions.
- 5 You can specify up to 3 SQL statements.
- 6 You can specify up to 10 SQLSTATES.
- 7 You can specify up to 3 SQL statements.
- 8 You can specify up to 10 SQLSTATES.

Tabella 46 lists the invocation parameters.

Tabella 46. ADDDPRSUB command parameter definitions for System i

Parameter	Definition and prompts
APYQUAL	Specifies the Apply qualifier that identifies which Apply program processes this subscription set. Subscription sets under an Apply qualifier run in a separate job. This parameter is required. <i>apply-qualifier</i> The name of the Apply qualifier.
SETNAME	Specifies the subscription-set name. This parameter is required. <i>set-name</i> The name of the subscription set. The subscription-set name that you enter must be unique for the specified Apply qualifier or the ADDDPRSUB command produces an error. Because the Apply program handles the set of target tables as a group, when one target table fails for any reason, the entire subscription set fails.

Tabella 46. ADDDPRSUB command parameter definitions for System i (Continua)

Parameter	Definition and prompts
SRCTBL	<p>Specifies the name of the source table that is used to copy information into your subscription set. You must register this table at the Capture control server before this table can become a member of a subscription set. This parameter is required.</p> <p><b>*NONE (default)</b> This subscription set does not have a source member. Use when creating a subscription set without members.</p> <p><i>library-name/file-name</i> The qualified name of the source table. Use when creating a subscription set with one member.</p>
TGTTBL	<p>Specifies the name of the target table. The target table is automatically created if you set the <b>CRTTGTTBL</b> parameter to <b>*YES</b> and the target table does not already exist. This parameter is required.</p> <p><b>*NONE (default)</b> This subscription set does not have a target member. Use when creating a subscription set without members.</p> <p><i>library-name/file-name</i> The qualified name of the target table. Use when creating a subscription set with one member.</p>
CTLSVR	<p>Specifies the relational database name of the system that contains the Apply control tables.</p> <p><b>*LOCAL (default)</b> The Apply control tables reside locally (on the machine from which you are running the ADDDPRSUB command).</p> <p><i>rdb-name</i> The name of the relational database where the Apply control tables reside. You can use the Work with RDB Directory Entries (WRKRDBDIRE) command to find this name.</p>
SRCSVR	<p>Specifies the relational database name of the system that contains the Capture control tables.</p> <p><b>*LOCAL (default)</b> The source table is registered on the local machine (the machine from which you are running the ADDDPRSUB command).</p> <p><i>rdb-name</i> The name of the relational database where the Capture control tables reside. You can use the Work with RDB Directory Entries (WRKRDBDIRE) command to find this name.</p>

Tabella 46. ADDDPRSUB command parameter definitions for System i (Continua)

Parameter	Definition and prompts
TGTTYPE	<p>Specifies the target table type. After you create a target table as one of these types, you can use this parameter value on the <b>SRCTBL</b> parameter of the Add DPR Registration (ADDDPRREG) command to register this target table as a source table for multi-tier replication.</p> <p><b>*USERCOPY (default)</b></p> <p>The target table is a user copy, which is a target table with content that matches all or part of the content of a source table. A user copy is handled like a point-in-time copy but does not contain any of the DB2 DataPropagator for System i system columns that are present in the point-in-time target table.</p> <p>This value is not valid when a value of *RRN is specified on the <b>KEYCOL</b> parameter.</p> <p>The table that you specified with the <b>SRCTBL</b> parameter must be one of the following types: user database, point-in-time copy, or consistent-change data (CCD).</p> <p><b>Important:</b> If the target table already exists, DB2 DataPropagator for System i does not automatically journal changes to it. You must start journaling outside of DB2 DataPropagator for System i.</p> <p><b>*POINTINTIME</b></p> <p>The target table is a point-in-time copy. A point-in-time copy is a target table with content that matches all or part of the content of the source table and includes the DB2 DataPropagator for System i system column (IBMSNAP_LOGMARKER), which identifies when a particular row was inserted or updated at the Capture control server.</p> <p><b>*BASEAGR</b></p> <p>The target table is a base aggregate copy, which is a target table that contains data that is aggregated (calculated) from a source table. The source table for a base aggregate target must be either a user table or a point-in-time table. This target table must contain the IBMSNAP_HLOGMARKER and IBMSNAP_LLOGMARKER system timestamp columns.</p> <p><b>*CHANGEAGR</b></p> <p>The table is a change aggregate copy, which is a target table that contains data that is aggregated (calculated) based on the contents of a change-data (CD) table. This target table is created with the IBMSNAP_HLOGMARKER and IBMSNAP_LLOGMARKER system timestamp columns.</p>

Tabella 46. ADDPRSUB command parameter definitions for System i (Continua)

Parameter	Definition and prompts
<b>TGTTYPE</b> (Continued)	<p><b>*CCD</b> The table is a consistent-change data (CCD) table, which is a target table created from a join of data in the change-data (CD) table and the unit-of-work (UOW) table. A CCD table provides transaction-consistent data for the Apply program and must include the following columns:</p> <ul style="list-style-type: none"> <li>• IBMSNAP_INTENTSEQ</li> <li>• IBMSNAP_OPERATION</li> <li>• IBMSNAP_COMMITSEQ</li> <li>• IBMSNAP_LOGMARKER</li> </ul> <p><b>*REPLICA</b> The target table is a replica table, which is used only for update-anywhere replication. The replica target table receives changes from the master source table, and changes to the replica target table are propagated back to the master source table. A replica table is automatically registered as a source table.</p>
<b>TIMING</b>	<p>Specifies the type of timing (scheduling) that the Apply program uses to process the subscription set.</p> <p><b>*INTERVAL (default)</b> The Apply program processes the subscription set at a specific time interval (for example, once a day).</p> <p><b>*EVENT</b> The Apply program processes the subscription set when a specific event occurs.</p> <p><b>*BOTH</b> The Apply program processes the subscription set either at a specific interval or when an event occurs, whichever occurs first.</p>
<b>EVENT</b>	<p>Specifies an event. The event that you enter must match an event name in the IBMSNAP_SUBS_EVENT) table.</p> <p><b>*NONE (default)</b> No event is used.</p> <p><i>event-name</i> A unique character string that represents an event described in the IBMSNAP_SUBS_EVENT table.</p>



Tabella 46. ADDDPRSUB command parameter definitions for System i (Continua)

Parameter	Definition and prompts
INTERVAL	<p>Specifies the time interval (weeks, days, hours, and minutes) from start time to start time between refreshes of the target copy. This is a two-part value. The first part is a number; the second part is the unit of time:</p> <p><b>*MIN</b> Minutes</p> <p><b>*HOUR</b> Hours</p> <p><b>*DAY</b> Days</p> <p><b>*WEEK</b> Weeks</p> <p>You can specify combinations of numbers with units of time. For example, ((2 *WEEK) (3 *DAY) (35 *MIN)) specifies a time interval of two weeks, three days, and 35 minutes. If you specify multiple occurrences of the same unit of time, the last occurrence is used.</p>
ACTIVATE	<p>Specifies whether the subscription set is active. The Apply program does not process this subscription set unless this parameter is set to *YES.</p> <p><b>*YES (default)</b> The subscription set is active.</p> <p><b>*NO</b> The subscription set is not active.</p>
CRITGTTBL	<p>Specifies whether the target table (or view) is created.</p> <p><b>*YES (default)</b> Creates the target table (or view) if it does not exist. Otherwise, the existing table or view becomes the target, and the format of this existing table or view is checked if the value of the <b>CHKFMT</b> parameter is set to *YES. An additional index, with the values that you specified by the <b>UNIQUE</b> and <b>KEYCOL</b> parameters, is created for a target table if no such index currently exists. The command fails if an existing target table contains rows that violate the conditions of the additional index.</p> <p><b>*NO</b> Does not create the target table or view. You must create the table or view with the correct attributes before starting the Apply program.</p> <p>If the table or view exists and you set <b>CHKFMT</b> to *YES, the ADDDPRSUB command ensures that the format of the existing table matches the subscription-set definition that you set. If <b>CHKFMT</b> is *NO, you must ensure that the format of the existing table matches the subscription-set definition.</p> <p><b>Important:</b> If the table or view already exists, DB2 DataPropagator for System i does not automatically journal changes to the existing object. You must start journaling outside of DB2 DataPropagator for System i.</p>

Tabella 46. ADDPRSUB command parameter definitions for System i (Continua)

Parameter	Definition and prompts
<b>CHKFMT</b>	<p>Specifies whether DB2 DataPropagator for System i checks the subscription set and the target table to ensure that the columns match. This parameter is ignored if the <b>CRTTGTTBL</b> parameter is *YES; this parameter is also ignored if the <b>CRTTGTTBL</b> parameter is set to *NO and the target table does not exist.</p> <p><b>*YES (default)</b> DB2 DataPropagator for System i verifies that the columns defined for this subscription set match the columns in the target table. This command fails if a mismatch is detected.</p> <p><b>*NO</b> DB2 DataPropagator for System i ignores the differences between the subscription set and the existing target table. You must ensure that the target table is compatible with the subscription set.</p>
<b>CAPCTLLIB</b>	<p>Specifies the Capture schema, which is the name of the library in which the Capture control tables reside. These Capture control tables process the source for this subscription set.</p> <p><b>ASN (default)</b> The Capture control tables reside in the ASN library.</p> <p><i>library-name</i> The name of a library that contains the Capture control tables. This is the library in which the source table was registered.</p>
<b>TGTCLIB</b>	<p>Specifies the target control library.</p> <p><b>*CAPCTLLIB (default)</b> The target control library is the same library in which the Capture control tables reside.</p> <p><i>library-name</i> The name of a library that contains the target control tables.</p> <p>If you are using a target table as a source for another subscription set (such as an external CCD table), this parameter value is the Capture schema when this table is used as a source.</p>
<b>FEDSVR</b>	<p>Specifies whether a federated database system is the source for this subscription set.</p> <p><b>*NONE (default)</b> The source server is not a federated database system.</p> <p><i>server-name</i> The name of the federated database system for this subscription set (for non-DB2 relational sources).</p>

Tabella 46. ADDDPRSUB command parameter definitions for System i (Continua)

Parameter	Definition and prompts
CMTCNT	<p>Specifies the commitment count, which is the number of transactions that the Apply program processes before a commit.</p> <p><b>*DEFAULT (default)</b> The command determines the value to use. If the TGTTYPE is set to *REPLICA, then the CMTCNT is zero (0). If the TGTTYPE is anything other than *REPLICA, the CMTCNT is null.</p> <p><b>*NULL</b> The subscription set is read-only. The Apply program will fetch answer sets for the subscription-set members one member at a time, until all data has been processed and then will issue a single commit for the entire subscription set.</p> <p><i>num-transactions</i> Specifies the number of transactions processed before the Apply program commits the changes. This parameter is valid only if the TGTTYPE parameter is set to *REPLICA.</p>
TGTKEYCHG	<p>Specifies how the Apply program handles updates when changes occur in source columns that are part of the target key columns for the target table. This parameter works in conjunction with the USEDELINS parameter on the ADDDPRREG command:</p> <ul style="list-style-type: none"> <li>• If USEDELINS is YES and TGTKEYCHG is YES, updates are not allowed.</li> <li>• If USEDELINS is YES and TGTKEYCHG is NO, updates become delete and insert pairs.</li> <li>• If USEDELINS is NO and TGTKEYCHG is YES, the Apply program handles this condition with special logic.</li> <li>• If USEDELINS is NO and TGTKEYCHG is NO, the Apply program processes the changes as normal updates.</li> </ul> <p><b>*NO (default)</b> Updates to the source table are staged by the Capture program and processed by the Apply program to the target table.</p> <p><b>*YES</b> The Apply program updates the target table based on the before images of the target key column, meaning that the Apply program changes the predicate to the old values instead of the new.</p>

Tabella 46. ADDPRSUB command parameter definitions for System i (Continua)

Parameter	Definition and prompts
COLUMN	<p>Specifies the columns to be included in the target table. The column names must be unqualified. Choose the column names from the list of column names that you specified with the <b>CAPCOL</b> parameter when you registered the source table.</p> <p>If you set the <b>IMAGE</b> parameter to <b>*BOTH</b> when registering this table, you can specify before-image column names. The before-image column names are the original column names with a prefix. This prefix is the character that you specified in the <b>PREFIX</b> parameter of the <b>ADDDPRREG</b> command.</p> <p><b>*ALL (default)</b> All of the columns that you registered in the source are included in the target table.</p> <p><b>*NONE</b> No columns from the source table are included in the target table. You can use <b>*NONE</b> when you want only computed columns in the target table. This value is required if the <b>CALCCOL</b> parameter contains summary functions but no <b>GROUP BY</b> is performed.</p> <p><i>column-name</i> The names of up to 300 source columns that you want to include in the target table. Separate the column names with spaces.</p>
UNIQUE	<p>Specifies whether the target table has unique keys as indicated by the <b>KEYCOL</b> parameter.</p> <p><b>*YES (default)</b> The target table supports one net change per key; only one row exists in the target table for that key regardless of how many changes are made to the key.</p> <p>This value specifies that the table contains current data rather than a history of changes to the data. A condensed table includes no more than one row for each primary key value in the table and can be used to supply current information for a refresh.</p> <p><b>*NO</b> The target table supports multiple changes per key. The changes are appended to the target table.</p> <p>This value specifies that the table contains a history of changes to the data rather than current data. A non-condensed table includes more than one row for each key value in the table and can be used to supply a history of changes to the data. A non-condensed table cannot supply current data for a refresh.</p>

Tabella 46. ADDDPRSUB command parameter definitions for System i (Continua)

Parameter	Definition and prompts
KEYCOL	<p>Specifies columns that describe the key of the target table. The column names must be unqualified. For *POINTINTIME, *REPLICA, and *USERCOPY target tables (as specified on the TGTTYPE parameter), you must identify one or more columns as the target key for the target table. This target key is used by the Apply program to identify each unique row that changes during change-capture replication.</p> <p><b>*SRCTBL (default)</b></p> <p>The key columns in the target table are the same as those in the source table. The ADDDPRREG command uses the key that is specified in the source table if the source table is keyed. The following key columns are used:</p> <ul style="list-style-type: none"> <li>• Key columns that you defined through DDS when creating the table with the Create Physical File (CRTPF) command</li> <li>• Primary and unique keys that you defined with the CREATE TABLE and ALTER TABLE SQL statements</li> <li>• Unique keys that you defined with the CREATE INDEX SQL statements</li> </ul> <p>If you use a column as a key more than once and with different ordering, the target table key is defined with ascending order.</p> <p><b>*RRN</b></p> <p>The key column in the target table is the IBMQSQ_RRN column. The target table is created with an IBMQSQ_RRN column, and this column is used as the key. When the Apply program runs, if the source table is a user table and the target table is a point-in-time or user copy, the IBMQSQ_RRN column in the target table is updated with the relative record number of the associated record in the source table. Otherwise, the IBMQSQ_RRN column in the target table is updated with the value of the IBMQSQ_RRN column in the source table.</p> <p><b>*NONE</b></p> <p>The target copy does not contain a target key. You cannot specify *NONE if the target table type is *POINTINTIME, *REPLICA, or *USERCOPY.</p> <p><i>column-name</i></p> <p>The names of the target columns that you want to use as the target key columns. You can specify up to 120 column names. Separate the column names with spaces.</p>

Tabella 46. ADDDPRSUB command parameter definitions for System i (Continua)

Parameter	Definition and prompts
TGTCOL	<p>Specifies the new names for all the columns that the Apply program updates in the target table. These names override the column names taken from the source table. The column names must be unqualified. If you specified a value of *NONE for the <b>COLUMN</b> parameter, do not use this parameter.</p> <p>Use this parameter to give more meaningful names to the target table columns. Specify each source column name and the name of the corresponding column on the target table.</p> <p><b>*COLUMN (default)</b> The target columns are the same as the columns you specified in the <b>COLUMN</b> parameter.</p> <p><i>column-name</i> The column names from the source table that you want to change at the target. You can list up to 300 column names.</p> <p><i>new-name</i> The new names of the target columns. You can list up to 300 new column names. If you do not use this parameter, the name of the column on the target table will be the same as the source column name.</p>
CALCCOL	<p>Specifies the list of user-defined or calculated columns in the target table. The column names must be unqualified. Enclose each column name and expression pair in parenthesis.</p> <p>You must specify a column name for each SQL expression. If you want to define any column as an SQL expression without a GROUP BY statement, you must set the <b>COLUMN</b> parameter to *NONE.</p> <p><b>*NONE (default)</b> No user-defined or calculated columns are included in the target table.</p> <p><i>column-name</i> The column names of the user-defined or calculated columns in the target table. You can list up to 100 column names.</p> <p><i>expression</i> The expressions for the user-defined or calculated columns in the target table. You can list up to 100 SQL column expressions.</p>
ADDREG	<p>Specifies whether the target table is automatically registered as a source table. Use this parameter to register CCD target type tables.</p> <p><b>*NO (default)</b> The target table is not registered as a source table. DB2 DataPropagator for System i ignores this parameter value if the target type is *REPLICA. Replica target tables are always automatically registered as source tables.</p> <p><b>*YES</b> The target table is registered as a source table. This command fails if you already registered the target table.</p> <p>Do not set this parameter to *YES if the target table type is *USERCOPY, *POINTINTIME, *BASEAGR, or *CHANGEAGR.</p> <p>If you set the <b>CRTTGTTBL</b> parameter to *NO, you must create the target table before attempting to register it as a source.</p>

Tabella 46. ADDPRSUB command parameter definitions for System i (Continua)

Parameter	Definition and prompts
ROWSLT	<p>Specifies the predicates to be placed in an SQL WHERE clause. The Apply program uses these predicates to determine which rows in the change-data (CD) table of the source to apply to the target table. Use this parameter if you want only a subset of the source changes to be replicated to the target table.</p> <p><b>*ALL (default)</b> The Apply program applies all changes in the CD table to the target table.</p> <p><i>WHERE-clause</i> The SQL WHERE clause that specifies which rows from the CD table the Apply program applies to the target table. Do not include the WHERE keyword; it is implied on this parameter. This WHERE clause must be valid on the data server you are using to run the clause.</p> <p><b>Note:</b> The WHERE clause on this parameter is unrelated to any WHERE clauses specified on the SQLBEFORE or SQLAFTER parameters.</p>
MAXSYNCH	<p>Specifies the maximum synch minutes. This parameter is the time-threshold limit used to regulate the amount of change data that the Capture and Apply programs process during a subscription cycle. You can specify the time-threshold limit by using a two-part value. The first part is a number; the second part is the unit of time:</p> <p><b>*MIN</b> Minutes</p> <p><b>*HOUR</b> Hours</p> <p><b>*DAY</b> Days</p> <p><b>*WEEK</b> Weeks</p> <p>You can specify combinations of numbers with units of time. For example, ((1 *WEEK) (2 *DAY) (35 *MIN)) specifies a time interval of one week, two days, and 35 minutes. If you specify multiple occurrences of the same unit of time, the last occurrence is used.</p> <p>The default is zero (0), which indicates that all of the change data is to be applied.</p>

Tabella 46. ADDDPRESUB command parameter definitions for System i (Continua)

Parameter	Definition and prompts
SQLBEFORE	<p>Specifies the SQL statements that run before the Apply program refreshes the target table. This parameter has three elements:</p> <p>Element 1: SQL code</p> <p><b>*NONE (default)</b> No SQL statement is specified.</p> <p><i>SQL-statement</i> The SQL statement that you want to run. Ensure that the syntax of the SQL statement is correct. DB2 DataPropagator for System i does not validate the syntax. In addition, you must use the proper SQL naming conventions. SQL file references must be in the form of LIBRARY.FILE instead of the system naming convention (LIBRARY/FILE). You can specify up to three SQL statements.</p> <p>Element 2: Server to run on</p> <p><b>*TGTSVR (default)</b> The SQL statement runs at the target server on which the target table is located.</p> <p><b>*SRCSVR</b> The SQL statement runs at the Capture control server on which the source table is located.</p> <p>Element 3: Allowed SQLSTATE values</p> <p><b>*NONE (default)</b> Only an SQLSTATE value of 00000 is considered successful.</p> <p><i>SQL-states</i> A list of one to ten allowable SQLSTATE values. Separate the SQLSTATE values with spaces. An SQLSTATE value is a five-digit hexadecimal number ranging from 00000 to FFFFF.</p> <p>The SQL statement is successful if it completes with an SQLSTATE value of 00000 or with one of the allowable SQLSTATE values that you listed.</p>



Tabella 46. ADDDPRSUB command parameter definitions for System i (Continua)

Parameter	Definition and prompts
SQLAFTER	<p>Specifies SQL statements that run after the Apply program refreshes the target table. This parameter has three elements:</p> <p>Element 1: SQL code</p> <p><b>*NONE (default)</b> No SQL statement is specified.</p> <p><i>SQL-statement</i> The SQL statement that you want to run. Ensure that the syntax of the SQL statement is correct. DB2 DataPropagator for System i does not validate the syntax. In addition, you must use the proper SQL naming conventions. SQL file references must be in the form of LIBRARY.FILE instead of the system naming convention (LIBRARY/FILE). You can specify up to three SQL statements.</p> <p>Element 2: Server to run on</p> <p><b>*TGTSVR (default)</b> The SQL statement runs at the target server on which the target table is located.</p> <p>Element 3: Allowed SQLSTATE values</p> <p><b>*NONE (default)</b> Only an SQLSTATE value of 00000 is considered successful.</p> <p><i>SQL-states</i> A list of one to ten allowable SQLSTATE values. Separate the SQLSTATE values with spaces. An SQLSTATE value is a five-digit hexadecimal number ranging from 00000 to FFFFF.</p> <p>The SQL statement is successful if it completes with an SQLSTATE value of 00000 or with one of the allowable SQLSTATE values that you listed.</p>

## Examples for ADDDPRSUB

The following examples illustrate how to use the ADDDPRSUB command.

### Example 1:

To create a subscription set named SETHR under the AQHR Apply qualifier:

```
ADDDPRSUB APYQUAL(AQHR) SETNAME(SETHR) SRCTBL(HR/EMPLOYEE)
TGTTBL(TGTLIB/TGTEMPL)
```

This subscription set, which contains one subscription-set member, replicates data from the registered source table named EMPLOYEE under the HR library to the target table named TGTEMPL under the TGTLIB library.

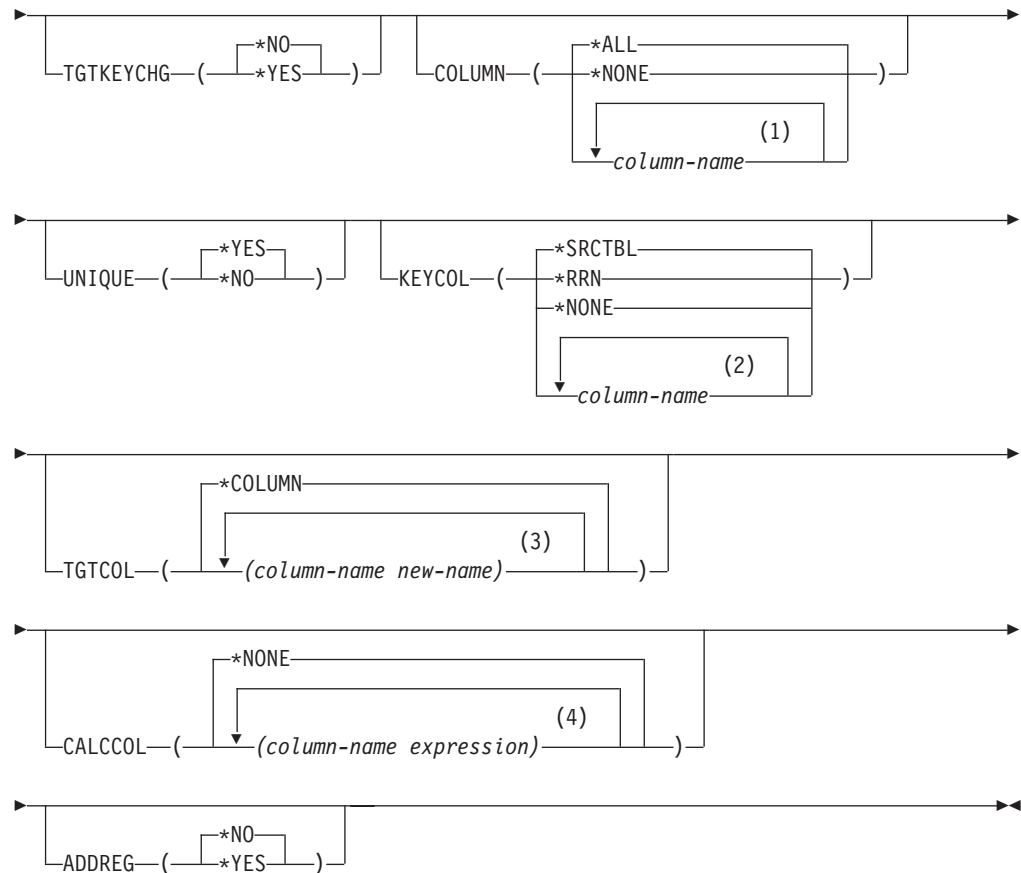
### Example 2:

To create a subscription set named SETHR with only two columns, EMPNO (the key) and NAME, from the registered source table named EMPLOYEE and replicate these columns to an existing target table named TGTEMPL:

```
ADDDPRSUB APYQUAL(AQHR) SETNAME(SETHR) SRCTBL(HR/EMPLOYEE)
TGTTBL(TGTLIB/TGTEMPL) CRTTGTTBL(*NO) COLUMN(EMPNO NAME) KEYCOL(EMPNO)
```

### Example 3:





**Note:**

- 1 You can specify up to 300 column names.
- 2 You can specify up to 120 column names.
- 3 You can specify up to 300 column names.
- 4 You can specify up to 100 column names and expressions.

Tabella 47 lists the invocation parameters.

*Tabella 47. ADDDPRSUBM command parameter definitions for System i*

Parameter	Definition and prompts
<b>APYQUAL</b>	Specifies the Apply qualifier that identifies which Apply program processes this subscription set. Subscription sets under an Apply qualifier run in a separate job. This parameter is required.  <i>apply-qualifier</i> The name of the Apply qualifier.
<b>SETNAME</b>	Specifies the name of the subscription set. This parameter is required.  <i>set-name</i> The name of the subscription set. The subscription-set name that you enter must be unique for the specified Apply qualifier or the ADDDPRSUBM command produces an error. Because the Apply program handles the set of target tables as a group, when one target table fails for any reason, the entire set fails.

Tabella 47. *ADDDPRSUBM* command parameter definitions for System *i* (Continua)

Parameter	Definition and prompts
SRCTBL	<p>Specifies the name of the table that is the source for this subscription-set member. You must register this table at the Capture control server before this table can become a member of a subscription set. This parameter is required.</p> <p><i>library-name/file-name</i> The qualified name of the source table.</p>
TGTTBL	<p>Specifies the name of the target table for this subscription-set member. The target table is automatically created if you set the <b>CRTTGTTBL</b> parameter to *YES and the target table does not already exist. This parameter is required.</p> <p><i>library-name/file-name</i> The qualified name of the target table.</p>
CTLSVR	<p>Specifies the relational database name of the system that contains the Apply control tables.</p> <p><b>*LOCAL (default)</b> The Apply control tables reside locally (on the machine from which you are running the <i>ADDDPRSUBM</i> command).</p> <p><i>rdb-name</i> The name of the relational database where the Apply control tables reside. You can use the Work with RDB Directory Entries (<i>WRKRDBDIRE</i>) command to find this name.</p>
SRCSVR	<p>Specifies the relational database name of the system that contains the Capture control tables.</p> <p><b>*LOCAL (default)</b> The source table is registered on the local machine (the machine from which you are running the <i>ADDDPRSUBM</i> command).</p> <p><i>rdb-name</i> The name of the relational database where the Capture control tables reside. You can use the Work with RDB Directory Entries (<i>WRKRDBDIRE</i>) command to find this name.</p>

Tabella 47. ADDDPRSUBM command parameter definitions for System i (Continua)

Parameter	Definition and prompts
TGTTYPE	<p>Specifies the target table type. These are SQL replication terms that describe the contents of the target table. After you create a target table as one of these types, you can use this parameter value on the <b>SRCTBL</b> parameter of the Add DPR Registration (ADDDPRREG) command to register this target table as a source table.</p> <p><b>*USERCOPY (default)</b>            The target table is a user copy, which is a target table with content that matches all or part of the content of a source table. A user copy is handled like a point-in-time table but does not contain any of the DB2 DataPropagator for System i system columns that are present in the point-in-time target table.</p> <p>This value is not valid when a value of *RRN is specified on the <b>KEYCOL</b> parameter.</p> <p>The table that you specified with the <b>SRCTBL</b> parameter must be one of the following types: user database, point-in-time table, or consistent-change data (CCD).</p> <p><b>Important:</b> If the target table already exists, DB2 DataPropagator for System i does not automatically journal changes to it. You must start journaling outside of DB2 DataPropagator for System i.</p> <p><b>*POINTINTIME</b>            The target table is a point-in-time table. A point-in-time table is a target table with content that matches all or part of the content of the source table and includes the DB2 DataPropagator for System i system column (IBMSNAP_LOGMARKER), which identifies when a particular row was inserted or updated at the Capture control server.</p> <p><b>*BASEAGR</b>            The target table is a base aggregate table, which is a target table that contains data that is aggregated (calculated) from a source table. The source table for a base aggregate target must be either a user table or a point-in-time table. This target table must contain the IBMSNAP_HLOGMARKER and IBMSNAP_LLOGMARKER system timestamp columns.</p> <p><b>*CHANGEAGR</b>            The table is a change aggregate table, which is a target table that contains data that is aggregated (calculated) based on the contents of a change-data (CD) table. This target table is created with the IBMSNAP_HLOGMARKER and IBMSNAP_LLOGMARKER system timestamp columns.</p>

Tabella 47. ADDDPRSUBM command parameter definitions for System i (Continua)

Parameter	Definition and prompts
TGTTYPE (Continued)	<p><b>*CCD</b></p> <p>The table is a consistent-change data (CCD) table, which is a target table created from a join of data in the change-data (CD) table and the unit-of-work (UOW) table. A CCD table provides transaction-consistent data for the Apply program and must include the following columns:</p> <ul style="list-style-type: none"> <li>• IBMSNAP_INTENTSEQ</li> <li>• IBMSNAP_OPERATION</li> <li>• IBMSNAP_COMMITSEQ</li> <li>• IBMSNAP_LOGMARKER</li> </ul> <p><b>*REPLICA</b></p> <p>The target table is a replica table, which is used only for update-anywhere replication. The replica target table receives changes from the master source table, and changes to the replica target table are propagated back to the master source table. A replica table is automatically registered as a source table.</p>
ROWSLT	<p>Specifies the predicates to be placed in an SQL WHERE clause. The Apply program uses these predicates to determine which rows in the change-data (CD) table of the source to apply to the target table. Use this parameter if you want only a subset of the source changes to be replicated to the target table.</p> <p><b>*ALL (default)</b></p> <p>The Apply program applies all changes in the CD table to the target table.</p> <p><i>WHERE-clause</i></p> <p>The SQL WHERE clause that specifies which rows from the CD table the Apply program applies to the target table. Do not include the WHERE keyword; it is implied on this parameter. This WHERE clause must be valid on the data server you are using to run the clause.</p> <p><b>Note:</b> The WHERE clause on this parameter is unrelated to any WHERE clauses specified on the SQLBEFORE or SQLAFTER parameters.</p>

Tabella 47. *ADDDPRSUBM* command parameter definitions for System i (Continua)

Parameter	Definition and prompts
<b>CRTTGTTBL</b>	<p>Specifies whether the target table (or view) is created.</p> <p><b>*YES (default)</b> Creates the target table (or view) if it does not exist. Otherwise, the existing table or view becomes the target, and the format of this existing table or view is checked if the value of the <b>CHKFMT</b> parameter is set to *YES. An additional index, with the values that you specified by the <b>UNIQUE</b> and <b>KEYCOL</b> parameters, is created for a target table if no such index currently exists. The command fails if an existing target table contains rows that violate the conditions of the additional index.</p> <p><b>*NO</b> Does not create the target table or view. You must create the table or view with the correct attributes before starting the Apply program.</p> <p>If the table or view exists and you set <b>CHKFMT</b> to *YES, the <i>ADDDPRSUBM</i> command ensures that the format of the existing table matches the subscription-set definition that you set. If <b>CHKFMT</b> is *NO, you must ensure that the format of the existing table matches the subscription-set definition.</p> <p><b>Important:</b> If the table or view already exists, DB2 DataPropagator for System i does not automatically journal changes to the existing object. You must start journaling outside of DB2 DataPropagator for System i.</p>
<b>CHKFMT</b>	<p>Specifies whether DB2 DataPropagator for System i checks the definition of the subscription-set member against the existing target table to ensure that the columns match. This parameter is ignored if the <b>CRTTGTTBL</b> parameter is *YES; this parameter is also ignored if the <b>CRTTGTTBL</b> parameter is set to *NO and the target table does not exist.</p> <p><b>*YES (default)</b> DB2 DataPropagator for System i verifies that the columns defined for this subscription-set member match the columns in the target table. This command fails if a mismatch is detected.</p> <p><b>*NO</b> DB2 DataPropagator for System i ignores differences between the subscription-set member and the existing target table. You must ensure that the target table is compatible with the subscription-set member.</p>

Tabella 47. ADDDPRSUBM command parameter definitions for System i (Continua)

Parameter	Definition and prompts
TGTKEYCHG	<p>Specifies how the Apply program handles updates when changes occur in source columns that are part of the target key columns for the target table. This parameter works in conjunction with the <b>USEDELINS</b> parameter on the ADDDPRREG command:</p> <ul style="list-style-type: none"> <li>• If <b>USEDELINS</b> is YES and <b>TGTKEYCHG</b> is YES, updates are not allowed.</li> <li>• If <b>USEDELINS</b> is YES and <b>TGTKEYCHG</b> is NO, updates become delete and insert pairs.</li> <li>• If <b>USEDELINS</b> is NO and <b>TGTKEYCHG</b> is YES, the Apply program handles this condition with special logic.</li> <li>• If <b>USEDELINS</b> is NO and <b>TGTKEYCHG</b> is NO, the Apply program processes the changes as normal updates.</li> </ul> <p><b>*NO (default)</b> Updates to the source table are staged by the Capture program and processed by the Apply program to the target table.</p> <p><b>*YES</b> The Apply program updates the target table based on the before images of the target key column, meaning that the Apply program changes the predicate to the old values instead of the new.</p>
COLUMN	<p>Specifies the columns to be included in the target table. The column names must be unqualified. Choose the column names from the list of column names that you specified on the <b>CAPCOL</b> parameter when you registered the source table.</p> <p>If you set the <b>IMAGE</b> parameter to <b>*BOTH</b> when registering this table, you can specify before-image column names. The before-image column names are the original column names with a prefix. This prefix is the character that you specified in the <b>PREFIX</b> parameter of the ADDDPRREG command.</p> <p><b>*ALL (default)</b> All of the columns that you registered in the source are included in the target table.</p> <p><b>*NONE</b> No columns from the source table are included in the target table. You can use <b>*NONE</b> when you want only computed columns in the target table. This value is required if the <b>CALCCOL</b> parameter contains summary functions but no grouping is performed.</p> <p><i>column-name</i> The names of up to 300 source columns that you want to include in the target table. Separate the column names with spaces.</p>



Tabella 47. *ADDDPRSUBM* command parameter definitions for System i (Continua)

Parameter	Definition and prompts
UNIQUE	<p>Specifies whether the target table has unique keys as indicated by the <b>KEYCOL</b> parameter.</p> <p><b>*YES (default)</b></p> <p>The target table supports one net change per key; only one row exists in the target table for that key regardless of how many changes are made to the key.</p> <p>This value specifies that the table contains current data rather than a history of changes to the data. A condensed table includes no more than one row for each primary key value in the table and can be used to supply current information for a refresh.</p> <p><b>*NO</b></p> <p>The target table supports multiple changes per key. The changes are appended to the target table.</p> <p>This value specifies that the table contains a history of changes to the data rather than current data. A non-condensed table includes more than one row for each key value in the table and can be used to supply a history of changes to the data. A non-condensed table cannot supply current data for a refresh.</p>

Tabella 47. ADDDPRSUBM command parameter definitions for System i (Continua)

Parameter	Definition and prompts
KEYCOL	<p>Specifies columns that describe the key of the target table. The column names must be unqualified. For *POINTINTIME, *REPLICA, and *USERCOPY target tables (as specified on the TGTTYPE parameter), you must identify one or more columns as the target key for the target table. This target key is used by the Apply program to identify each unique row that changes during change-capture replication.</p> <p><b>*SRCTBL (default)</b></p> <p>The key columns in the target table are the same as those in the source table. The ADDDPRREG command uses the key that is specified in the source table if the source table has a key. The following key columns are used:</p> <ul style="list-style-type: none"> <li>• Key columns that you defined through DDS when creating the table with the Create Physical File (CRTPF) command</li> <li>• Primary and unique keys that you defined with the CREATE TABLE and ALTER TABLE SQL statements</li> <li>• Unique keys that you defined with the CREATE INDEX SQL statements</li> </ul> <p>If you use a column as a key more than once and with different ordering, the target table key is defined with ascending order.</p> <p><b>*RRN</b></p> <p>The key column in the target table is the IBMQSQ_RRN column. The target table is created with an IBMQSQ_RRN column, and this column is used as the key. When the Apply program runs, if the source table is a user table and the target table is a point-in-time table or user copy, the IBMQSQ_RRN column in the target table is updated with the relative record number of the associated record in the source table. Otherwise, the IBMQSQ_RRN column in the target table is updated with the value of the IBMQSQ_RRN column in the source table.</p> <p><b>*NONE</b></p> <p>The target copy does not contain a target key. You cannot specify *NONE if the target table type is *POINTINTIME, *REPLICA, or *USERCOPY.</p> <p><i>column-name</i></p> <p>The names of the target columns that you want to use as the target key columns. You can specify up to 120 column names. Separate the column names with spaces.</p>

Tabella 47. ADDDPRSUBM command parameter definitions for System i (Continua)

Parameter	Definition and prompts
TGTCOL	<p>Specifies the new names for all the columns that the Apply program updates in the target table. These names override the column names taken from the source table. The column names must be unqualified. If you specified a value of *NONE for the <b>COLUMN</b> parameter, do not use the <b>TGTCOL</b> parameter.</p> <p>Use this parameter to give more meaningful names to the target table columns. Specify each source column name and the name of the corresponding column on the target table.</p> <p><b>*COLUMN (default)</b> The target columns are the same as the columns you specified in the <b>COLUMN</b> parameter.</p> <p><i>column-name</i> The column names from the source table that you want to change at the target. You can list up to 300 column names.</p> <p><i>new-name</i> The new names of the target columns. You can list up to 300 new column names. If you do not use this parameter, the name of the column on the target table will be the same as the source column name.</p>
CALCCOL	<p>Specifies the list of user-defined or calculated columns in the target table. The column names must be unqualified. Enclose each column name and expression pair in parenthesis.</p> <p>You must specify a column name for each SQL expression. If you want to define any column as an SQL expression without a GROUP BY clause, you must set the <b>COLUMN</b> parameter to *NONE.</p> <p><b>*NONE (default)</b> No user-defined or calculated columns are included in the target table.</p> <p><i>column-name</i> The column names of the user-defined or calculated columns in the target table. You can list up to 100 column names.</p> <p><i>expression</i> The expressions for the user-defined or calculated columns in the target table. You can list up to 100 SQL column expressions.</p>
ADDREG	<p>Specifies whether the target table is automatically registered as a source table. Use this parameter to register CCD target type tables.</p> <p><b>*NO (default)</b> The target table is not registered as a source table. DB2 DataPropagator for System i ignores this parameter value if the target type is *REPLICA. Replica target tables are always automatically registered as source tables.</p> <p><b>*YES</b> The target table is registered as a source table. This command fails if you already registered the target table.</p> <p>Do not set this parameter to *YES if the target table type is *USERCOPY, *POINTINTIME, *BASEAGR, or *CHANGEAGR.</p> <p>If you set the <b>CRTTGTTBL</b> parameter to *NO, you must create the target table before attempting to register it as a source.</p>

## Examples for ADDDPRSUBM

The following examples illustrate how to use the ADDDPRSUBM command.

### Example 1:

To add a subscription-set member to a subscription set named SETHR under the AQHR Apply qualifier:

```
ADDDPRSUBM APYQUAL(AQHR) SETNAME(SETHR) SRCTBL(HR/YTDTAX) TGTTBL(TGTHR/TGTTAX)
```

### Example 2:

To add a subscription-set member with only two columns, AMOUNT and NAME, from the registered source table named YTDTAX and to replicate these columns to an existing target table named TGTTAX:

```
ADDDPRSUBM APYQUAL(AQHR) SETNAME(SETHR) SRCTBL(HR/YTDTAX) TGTTBL(TGTLIB/TGTTAX)
  CRTTGTTBL(*NO) COLUMN(AMOUNT NAME) CHKFM(*YES)
```

This command verifies that the AMOUNT and NAME columns defined for this subscription-set member match the columns in the target table.

### Example 3:

To add a subscription-set member to subscription set named SETHR and to replicate this data to a consistent-change data target table named TGTYTD:

```
ADDDPRSUBM APYQUAL(AQHR) SETNAME(SETHR) SRCTBL(HR/YTDTAX) TGTTBL(TGTLIB/TGTYTD)
  TGTTYPE(*CCD) ADDREG (*YES)
```

This command registers the target table as a source table for DB2 DataPropagator for System i.

---

## ANZDPR: Operating the Analyzer (System i)

### System i

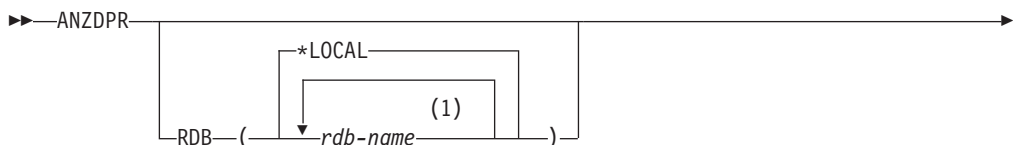
Use the Analyze DPR (ANZDPR) command to analyze a failure from a Capture or Apply program, to verify the setup of your replication configuration, or to obtain problem diagnosis and performance tuning information.

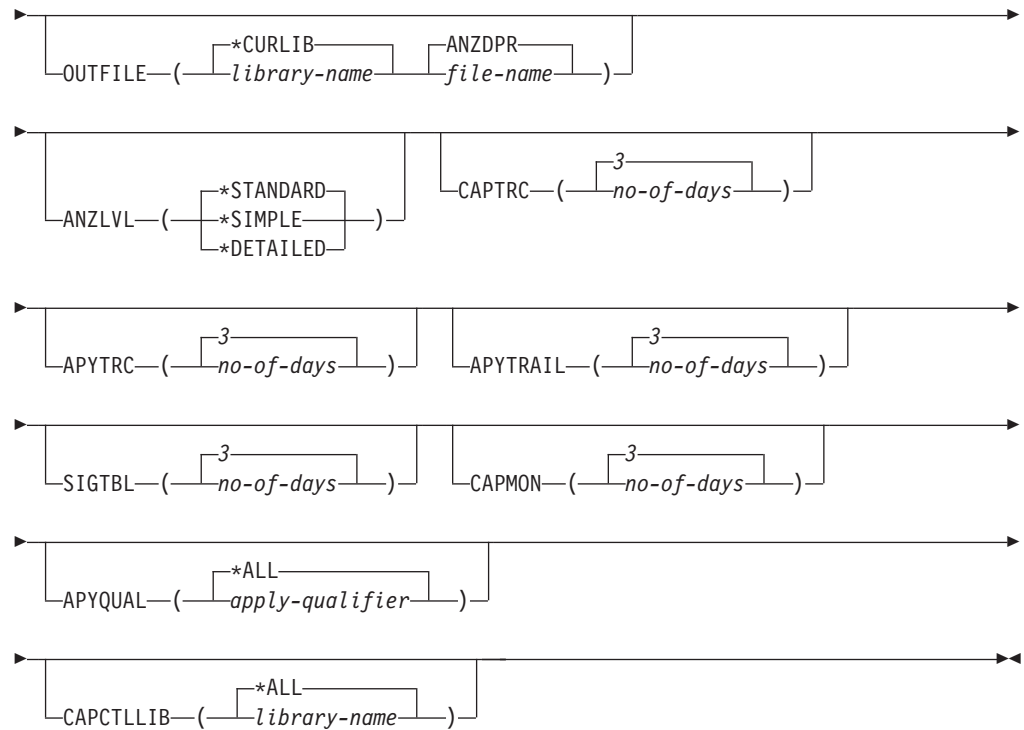
Run this command after you set up your replication configuration.

After you type the command name on the command line, you can press the F4 key to display the command syntax.

To display a complete description of this command and all of its parameters, move the cursor to the command at the top of the screen and press the F1 key. To display a description of a specific parameter, place the cursor on that parameter and press the F1 key.

### Syntax





**Note:**

1 You can specify up to 10 databases.

Tabella 48 lists the invocation parameters.

*Tabella 48. ANZDPR command parameter definitions for System i*

Parameter	Definition and prompts
<b>RDB</b>	<p>Specifies the databases to be analyzed.</p> <p><b>*LOCAL (default)</b> The database on your local system.</p> <p><i>rdb-name</i> The RDB Directory Entry name, which indicates the database.</p> <p>You can enter up to 10 databases. If you want to analyze multiple databases including the database on your local system, make sure that *LOCAL is the first entry in the list. Also, verify that you can connect to all these databases from your current system.</p>

Tabella 48. ANZDPR command parameter definitions for System i (Continua)

Parameter	Definition and prompts
OUTFILE	<p>Specifies the library and file name used to store the analyzer output. This command writes the output to an HTML file.</p> <p><b>*CURLIB (default)</b> The current library.</p> <p><i>library-name</i> The name of the library.</p> <p><b>ANZDPR (default)</b> The output is written to an HTML file named ANZDPR.</p> <p><i>file-name</i> The name of the HTML output file.</p> <p>If the file name already exists, the file is overwritten. If the file name does not exist, the command creates the file with attributes of RCDLEN(512) and SIZE(*NOMAX).</p>
ANZLVL	<p>Specifies the level of analysis to be reported. The level of analysis can be:</p> <p><b>*STANDARD (default)</b> Generates a report that includes the contents of the control tables as well as Capture and Apply program status information.</p> <p><b>*SIMPLE</b> Generates the information in the standard report but excludes subcolumn details. Use this option if you want to generate a smaller report that requires less system resources.</p> <p><b>*DETAILED</b> Generates a report with the most complete analysis. The detailed report includes the information from the standard report in addition to subscription set information.</p>
CAPTRC	<p>Specifies the date range (0 to 30 days) of entries to be reported from the IBMSNAP_CAPTRACE table. The default is 3.</p> <p><i>no-of-days</i> The number of days to be reported.</p>
APYTRC	<p>Specifies the date range (0 to 30 days) of entries to be reported from the IBMSNAP_APPLYTRACE table. The default is 3.</p> <p><i>no-of-days</i> The number of days to be reported.</p>
APYTRAIL	<p>Specifies the date range (0 to 30 days) of entries to be reported from the IBMSNAP_APPLYTRAIL table. The default is 3.</p> <p><i>no-of-days</i> The number of days to be reported.</p>
SIGTBL	<p>Specifies the date range (0 to 30 days) of entries to be reported from the IBMSNAP_SIGNAL table. The default is 3.</p> <p><i>no-of-days</i> The number of days to be reported.</p>
CAPMON	<p>Specifies the date range (0 to 30 days) of entries to be reported from the IBMSNAP_CAPMON table. The default is 3.</p> <p><i>no-of-days</i> The number of days to be reported.</p>

Tabella 48. ANZDPR command parameter definitions for System i (Continua)

Parameter	Definition and prompts
APYQUAL	<p>Specifies the Apply qualifiers to be analyzed.</p> <p><b>*ALL (default)</b> All Apply qualifiers are analyzed.</p> <p><i>apply-qualifier</i> The name of the Apply qualifier to be analyzed. You can enter up to 10 Apply qualifiers.</p>
CAPCTLLIB	<p>Specifies the Capture schemas, which are the names of the Capture control libraries that you want to analyze. You can analyze a specific Capture control library, or you can choose the default of *ALL to analyze all the Capture control libraries.</p> <p><b>*ALL (default)</b> All of the Capture control libraries will be analyzed.</p> <p><i>library-name</i> The name of the specific Capture control library that you want to analyze.</p>

## Examples for ANZDPR

The following examples illustrate how to use the ANZDPR command.

### Example 1:

To run the Analyzer on both your local database and a remote database named RMTRDB1 using a standard level of analysis:

```
ANZDPR RDB(*LOCAL RMTRDB1) OUTFILE(MYLIB/ANZDPR) ANZLVL(*STANDARD) CAPTRC(1)
  APYTRC(1) APYTRAIL(1) SIGTBL(1) CAPMON(1) APYQUAL(*ALL)
```

This example generates one day of entries from the IBMSNAP\_CAPTRACE, IBMSNAP\_APPLYTRACE, IBMSNAP\_APPLYTRAIL, IBMSNAP\_SIGNAL, and IBMSNAP\_CAPMON tables for all Apply qualifiers and writes the output to an HTML file named ANZDPR in the library called MYLIB.

### Example 2:

To run the Analyzer with all default values:

```
ANZDPR
```

## CHGDPRCAPA: Changing DPR Capture attributes (System i)

### System i

Use the Change DPR Capture Attributes (CHGDPRCAPA) command to change the global operating parameters that are used by the Capture program and are stored in the IBMSNAP\_CAPPARMS table.

These parameter changes do not take effect until you perform one of the following actions:

- Issue an INZDPRCAP command.
- End and then restart the Capture program.

After you type the command name on the command line, you can press the F4 key to display the command syntax.

To display a complete description of this command and all of its parameters, move the cursor to the command at the top of the screen and press the F1 key. To display a description of a specific parameter, place the cursor on that parameter and press the F1 key.

## Syntax

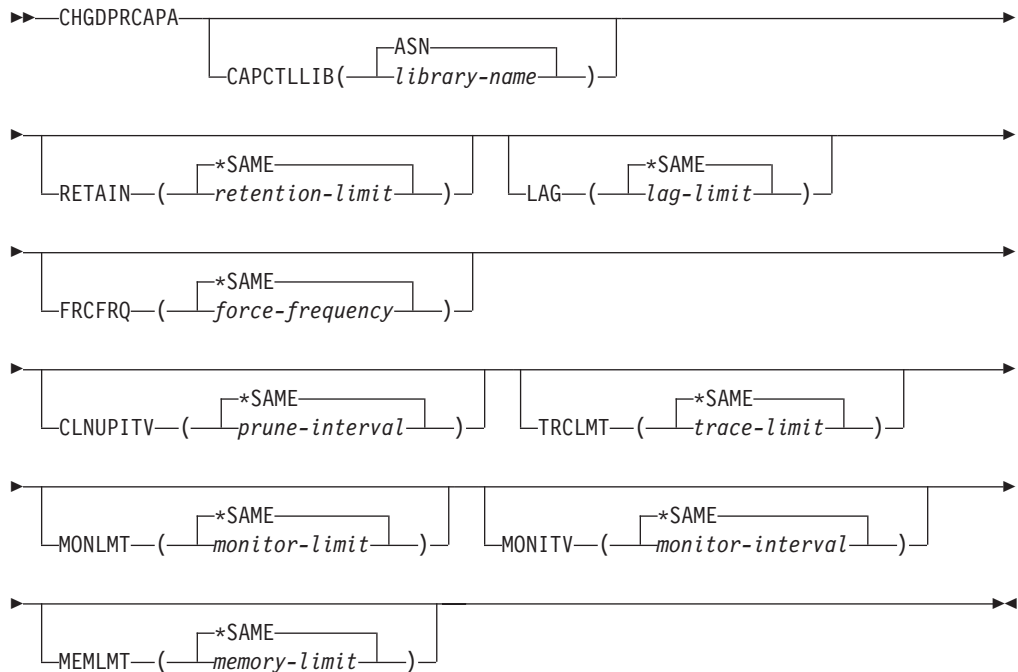


Tabella 49 lists the invocation parameters.

Tabella 49. CHGDPRCAPA command parameter definitions for System i

Parameter	Definition and prompts
<b>CAPCTLLIB</b>	Specifies the Capture schema, which is the name of the library in which the Capture control tables reside. <b>ASN (default)</b> The Capture control tables are in the ASN library. <i>library-name</i> The name of a library that contains the Capture control tables.



Tabella 49. CHGDPRCAPA command parameter definitions for System i (Continua)

Parameter	Definition and prompts
<b>RETAIN</b>	<p data-bbox="695 254 1458 394">Specifies the new retention limit, which is the number of minutes that data is retained in the change-data (CD), unit-of-work (UOW), IBMSNAP_SIGNAL, and IBMSNAP_AUTHTKN tables before this data is removed. This value is stored in the RETENTION_LIMIT column of the IBMSNAP_CAPPARMS table.</p> <p data-bbox="695 426 1458 537">This value works with the <b>CLNUPITV</b> parameter value. When the <b>CLNUPITV</b> value is reached, the CD, UOW, IBMSNAP_SIGNAL, and IBMSNAP_AUTHTKN data is removed if this data is older than the retention limit.</p> <p data-bbox="695 569 1458 680">Ensure that the Apply intervals are set to copy changed information before the data reaches this <b>RETAIN</b> parameter value to prevent inconsistent data in your tables. If the data becomes inconsistent, the Apply program performs a full refresh.</p> <p data-bbox="695 711 1458 762">The default is 10 080 minutes (seven days). The maximum is 35000000 minutes.</p> <p data-bbox="695 783 1027 833"><b>*SAME (default)</b> This value is not changed.</p> <p data-bbox="695 854 1070 905"><i>retention-limit</i> The new retention limit value.</p>
<b>LAG</b>	<p data-bbox="695 926 1458 1037">Specifies the new lag limit, which is the number of minutes that the Capture program can fall behind in processing before restarting. This value is stored in the LAG_LIMIT column of the IBMSNAP_CAPPARMS table.</p> <p data-bbox="695 1068 1458 1209">When the lag limit is reached (that is, when the timestamp of the journal entry is older than the current timestamp minus the lag limit), the Capture program initiates a cold start for the tables that it is processing for that journal. The Apply program then performs a full refresh to provide the Capture program with a new starting point.</p> <p data-bbox="695 1241 1458 1291">The default is 10 080 minutes (seven days). The maximum is 35000000 minutes.</p> <p data-bbox="695 1312 1027 1362"><b>*SAME (default)</b> This value is not changed.</p> <p data-bbox="695 1383 1008 1442"><i>lag-limit</i> The new lag limit value.</p>

Tabella 49. CHGDPRCAPA command parameter definitions for System i (Continua)

Parameter	Definition and prompts
<b>FRCFRQ</b>	<p>Specifies how often (from 30 to 600 seconds) the Capture program writes changes to the change-data (CD) and UOW tables. This value is stored in the COMMIT_INTERVAL column of the IBMSNAP_CAPPARMS table.</p> <p>The Capture program makes these changes available to the Apply program either when the buffers are filled or when the <b>FRCFRQ</b> time limit expires, whichever is sooner.</p> <p>Use this parameter to make changes more readily available to the Apply program on servers with a low rate of source table changes. The <b>FRCFRQ</b> parameter value is a global value used for all defined source tables. Setting the <b>FRCFRQ</b> value to a low number can affect system performance.</p> <p>The default is 30 seconds.</p> <p><b>*SAME (default)</b> This value is not changed.</p> <p><i>force-frequency</i> The new commit interval value, which is the number of seconds that the Capture program keeps CD and UOW table changes in buffer space before making these changes available to the Apply program.</p>
<b>CLNUPITV</b>	<p>Specifies the maximum amount of time (in hours) before the Capture program prunes old records from the change-data (CD), UOW, IBMSNAP_SIGNAL, IBMSNAP_CAPMON, IBMSNAP_CAPTRACE, and IBMSNAP_AUTHTKN tables.</p> <p>This parameter works in conjunction with the <b>RETAIN</b> parameter to control pruning of the CD, UOW, IBMSNAP_SIGNAL, and IBMSNAP_AUTHTKN tables, with the <b>MONLMT</b> parameter to control pruning of the IBMSNAP_CAPMON table, and with the <b>TRCLMT</b> parameter to control pruning of the IBMSNAP_CAPTRACE table. (Use the STRDPRCAP command to set the <b>RETAIN</b>, <b>MONLMT</b>, and <b>TRCLMT</b> parameters for a Capture program.)</p> <p>The value of this parameter is automatically converted from hours to seconds and is stored in the PRUNE_INTERVAL column of the IBMSNAP_CAPPARMS table. If the PRUNE_INTERVAL column is changed manually (not by using the CHGDPRCAPA command), you might see changes because of rounding when you prompt by using the F4 key.</p> <p><b>*SAME (default)</b> This Capture attribute value is not changed.</p> <p><i>prune-interval</i> The pruning interval expressed as a specific number of hours (1 to 100).</p>

Tabella 49. CHGDPRCAPA command parameter definitions for System i (Continua)

Parameter	Definition and prompts
TRCLMT	<p>Specifies the trace limit (in minutes). This value is stored in the TRACE_LIMIT column of the IBMSNAP_CAPPARMS table.</p> <p>The Capture programs prune any IBMSNAP_CAPTRACE rows that are older than the trace limit. The default is 10 080 minutes (seven days of trace entries).</p> <p><b>*SAME (default)</b> This value is not changed.</p> <p><i>trace-limit</i> The number of minutes of trace data kept in the IBMSNAP_CAPTRACE table after pruning.</p>
MONLMT	<p>Specifies the monitor limit (in minutes). This value is stored in the MONITOR_LIMIT column of the IBMSNAP_CAPPARMS table.</p> <p>The Capture program prunes any IBMSNAP_CAPMON rows that are older than the monitor limit.</p> <p>The default is 10 080 minutes (seven days of monitor entries).</p> <p><b>*SAME (default)</b> This value is not changed.</p> <p><i>monitor-limit</i> The number of minutes of monitor data kept in the IBMSNAP_CAPMON table after pruning.</p>
MONITV	<p>Specifies how frequently (in seconds) the Capture program inserts rows into the IBMSNAP_CAPMON table. This value is stored in the MONITOR_INTERVAL column of the IBMSNAP_CAPPARMS table.</p> <p>The default is 300 seconds (five minutes).</p> <p><b>*SAME (default)</b> This value is not changed.</p> <p><i>monitor-interval</i> The number of seconds between row insertion into the IBMSNAP_CAPMON table. The monitor interval must be at least 120 seconds (two minutes). If you specify a number that is less than 120, this command automatically sets this parameter value to 120.</p>
MEMLMT	<p>Specifies the maximum size (in megabytes) of memory that the Capture journal job can use. This value is stored in the MEMORY_LIMIT column of the IBMSNAP_CAPPARMS table.</p> <p>The default is 32 megabytes.</p> <p><b>*SAME (default)</b> This value is not changed.</p> <p><i>memory-limit</i> The maximum number of megabytes for memory.</p>

## Examples for CHGDPRCAPA

The following examples illustrate how to use the CHGDPRCAPA command.

### Example 1:



Tabella 50 lists the invocation parameters.

Tabella 50. CRTDPRTBL command parameter definitions for System i

Parameter	Definition and prompts
CAPCTLLIB	Specifies the Capture schema, which is the name of the library where the newly created Capture control tables are placed.  <b>ASN (default)</b> The Capture control tables are placed in the ASN library.  <i>library-name</i> The name of the library where the Capture control tables are placed.

## Examples for CRTDPRTBL

The following examples illustrate how to use the CRTDPRTBL command.

### Example 1:

To create new replication control tables in the default ASN library:

```
CRTDPRTBL CAPCTLLIB(ASN)
```

### Example 2:

To create new replication control tables for a Capture schema called DPRSALES:

```
CRTDPRTBL CAPCTLLIB(DPRSALES)
```

## ENDDPRAPY: Stopping Apply (System i)

### System i

Use the End DPR Apply (ENDDPRAPY) command to stop an Apply program on your local system.

You should stop the Apply program before any planned system down time. You might also want to end the Apply program during periods of peak system activity.

After you type the command name on the command line, you can press the F4 key to display the command syntax.

To display a complete description of this command and all of its parameters, move the cursor to the command at the top of the screen and press the F1 key. To display a description of a specific parameter, place the cursor on that parameter and press the F1 key.

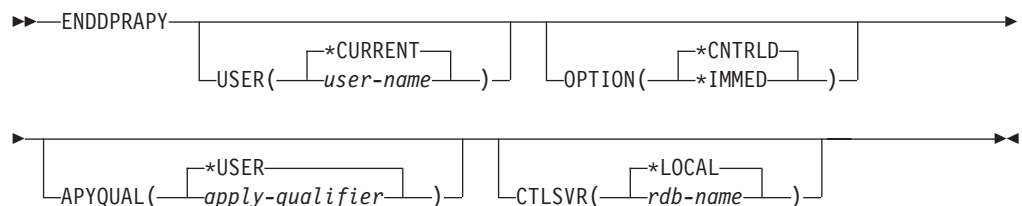


Tabella 51 lists the invocation parameters.

Tabella 51. ENDDPRAPY command parameter definitions for System i

Parameter	Definition and prompts
<b>USER</b>	<p>This parameter is ignored unless the <b>APYQUAL</b> parameter has a value of <b>*USER</b>, in which case this parameter specifies the Apply qualifier associated with the Apply program.</p> <p><b>*CURRENT (default)</b> The Apply program of the user associated with the current job.</p> <p><i>user-name</i> The Apply program of the specified user.</p> <p>When prompting on the ENDDPRAPY command, you can press the F4 key to see a list of users who defined subscriptions.</p>
<b>OPTION</b>	<p>Specifies how to stop the Apply program.</p> <p><b>*CNTRLD (default)</b> The Apply program completes all of its tasks before stopping. These tasks might take a considerable period of time if the Apply program is completing a subscription set.</p> <p><b>*IMMED</b> The Apply program completes all of its tasks with the ENDJOB OPTION(*IMMED) command. The tasks end immediately, without any cleanup. Use this option only after a controlled end is unsuccessful, because it can cause undesirable results. (Unless the Apply program was asleep when you issued the ENDDPRAPY command, you should verify the target table contents.)</p> <p>If the Apply program was performing a full refresh to the target table, the target table might be empty as a result of ending the Apply program before the table was refreshed with the source table contents. If the target table is empty, you must force a full refresh for this replication target.</p> <p>You might find that a subscription set is considered IN USE (the STATUS column in the IBMSNAP_SUBS_SET table has a value of 1). If it is, reset the value to 0 or -1. This allows the subscription set to be run again by the Apply program.</p>
<b>APYQUAL</b>	<p>Specifies the Apply qualifier that is used by the Apply program.</p> <p><b>*USER (default)</b> The user name specified on the <b>USER</b> parameter is the Apply qualifier.</p> <p><i>apply-qualifier</i> The name used to group the subscription sets that this Apply program runs. You can specify a maximum of 18 characters for the Apply qualifier name. This name follows the same naming conventions as a relational database name. You identify the subscriptions being run by the records in the IBMSNAP_SUBS_SET table with this value in the APPLY_QUAL column.</p> <p>When prompting on the ENDDPRAPY command, you can press the F4 key to see a list of Apply qualifier names with existing subscriptions.</p>

Tabella 51. ENDDPRAPY command parameter definitions for System i (Continua)

Parameter	Definition and prompts
CTLSVR	<p>Specifies the relational database name of the system that contains the Apply control tables.</p> <p><b>*LOCAL (default)</b> The Apply control tables reside locally (from the machine on which you are running the ENDDPRAPY command).</p> <p><i>rdb-name</i> The name of the relational database where the Apply control tables reside. You can use the Work with RDB Directory Entries (WRKRDBDIRE) command to find this name.</p> <p>When prompting on the ENDDPRAPY command, you can press the F4 key to choose from the list of databases in the RDB directory.</p>

## Usage notes

The ENDDPRAPY command uses the value of the **APYQUAL** and **CTLSVR** parameters to search the IBMSNAP\_APPLY\_JOB table for the job name, job number, and job user for the referenced Apply program, and ends that job.

ENDDPRAPY issues an error message if the following conditions occur:

- If the IBMSNAP\_APPLY\_JOB table does not exist or is corrupted.
- If there is no record in the IBMSNAP\_APPLY\_JOB table for the Apply qualifier and control server name.
- If the Apply job already ended.
- If the user ID running the command is not authorized to end the Apply job.

## Examples for ENDDPRAPY

The following examples illustrate how to use the ENDDPRAPY command.

### Example 1:

To end the Apply program that uses the AQHR Apply qualifier:

```
ENDDPRAPY OPTION(*CNTRLD) APYQUAL(AQHR)
```

The Apply program ends after all of its tasks are completed.

### Example 2:

To end the Apply program immediately:

```
ENDDPRAPY OPTION(*IMMED) APYQUAL(AQHR)
```

The tasks of the Apply program end immediately, without any cleanup.

### Example 3:

To end an Apply program that uses Apply control tables that reside on a relational database named DB1X:

```
ENDDPRAPY OPTION(*CNTRLD) APYQUAL(AQHR) CTLSVR(DB1X)
```

## ENDDPRCAP: Stopping Capture (System i)

### System i

Use the End DPR Capture (ENDDPRCAP) command to stop the Capture program.

Use this command to stop the Capture program before shutting down the system. You might also want to stop the program during periods of peak system use to increase the performance of other programs that run on the system.

After you type the command name on the command line, you can press the F4 key to display the command syntax.

To display a complete description of this command and all of its parameters, move the cursor to the command at the top of the screen and press the F1 key. To display a description of a specific parameter, place the cursor on that parameter and press the F1 key.

### Syntax

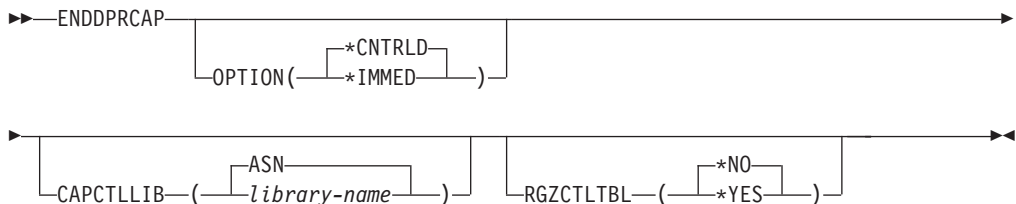


Tabella 52 lists the invocation parameters.

Tabella 52. ENDDPRCAP command parameter definitions for System i

Parameter	Definition and prompts
<b>OPTION</b>	<p>Specifies how to stop the Capture program.</p> <p><b>*CNTRLD (default)</b> The Capture program stops normally after completing all tasks. The ENDDPRCAP command might take longer when you specify the *CNTRLD option because the Capture program completes all of its subordinate processes before stopping.</p> <p><b>*IMMED</b> The Capture program stops normally after completing all tasks with the ENDJOB OPTION(*IMMED) command.</p>
<b>CAPCTLLIB</b>	<p>Specifies the Capture schema, which is the name of the library in which the Capture control tables are located. This library includes the IBMSNAP_REGISTER table, which stores the registration information of the source tables.</p> <p><b>ASN (default)</b> The Capture control tables are in the ASN library. The ASN library is the default library.</p> <p><i>library-name</i> The name of a library that contains the Capture control tables.</p>



Tabella 52. ENDDPRCAP command parameter definitions for System i (Continua)

Parameter	Definition and prompts
RGZCTLTBL	<p>Specifies whether a Reorganize Physical File Member (RGZPFM) command is performed on the control tables (including the change-data (CD) and unit-of-work (UOW) tables) when the Capture program ends. The system does not recover disk space unless the RGZPFM command process is performed on the tables. The RGZPFM command will not be performed if the control tables are being accessed by an Apply program or by other application programs.</p> <p><b>*NO (default)</b> The RGZPFM command is not performed.</p> <p><b>*YES</b> The RGZPFM command is performed.</p>

## Usage notes

If you use the ENDJOB command, temporary objects might be left in the QDP4 library. These objects have the types \*DTAQ and \*USRSPC, and are named QDP4nnnnnn, where nnnnnn is the job number of the job that used them. You can delete these objects when the job that used them (identified by the job number in the object name) is not active.

If the job under the Capture control library will not end after issuing this command, use the ENDJOB command with \*IMMED option to end this job and all the journal jobs running in the DB2 DataPropagator for System i subsystem. Do not end Apply jobs running in the same subsystem if you want to end only the Capture program.

In rare cases when the Capture control job ends abnormally, the journal jobs created by Capture control job (which is named according to the CAPCTLLIB parameter) might still be left running. The only way to end these jobs is to use the ENDJOB command with either the \*IMMED or \*CNTRLD option.

## Examples for ENDDPRCAP

The following examples illustrate how to use the ENDDPRCAP command.

### Example 1:

To end the Capture program, which uses Capture control tables in the ASN library, after all processing tasks are completed:

```
ENDDPRCAP OPTION(*CNTRLD) CAPCTLLIB(ASN) RGZCTLTBL(*NO)
```

### Example 2:

To end the Capture program immediately for the Capture schema BSN:

```
ENDDPRCAP OPTION(*IMMED) CAPCTLLIB(BSN) RGZCTLTBL(*NO)
```

### Example 3:

To end the Capture program after all processing tasks are completed and to reorganize the Capture control tables:

```
ENDDPRCAP OPTION(*CNTRLD) CAPCTLLIB(ASN) RGZCTLTBL(*YES)
```

## GRTDPRAUT: Authorizing users (System i)

### System i

Use the Grant DPR Authority (GRTDPRAUT) command to authorize a list of users to access the replication control tables in order to run the Capture and Apply programs.

For example, the authority requirements for the user who is running the Capture and Apply programs might differ from the authority requirements for the user who defines replication sources and targets.

You must have \*ALLOBJ authority to grant authorities.

After you type the command name on the command line, you can press the F4 key to display the command syntax.

To display a complete description of this command and all of its parameters, move the cursor to the command at the top of the screen and press the F1 key. To display a description of a specific parameter, place the cursor on that parameter and press the F1 key.

### Syntax

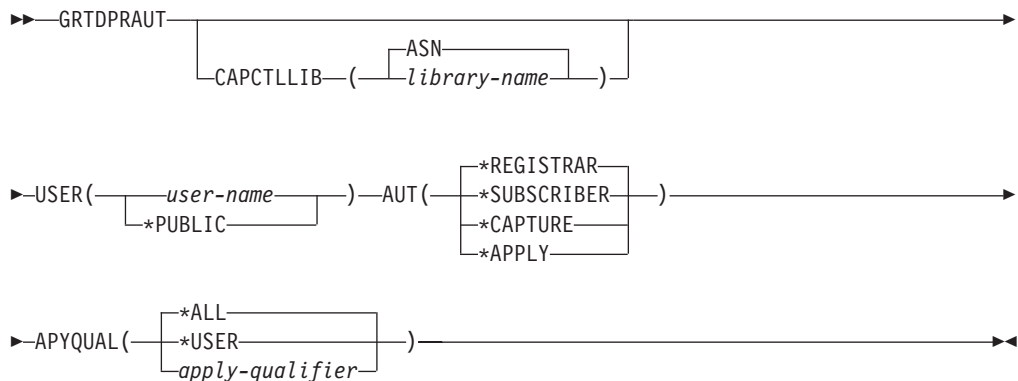


Tabella 53 lists the invocation parameters.

Tabella 53. GRTDPRAUT command parameter definitions for System i

Parameter	Definition and prompts
CAPCTLLIB	Specifies the Capture schema, which is the library that contains the replication control tables to which the user is granted authority.
ASN (default)	The Capture control tables reside in the ASN library.
library-name	The name of the library that contains the replication control tables.

Tabella 53. GRTDPRAUT command parameter definitions for System i (Continua)

Parameter	Definition and prompts
<b>USER</b>	<p>Specifies the users who have authority.</p> <p><i>user-name</i> The names of up to 50 users who have authority.</p> <p><b>*PUBLIC</b> Indicates that *PUBLIC authority is granted to the file, but (if insufficient for the task) is used only for those users who have no specific authority, who are not on the authorization list associated with the file, and whose group profile does not have any authority.</p>
<b>AUT</b>	<p>Specifies the type of authority being granted.</p> <p><b>*REGISTRAR (default)</b> The users are granted the authorities to define, change, and remove registrations.  For a complete list of authorities with AUT(*REGISTRAR), see Tabella 54 a pagina 370.</p> <p><b>*SUBSCRIBER</b> The users are granted authority to define, change, and remove subscription sets.  For a complete list of authorities with AUT(*SUBSCRIBER), see Tabella 55 a pagina 371.</p> <p><b>*CAPTURE</b> The users are granted authority to run the Capture program.  For a complete list of authorities granted with AUT(*CAPTURE), see Tabella 56 a pagina 372.</p> <p><b>*APPLY</b> The users are granted authority to run the Apply program.  The command does not grant authority to any of the objects that reside on other databases accessed by the Apply program.  When an Apply program is invoked, the user associated with the DRDA application server job must also be granted *APPLY authority. If the source is a System i server, you should run the GRTDPRAUT command on the source server system, with the application server job user specified on the <b>USER</b> parameter and the Apply qualifier specified on the <b>APYQUAL</b> parameter.  Authorities are not granted to the target tables unless the target server is the same as the control server and both reside on the system where the command is run.  For a complete list of authorities granted with AUT(*APPLY), see Tabella 57 a pagina 374.</p>

Tabella 53. GRTDPRAUT command parameter definitions for System i (Continua)

Parameter	Definition and prompts
APYQUAL	<p>Specifies the Apply qualifier to be used by the user as specified with the <b>USER</b> parameter. This parameter is used only when AUT(*APPLY) or AUT(*SUBSCRIBER) is specified.</p> <p><b>*ALL (default)</b> The user is granted authority to run the Apply program or to define and remove subscription sets for all Apply qualifiers.</p> <p><b>*USER</b> The users specified on the <b>USER</b> parameter are granted authority to the subscription sets with an Apply qualifier that is the same as the user name.</p> <p><i>apply-qualifier</i> The user is granted authority to run the Apply program or define and remove subscription sets for the Apply qualifiers associated with this Apply qualifier.</p> <ul style="list-style-type: none"> <li>• The user is granted authority to all replication sources, change-data (CD) tables, and consistent-change data (CCD) tables associated with records in the IBMSNAP_PRUNCNTL table that have a value in the APPLY_QUAL column matching the value input with the <b>APYQUAL</b> parameter.</li> <li>• The user is granted authority to the subscription sets listed in the IBMSNAP_SUBS_MEMBR table that reside on this system.</li> </ul>

## Usage notes

You cannot use the GRTDPRAUT command while the Capture or Apply programs are running, or when applications that use the source tables are active because authorizations cannot be changed on files that are in use.

The following tables list the authorities granted when you specify:

- AUT(\*REGISTRAR)
- AUT\*(SUBSCRIBER)
- AUT(\*CAPTURE)
- AUT(\*APPLY)

on the GRTDPRAUT command.

The following table lists the authorities granted when you specify the AUT(\*REGISTRAR) parameter on the GRTDPRAUT command.

Tabella 54. Authorities granted with GRTDPRAUT AUT(\*REGISTRAR)

Library	Object	Type	Authorizations
QSYS	capctllib	*LIB	*USE, *ADD
capctllib <sup>1</sup>	QSQJRN	*JRN	*OBJOPR, *OBJMGT
capctllib <sup>1</sup>	QZS8CTLBLK	*USRSPC	*CHANGE
capctllib <sup>1</sup>	IBMSNAP_REGISTER	*FILE	*OBJOPR, *READ, *ADD, *UPDT, *DLT

Tabella 54. Authorities granted with *GRTDPRAUT AUT(\*REGISTRAR)* (Continua)

Library	Object	Type	Authorizations
capctllib <sup>1</sup>	IBMSNAP_REGISTERX	*FILE	*OBJOPR, *READ, *ADD, *UPDT, *DLT
capctllib <sup>1</sup>	IBMSNAP_REGISTERX1	*FILE	*OBJOPR, *READ, *ADD, *UPDT, *DLT
capctllib <sup>1</sup>	IBMSNAP_REGISTERX2	*FILE	*OBJOPR, *READ, *ADD, *UPDT, *DLT
capctllib <sup>1</sup>	IBMSNAP_REG_EXT	*FILE	*OBJOPR, *READ, *ADD, *UPDT, *DLT
capctllib <sup>1</sup>	IBMSNAP_REG_EXTX	*FILE	*OBJOPR, *READ, *ADD, *UPDT, *DLT
capctllib <sup>1</sup>	IBMSNAP_PRUNCNTL	*FILE	*OBJOPR, *READ
capctllib <sup>1</sup>	IBMSNAP_PRUNCNTLX	*FILE	*OBJOPR, *READ
capctllib <sup>1</sup>	IBMSNAP_PRUNCNTLX1	*FILE	*OBJOPR, *READ
capctllib <sup>1</sup>	IBMSNAP_PRUNCNTLX2	*FILE	*OBJOPR, *READ
capctllib <sup>1</sup>	IBMSNAP_PRUNCNTLX3	*FILE	*OBJOPR, *READ
ASN	ASN4B*	*SQLPKG	*USE
ASN	ASN4C*	*SQLPKG	*USE

**Nota:**

1. The entry *capctllib* in the Library column refers to the value passed to the **CAPCTLLIB** parameter of the *GRTDPRAUT* command; this command updates authority to only one Capture control library at a time.

The following table lists the authorities granted when you specify the *AUT(\*SUBSCRIBER)* parameter on the *GRTDPRAUT* command.

Tabella 55. Authorities granted with *GRTDPRAUT AUT(\*SUBSCRIBER)*

Library	Object	Type	Authorizations
QSYS	ASN	*LIB	*OBJOPR, *READ, *ADD, *EXECUTE
QSYS	capctllib	*LIB	*OBJOPR, *READ, *ADD, *EXECUTE
ASN	IBMSNAP_SUBS_SET	*FILE	*CHANGE
ASN	IBMSNAP_SUBS_COLS	*FILE	*CHANGE
ASN	IBMSNAP_SUBS_EVENT	*FILE	*CHANGE
ASN	IBMSNAP_SUBS_STMTS	*FILE	*CHANGE
ASN	IBMSNAP_SUBS_MEMBR	*FILE	*CHANGE
capctllib <sup>1</sup>	IBMSNAP_REGISTER	*FILE	*OBJOPR, *READ, *UPD, *EXECUTE
capctllib <sup>1</sup>	IBMSNAP_REG_EXT	*FILE	*OBJOPR, *READ, *UPD, *EXECUTE

Tabella 55. Authorities granted with *GRTDPRAUT AUT(\*SUBSCRIBER)* (Continua)

Library	Object	Type	Authorizations
capctllib <sup>1</sup>	IBMSNAP_PRUNCNTL	*FILE	*OBJOPR, *READ, *DLT, *ADD, *EXECUTE
capctllib <sup>1</sup>	IBMSNAP_PRUNCNTLX	*FILE	*USE
ASN	ASN4A*	*SQLPKG	*USE
ASN	ASN4U*	*SQLPKG	*USE

**Nota:**

1. The entry *capctllib* in the Library column refers to the value passed to the **CAPCTLLIB** parameter of the *GRTDPRAUT* command; this command updates authority to only one Capture control library at a time.

The following table lists the authorities granted when you specify the *AUT(\*CAPTURE)* parameter on the *GRTDPRAUT* command.

Tabella 56. Authorities granted with *GRTDPRAUT AUT(\*CAPTURE)*

Library	Object	Type	Authorizations
QSYS	capctllib	*LIB	*OBJOPR, *OBJMGT, *READ, *EXECUTE
QSYS	QDP4	*LIB	*OBJOPR, *ADD, *READ, *EXECUTE
capctllib <sup>1</sup>	QZSN	*MSGQ	*CHANGE
capctllib <sup>1</sup>	IBMSNAP_REGISTER	*FILE	*OBJOPR, *OBJMGT, *READ, *ADD, *UPD, *EXECUTE
capctllib <sup>1</sup>	IBMSNAP_REGISTERX	*FILE	*OBJOPR, *OBJMGT, *READ, *ADD, *UPD, *EXECUTE
capctllib <sup>1</sup>	IBMSNAP_REGISTERX1	*FILE	*OBJOPR, *OBJMGT, *READ, *ADD, *UPD, *EXECUTE
capctllib <sup>1</sup>	IBMSNAP_REGISTERX2	*FILE	*OBJOPR, *OBJMGT, *READ, *ADD, *UPD, *EXECUTE
capctllib <sup>1</sup>	IBMSNAP_REG_EXT	*FILE	*OBJOPR, *OBJMGT, *READ, *ADD, *UPD, *EXECUTE
capctllib <sup>1</sup>	IBMSNAP_REG_EXTX	*FILE	*OBJOPR, *OBJMGT, *READ, *ADD, *UPD, *EXECUTE
capctllib <sup>1</sup>	IBMSNAP_PRUNCNTL	*FILE	*OBJOPR, *OBJMGT, *READ, *UPD, *EXECUTE

Tabella 56. Authorities granted with GRTDPRAUT AUT(\*CAPTURE) (Continua)

Library	Object	Type	Authorizations
capctllib <sup>1</sup>	IBMSNAP_PRUNCNTLX	*FILE	*OBJOPR, *OBJMGT, *READ, *UPD, *EXECUTE
capctllib <sup>1</sup>	IBMSNAP_PRUNCNTLX1	*FILE	*OBJOPR, *OBJMGT, *READ, *UPD, *EXECUTE
capctllib <sup>1</sup>	IBMSNAP_PRUNCNTLX2	*FILE	*OBJOPR, *OBJMGT, *READ, *UPD, *EXECUTE
capctllib <sup>1</sup>	IBMSNAP_PRUNCNTLX3	*FILE	*OBJOPR, *OBJMGT, *READ, *UPD, *EXECUTE
capctllib <sup>1</sup>	IBMSNAP_CAPTRACE	*FILE	*CHANGE
capctllib <sup>1</sup>	IBMSNAP_CAPTRACEX	*FILE	*CHANGE
capctllib <sup>1</sup>	IBMSNAP_RESTART	*FILE	*CHANGE
capctllib <sup>1</sup>	IBMSNAP_RESTARTX	*FILE	*CHANGE
capctllib <sup>1</sup>	IBMSNAP_AUTHTKN	*FILE	*CHANGE
capctllib <sup>1</sup>	IBMSNAP_AUTHTKNX	*FILE	*CHANGE
capctllib <sup>1</sup>	IBMSNAP_UOW	*FILE	*OBJOPR, *OBJMGT, *READ, *UPD, *DLT, *ADD, *EXECUTE
capctllib <sup>1</sup>	IBMSNAP_UOW_IDX	*FILE	*CHANGE
capctllib <sup>1</sup>	IBMSNAP_PRUNE_SET	*FILE	*CHANGE
capctllib <sup>1</sup>	IBMSNAP_PRUNE_SETX	*FILE	*CHANGE
capctllib <sup>1</sup>	IBMSNAP_CAPPARMS	*FILE	*READ, *EXECUTE
capctllib <sup>1</sup>	IBMSNAP_SIGNAL	*FILE	*CHANGE
capctllib <sup>1</sup>	IBMSNAP_SIGNALX	*FILE	*CHANGE
capctllib <sup>1</sup>	IBMSNAP_CAPMON	*FILE	*CHANGE
capctllib <sup>1</sup>	IBMSNAP_CAPMONX	*FILE	*CHANGE
capctllib <sup>1</sup>	IBMSNAP_PRUNE_LOCK	*FILE	*CHANGE
ASN	ASN4B*	*SQLPKG	*USE
ASN	ASN4C*	*SQLPKG	*USE
ASN	QZS8CTLBLK	*USRSPC	*CHANGE

**Nota:**

1. The entry *capctllib* in the Library column refers to the value passed to the **CAPCTLLIB** parameter of the GRTDPRAUT command; this command updates authority to only one Capture control library at a time.

The following table lists the authorities granted when you specify the AUT(\*APPLY) parameter on the GRTDPRAUT command.

Tabella 57. Authorities granted with GRTDPRAUT AUT(\*APPLY)

Library	Object	Type	Authorizations
QSYS	ASN	*LIB	*OBJOPR, *READ, *EXECUTE
QSYS	capctllib	*LIB	*OBJOPR, *READ, *EXECUTE
QDP4	QZSNAPV2	*PGM	*OBJOPR, *READ, *OBMGT, *OBJALTER, *EXECUTE
capctllib <sup>1</sup>	IBMSNAP_REGISTER	*FILE	*OBJOPR, *READ, *UPD, *EXECUTE
capctllib <sup>1</sup>	IBMSNAP_REGISTERX	*FILE	*OBJOPR, *READ, *UPD, *EXECUTE
capctllib <sup>1</sup>	IBMSNAP_REGISTERX1	*FILE	*OBJOPR, *READ, *UPD, *EXECUTE
capctllib <sup>1</sup>	IBMSNAP_REGISTERX2	*FILE	*OBJOPR, *READ, *UPD, *EXECUTE
capctllib <sup>1</sup>	IBMSNAP_REGISTER_EXT	*FILE	*OBJOPR, *READ, *UPD, *EXECUTE
capctllib <sup>1</sup>	IBMSNAP_REGISTER_EXTX	*FILE	*OBJOPR, *READ, *UPD, *EXECUTE
capctllib <sup>1</sup>	IBMSNAP_SIGNAL	*FILE	*OBJOPR, *READ, *UPD, *ADD, *EXECUTE
capctllib <sup>1</sup>	IBMSNAP_SIGNALX	*FILE	*OBJOPR, *READ, *UPD, *ADD, *EXECUTE
capctllib <sup>1</sup>	IBMSNAP_PRUNE_LOCK	*FILE	*CHANGE
capctllib <sup>1</sup>	IBMSNAP_UOW	*FILE	*OBJOPR, *READ, *UPD, *ADD, *EXECUTE
capctllib <sup>1</sup>	IBMSNAP_PRUNCNTL	*FILE	*OBJOPR, *READ, *UPD, *ADD, *EXECUTE
capctllib <sup>1</sup>	IBMSNAP_AUTHTKN	*FILE	*OBJOPR, *READ, *UPD, *ADD, *EXECUTE
capctllib <sup>1</sup>	IBMSNAP_AUTHTKNX	*FILE	*OBJOPR, *READ, *UPD, *ADD, *EXECUTE
ASN	IBMSNAP_SUBS_SET	*FILE	*OBJOPR, *READ, *UPD, *EXECUTE
ASN	IBMSNAP_SUBS_SETX	*FILE	*OBJOPR, *READ, *UPD, *EXECUTE
ASN	IBMSNAP_APPLYTRAIL	*FILE	*OBJOPR, *READ, *UPD, *ADD, *EXECUTE
ASN	IBMSNAP_APPLYTRACE	*FILE	*OBJOPR, *READ, *UPD, *EXECUTE



Tabella 57. Authorities granted with GRTDPRAUT AUT(\*APPLY) (Continua)

Library	Object	Type	Authorizations
ASN	IBMSNAP_APPLYTRACX	*FILE	*OBJOPR, *READ, *UPD, *EXECUTE
ASN	IBMSNAP_SUBS_COLS	*FILE	*USE
ASN	IBMSNAP_SUBS_EVENT	*FILE	*USE
ASN	IBMSNAP_SUBS_STMTS	*FILE	*USE
ASN	IBMSNAP_SUBS_MEMBR	*FILE	*USE
ASN	ASN4A*	*SQLPKG	*USE
ASN	ASN4U*	*SQLPKG	*USE
ASN	IBMSNAP_APPLY_JOB	*FILE	*OBJOPR, *READ, *UPD, *ADD, *EXECUTE

**Nota:**

1. The entry *capctllib* in the Library column refers to the value passed to the **CAPCTLLIB** parameter of the GRTDPRAUT command; this command updates authority to only one Capture control library at a time.

## Examples for GRTDPRAUT

The following examples illustrate how to use the GRTDPRAUT command.

### Example 1:

To authorize a user named USER1 to define and modify registrations:

```
GRTDPRAUT CAPCTLLIB(ASN) USER(USER1) AUT(*REGISTRAR)
```

### Example 2:

To authorize a user named USER1 to define and modify subscription sets:

```
GRTDPRAUT CAPCTLLIB(ASN) USER(USER1) AUT(*SUBSCRIBER)
```

### Example 3:

To authorize a user named USER1 to run Capture programs:

```
GRTDPRAUT CAPCTLLIB(ASN) USER(USER1) AUT(*CAPTURE)
```

### Example 4:

To authorize a user named USER1 to define and modify existing subscription sets that are associated with Apply qualifier A1:

```
GRTDPRAUT CAPCTLLIB(ASN) USER(USER1) AUT(*SUBSCRIBER) APYQUAL(A1)
```

### Example 5:

To authorize a user to run the Apply program on the control server system for all subscription sets associated with Apply qualifier A1, where the target server is the same as the control server:

1. Run the following command on the system where the Apply program will run:  
GRTDPRAUT CAPCTLLIB(ASN) USER(USER1) AUT(\*APPLY) APYQUAL(A1)
2. Run the appropriate GRTDPRAUT command on the source server system:

- If the application server job on the source server used by the Apply program runs under user profile USER1, run the following command on the source server systems:  
GRTPRAUT CAPCTLLIB(ASN) USER(USER1) AUT(\*APPLY) APYQUAL(A1)
- If the application server job on the source server used by the Apply program runs under a different user profile, for example, QUSER, the command is:  
GRTPRAUT CAPCTLLIB(ASN) USER(QUSER) AUT(\*APPLY) APYQUAL(A1)

## INZDPRCAP: Reinitializing DPR Capture (System i)

### System i

Use the Initialize DPR Capture (INZDPRCAP) command to initialize the Capture program by directing the Capture program to work with an updated list of source tables.

Source tables under the control of a Capture program can change while the Capture program is running. Use the INZDPRCAP command to ensure that the Capture program processes the most up-to-date replication sources.

The Capture program must be running before you can run this command.

After you type the command name on the command line, you can press the F4 key to display the command syntax.

To display a complete description of this command and all of its parameters, move the cursor to the command at the top of the screen and press the F1 key. To display a description of a specific parameter, place the cursor on that parameter and press the F1 key.

### Syntax

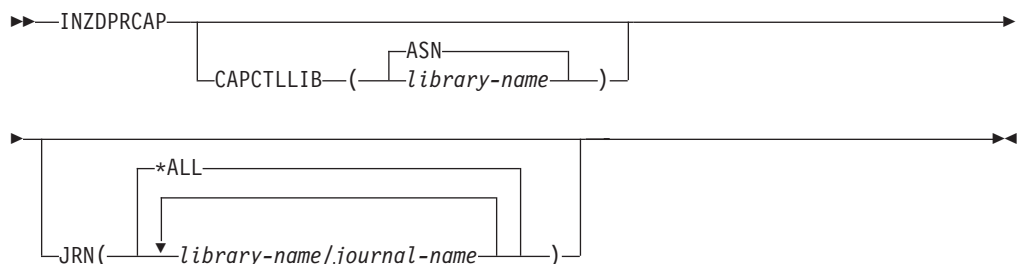


Tabella 58 lists the invocation parameters.

Tabella 58. INZDPRCAP command parameter definitions for System i

Parameter	Definition and prompts
CAPCTLLIB	Specifies the Capture schema, which is the name of the library in which the Capture control tables reside. <b>ASN (default)</b> The Capture control tables reside in the ASN library. The ASN library is the default library. <i>library-name</i> The name of a library that contains the Capture control tables.

Tabella 58. INZDPRCAP command parameter definitions for System i (Continua)

Parameter	Definition and prompts
JRN	<p>Specifies a subset of up to 50 journals that you want the Capture program to work with. The Capture program starts processing all the source tables that are currently journaled to this journal.</p> <p><b>*ALL (default)</b> The Capture program works with all the journals.</p> <p><i>library-name/journal-name</i> The qualified name of the journal that you want the Capture program to work with.</p>

## Examples for INZDPRCAP

The following examples illustrate how to use the INZDPRCAP command.

### Example 1:

To initialize a Capture program using the QSQJRN journal under a library named TRAINING:

```
INZDPRCAP CAPCTLLIB(ASN) JRN(TRAINING/QSQJRN)
```

The Capture control tables reside in the default ASN schema.

### Example 2:

To initialize a Capture program that works with all the journals:

```
INZDPRCAP CAPCTLLIB(BSN) JRN(*ALL)
```

The Capture control tables reside in a schema called BSN.

## OVRDPRCAPA: Overriding DPR Capture attributes (System i)

### System i

Use the Override DPR Capture attributes (OVRDPRCAPA) command to alter the behavior of a running Capture program.

This command alters the program behavior by overriding the values that were passed to the Capture program from the IBMSNAP\_CAPPARMS table or from the STRDPRCAP command when the Capture program started.

After you type the command name on the command line, you can press the F4 key to display the command syntax.

To display a complete description of this command and all of its parameters, move the cursor to the command at the top of the screen and press the F1 key. To display a description of a specific parameter, place the cursor on that parameter and press the F1 key.

### Syntax

```
►► OVRDPRCAPA CAPCTLLIB ( ( ASN library-name ) )
```

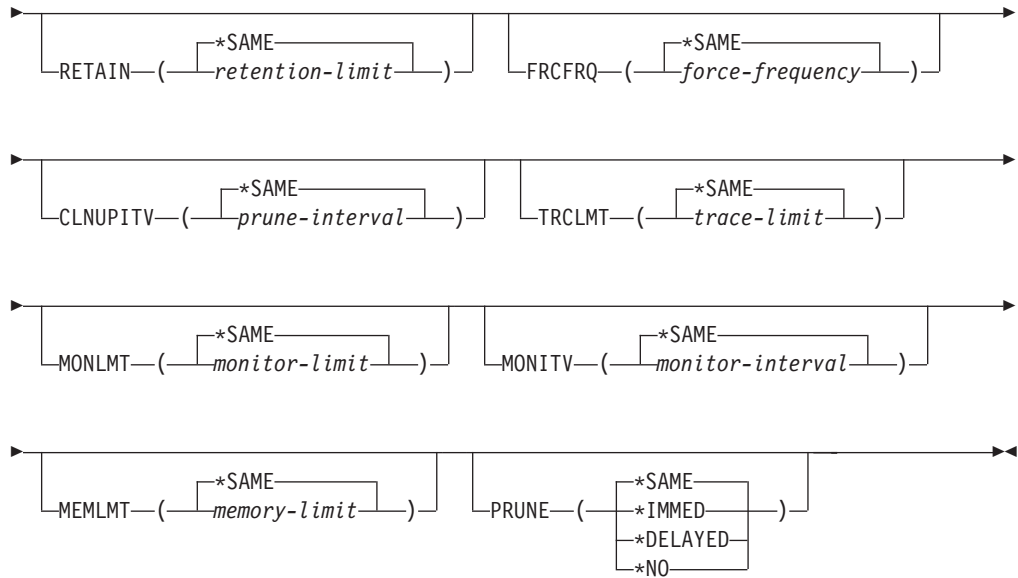


Tabella 59 lists the invocation parameters.

Tabella 59. OVRDPRCAPA command parameter definitions for System i

Parameter	Definition and prompts
<b>CAPCTLLIB</b>	<p>Specifies the Capture schema, which is the name of the library in which the Capture control tables reside. This library includes the IBMSNAP_REGISTER table, which stores the registration information of the source tables. This parameter is required.</p> <p><b>ASN (default)</b> The Capture control tables reside in the ASN library.</p> <p><i>library-name</i> The name of a library that contains the Capture control tables. You can create this library by using the CRTDPRTBL command with the <b>CAPCTLLIB</b> parameter.</p>

Tabella 59. OVRDPRCAPA command parameter definitions for System i (Continua)

Parameter	Definition and prompts
<b>RETAIN</b>	<p>Specifies the number of minutes that data is retained in the change-data (CD), UOW, IBMSNAP_SIGNAL, and IBMSNAP_AUTHTKN tables before the data is removed.</p> <p>This value works with the <b>CLNUPITV</b> parameter value from the Start DPR Capture (STRDPRCAP) command. First, the Capture program deletes any CD, UOW, IBMSNAP_SIGNAL, or IBMSNAP_AUTHTKN rows that are older than the oldest currently running Apply program. Then, a new or remaining row from the CD, UOW, IBMSNAP_SIGNAL, or IBMSNAP_AUTHTKN table is subsequently deleted when its age reaches the value of the <b>RETAIN</b> parameter.</p> <p>Ensure that the Apply intervals are set to copy changed information before the data reaches this <b>RETAIN</b> parameter value to prevent inconsistent data in your tables. If the data becomes inconsistent, the Apply program performs a full refresh.</p> <p>The default is 10 080 minutes (seven days). The maximum is 35000000 minutes.</p> <p><b>*SAME (default)</b> This value is not changed.</p> <p><i>retention-limit</i> The new retention limit value.</p>
<b>FRCFRQ</b>	<p>Specifies how often (from 30 to 600 seconds) the Capture program writes changes to the change-data (CD) and unit-of-work (UOW) tables.</p> <p>The Capture program makes these changes available to the Apply program either when the buffers are filled or when the <b>FRCFRQ</b> time limit expires, whichever is sooner. This parameter value affects the amount of time that it takes for the Capture program to respond to changes from the Initialize DPR Capture (INZDPRCAP) command.</p> <p>Use this parameter to make changes more readily available to the Apply program on servers with a low rate of source table changes. The <b>FRCFRQ</b> parameter value is a global value used for all registered source tables. Setting the <b>FRCFRQ</b> value to a low number can affect system performance.</p> <p>The default is 30 seconds.</p> <p><b>*SAME (default)</b> This value is not changed.</p> <p><i>force-frequency</i> The new number of seconds that the Capture program keeps CD and UOW table changes in buffer space before making these changes available to the Apply program.</p>

Tabella 59. OVRDPRCAPA command parameter definitions for System i (Continua)

Parameter	Definition and prompts
CLNUPITV	<p>Specifies the maximum amount of time (in hours) before the Capture program prunes old records from the change-data (CD), unit-of-work (UOW), IBMSNAP_SIGNAL, IBMSNAP_CAPMON, IBMSNAP_CAPTRACE, and IBMSNAP_AUTHTKN tables.</p> <p>This parameter works with the <b>RETAIN</b> parameter to control pruning of the CD, UOW, IBMSNAP_SIGNAL, and IBMSNAP_AUTHTKN tables, with the <b>MONLMT</b> parameter to control pruning of the IBMSNAP_CAPMON table, and with the <b>TRCLMT</b> parameter to control pruning of the IBMSNAP_CAPTRACE table.</p> <p>(Use the STRDPRCAP command to set the <b>RETAIN</b>, <b>MONLMT</b>, and <b>TRCLMT</b> parameters for a Capture program.)</p> <p>The value of the <b>CLNUPITV</b> parameter is automatically converted from hours to seconds and is stored in the PRUNE_INTERVAL column of the IBMSNAP_CAPPARMS table.</p> <p><b>*SAME (default)</b> This Capture attribute value is not changed.</p> <p><i>prune-interval</i> The pruning interval expressed as a specific number of hours (1 to 100).</p>
TRCLMT	<p>Specifies the trace limit, which indicates how frequently the IBMSNAP_CAPTRACE table is pruned.</p> <p><b>*SAME (default)</b> The Capture program continues and uses the current trace limit value.</p> <p><i>trace-limit</i> The number of minutes between each pruning operation of the IBMSNAP_CAPTRACE table.</p>
MONLMT	<p>Specifies the monitor limit, which indicates how frequently the IBMSNAP_CAPMON table is pruned.</p> <p><b>*SAME (default)</b> The Capture program continues and uses the current monitor limit value.</p> <p><i>monitor-limit</i> The number of minutes between each pruning operation of the IBMSNAP_CAPMON table.</p>
MONITV	<p>Specifies the monitor interval (in seconds), which indicates how frequently the Capture program inserts rows into the IBMSNAP_CAPMON table.</p> <p><b>*SAME (default)</b> The Capture program continues and uses the current monitor interval value.</p> <p><i>monitor-interval</i> The number of seconds between row insertion into the IBMSNAP_CAPMON table. The monitor interval must be at least 120 seconds (two minutes). If you type a number that is less than 120, the command automatically sets this parameter value to 120.</p>

Tabella 59. OVRDPRCAPA command parameter definitions for System i (Continua)

Parameter	Definition and prompts
MEMLMT	<p>Specifies the maximum size (in megabytes) of memory that the Capture journal job can use.</p> <p><b>*SAME (default)</b> The Capture program continues and uses the current memory limit value.</p> <p><i>memory-limit</i> The maximum number of megabytes for memory.</p>
PRUNE	<p>Use this parameter to change the way that the Capture program prunes rows from the change-data (CD), unit-of-work (UOW), IBMSNAP_SIGNAL, IBMSNAP_CAPMON, IBMSNAP_CAPTRACE, and IBMSNAP_AUTHTKN tables.</p> <p><b>*SAME (default)</b> The Capture program continues and uses the pruning parameters that you specified when you started the STRDPRCAP command.</p> <p><b>*IMMED</b> The Capture program starts pruning the tables immediately, regardless of the value of the CLNUPITV parameter that you specified when you started the STRDPRCAP command.</p> <p><b>*DELAYED</b> The Capture program prunes the old rows at the end of the specified pruning interval.</p> <p>PRUNE(*DELAYED) does not affect the frequency of pruning if you set the second part of the CLNUPITV parameter to *IMMED or *DELAYED on the STRDPRCAP command. However, PRUNE(*DELAYED) <i>does</i> initiate pruning if you set the second part of the CLNUPITV parameter to *NO when you started the STRDPRCAP command.</p> <p><b>*NO</b> The Capture program does not initiate pruning. This value overrides the CLNUPITV parameter setting from the STRDPRCAP command.</p>

## Examples for OVRDPRCAPA

The following examples illustrate how to use the OVRDPRCAPA command.

### Example 1:

To change the pruning parameters of the CD, UOW, IBMSNAP\_SIGNAL, IBMSNAP\_CAPMON, IBMSNAP\_CAPTRACE, and IBMSNAP\_AUTHTKN tables (which reside under the default ASN library) and to change the IBMSNAP\_CAPMON monitor interval and memory limit of Capture journal jobs in a running Capture program:

```
OVRDPRCAPA CAPCTLLIB(ASN) CLNUPITV(12) MONITV(600) MEMLMT(64)
```

### Example 2:

To initiate pruning of the CD, UOW, IBMSNAP\_SIGNAL, IBMSNAP\_CAPMON, IBMSNAP\_CAPTRACE, and IBMSNAP\_AUTHTKN tables, which reside in the BSN library:

```
OVRDPRCAPA CAPCTLLIB(BSN) PRUNE(*IMMED)
```

## RMVDPRREG: Removing a DPR registration (System i)

### System i

Use the Remove DPR registration (RMVDPRREG) command to remove a single source table from the IBMSNAP\_REGISTER table so that the source table is no longer used for replication.

After you type the command name on the command line, you can press the F4 key to display the command syntax.

To display a complete description of this command and all of its parameters, move the cursor to the command at the top of the screen and press the F1 key. To display a description of a specific parameter, place the cursor on that parameter and press the F1 key.

### Syntax

```
►► RMVDPRREG—SRCTBL(—library-name/file-name—)—————►  
  
►—————►  
┌──────────┐  
└──CAPCTLLIB──(──┐ASN──┐  
                  └──library-name──┘)──┘
```

Tabella 60 lists the invocation parameters.

Tabella 60. RMVDPRREG command parameter definitions for System i

Parameter	Definition and prompts
SRCTBL	Identifies the registration that you want to remove. This is a required parameter.  <i>library-name/file-name</i> The qualified name of the registered file.
CAPCTLLIB	Specifies the Capture schema, which is the name of the library in which the Capture control tables reside.  <b>ASN (default)</b> The Capture control tables are in the ASN library.  <i>library-name</i> The name of a library containing the Capture control tables.

### Examples for RMVDPRREG

The following examples illustrate how to use the RMVDPRREG command.

#### Example 1:

To remove the registration for the source table named EMPLOYEE of the HR library in the default ASN Capture schema:

```
RMVDPRREG SRCTBL(HR/EMPLOYEE)
```

#### Example 2:





Tabella 61. *RMVDPRSUB* command parameter definitions for System *i* (Continua)

Parameter	Definition and prompts
CTLSVR	<p>Specifies the relational database name of the system that contains the Apply control tables.</p> <p><b>*LOCAL (default)</b> The Apply control tables reside locally (on the machine from which you are running the <i>RMVDPRSUB</i> command).</p> <p><i>rdb-name</i> The name of the relational database where the Apply control tables reside. You can use the Work with RDB Directory Entries (<i>WRKRDBDIRE</i>) command to find this name.</p>
RMVREG	<p>Specifies whether this command removes the registrations that are associated with the target tables of all the subscription-set members in the subscription set. Use this parameter only if you have set the <b>RMVMBRS</b> parameter to *YES.</p> <p><b>*NO (default)</b> The registrations are not removed.</p> <p><b>*YES</b> The registrations are removed.</p>
DLTTGTTBL	<p>Specifies whether this command drops the target tables of the subscription-set members after the subscription set is removed. Use this parameter only if you set the <b>RMVMBRS</b> parameter to *YES.</p> <p><b>*NO (default)</b> The target tables are not dropped.</p> <p><b>*YES</b> The target tables are dropped.</p>
RMVMBRS	<p>Specifies whether this command removes the subscription set and all the members in that subscription set.</p> <p><b>*NO (default)</b> The subscription set is not removed if there are existing members in the subscription set.</p> <p><b>*YES</b> The subscription set and all its subscription-set members are removed.</p>

## Examples for *RMVDPRSUB*

The following examples illustrate how to use the **RMVDPRSUB** command.

### Example 1:

To remove a subscription set named **SETHR** that contains no subscription-set members:

```
RMVDPRSUB APYQUAL(AQHR) SETNAME(SETHR)
```

### Example 2:

To remove a subscription set named **SETHR** and all its subscription-set members:

```
RMVDPRSUB APYQUAL(AQHR) SETNAME(SETHR) RMVMBRS(*YES)
```

### Example 3:



Tabella 62. RMVDPRSUBM command parameter definitions for System i (Continua)

Parameter	Definition and prompts
CTLSVR	<p>Specifies the relational database name of the system that contains the Apply control tables.</p> <p><b>*LOCAL (default)</b> The Apply control tables reside locally (on the machine from which you are running the RMVDPRSUBM command).</p> <p><i>rdb-name</i> The name of the relational database where the Apply control tables reside. You can use the Work with RDB Directory Entries (WRKRDBDIRE) command to find this name.</p>
RMVREG	<p>Specifies whether this command removes the registration that is associated with the target table for the subscription-set member.</p> <p><b>*NO (default)</b> The registration is not removed.</p> <p><b>*YES</b> The registration is removed.</p>
DLTTGTTBL	<p>Specifies whether this command drops the target table of the subscription-set member after the subscription-set member is removed.</p> <p><b>*NO (default)</b> The target table is not dropped.</p> <p><b>*YES</b> The target table is dropped.</p>

## Examples for RMVDPRSUBM

The following examples illustrate how to use the RMVDPRSUBM command.

### Example 1:

To remove a subscription-set member, which uses a target table named EMP, from the SETEMP subscription set on the relational database named RMTRDB1:

```
RMVDPRSUBM APYQUAL(AQHR) SETNAME(SETEMP) TGTTBL(TGTEMP/EMP) CTLSVR(RMTRDB1)
```

### Example 2:

To remove a subscription-set member from the SETHR subscription set, remove the registration, and then drop the table:

```
RMVDPRSUBM APYQUAL(AQHR) SETNAME(SETHR) TGTTBL(TGTHR/YTD TAX) RMVREG(*YES)
DLTTGTTBL(*YES)
```

## RVKDPRAUT: Revoking authority (System i)

### System i

The Revoke DPR Authority (RVKDPRAUT) command revokes authority to the replication control tables so that users can no longer define or modify replication sources and subscription sets.

After you type the command name on the command line, you can press the F4 key to display the command syntax.

To display a complete description of this command and all of its parameters, move the cursor to the command at the top of the screen and press the F1 key. To display a description of a specific parameter, place the cursor on that parameter and press the F1 key.

## Syntax

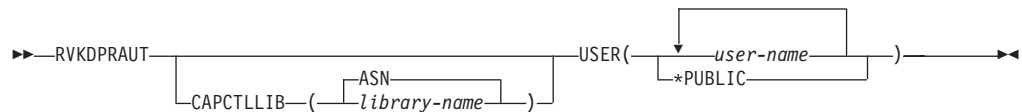


Tabella 63 lists the invocation parameters.

Tabella 63. RVKDPRAUT command parameter definitions for System i

Parameter	Definition and prompts
<b>CAPCTLLIB</b>	Specifies the Capture schema, which is the name of the library under which user authority is being revoked.  <b>ASN (default)</b> The Capture control tables reside in the ASN library.  <i>library-name</i> The name of the library that contains the replication control tables.
<b>USER</b>	Specifies the users whose authority is revoked. This parameter is required.  <i>user-name</i> Specifies the names of up to 50 users whose authority is revoked.  <b>*PUBLIC</b> Specifies that authority is revoked from all users without specific authority, who are not on the authorization list, and whose group profile does not have any authority.

## Usage notes

The command returns an error message if any of the following conditions exist:

- A specified user does not exist.
- The user running the command is not authorized to the specified user profiles.
- The user running the command does not have permission to revoke authorities to the DB2 DataPropagator for System i control tables.
- The DB2 DataPropagator for System i control tables do not exist.
- The Capture or Apply programs are running.

## Examples for RVKDPRAUT

The following examples illustrate how to use the RVKDPRAUT command.

### Example 1:

To revoke the authority from a user named HJONES to the control tables under the ASN library:

```
RVKDPRAUT CAPCTLLIB(ASN) USER(HJONES)
```

## Example 2:

To revoke the authority from all users that were not specified in the GRTPRAUT command so that they cannot access the control tables in the ASN library:

```
RVKDPRAUT CAPCTLLIB(ASN) USER(*PUBLIC)
```

---

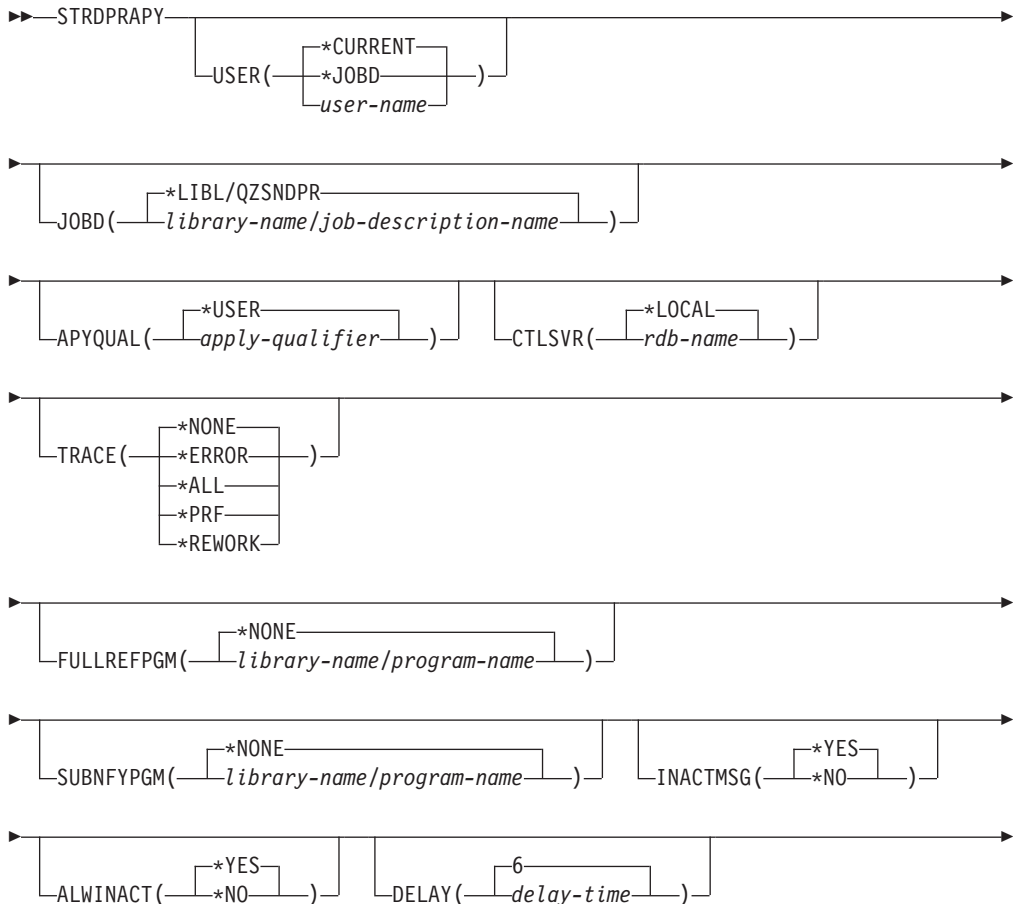
## STRDPRAPY: Starting Apply (System i)

### System i

Use the Start DPR Apply (STRDPRAPY) command to start an Apply program on your local system. The Apply program continues to run until you stop it or until it detects an unrecoverable error.

After you type the command name on the command line, you can press the F4 key to display the command syntax.

To display a complete description of this command and all of its parameters, move the cursor to the command at the top of the screen and press the F1 key. To display a description of a specific parameter, place the cursor on that parameter and press the F1 key.



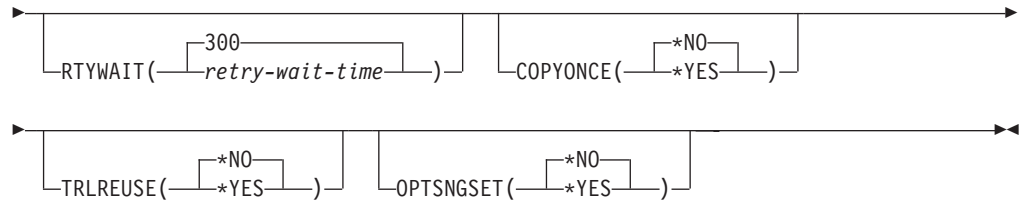


Tabella 64 lists the invocation parameters.

Tabella 64. STRDPRAPY command parameter definitions for System i

Parameter	Definition and prompts
<b>USER</b>	<p>Specifies the name of the user ID for which to start the Apply program. When you run this command, you must be authorized (have *USE rights) to the specified user profile; the Apply program runs under this specified user profile.</p> <p>The control tables are located on the relational database specified by the <b>CTLSVR</b> parameter. The same control tables are used regardless of the value specified on the <b>USER</b> parameter.</p> <p><b>*CURRENT (default)</b> The user ID associated with the current job is the same user ID associated with this Apply program.</p> <p><b>*JOBID</b> The user ID specified in the job description associated with this Apply program. The job description cannot specify USER(*RQD).</p> <p><i>user-name</i> The user ID associated with this Apply program. The following IBM-supplied objects are <i>not</i> valid on this parameter: QDBSHR, QDFTOWN, QDOC, QLPAUTO, QLPINSTALL, QRJE, QSECOFR, QSPL, QSYS, or QTSTRQS.</p> <p>When prompting on the STRDPRAPY command, you can press the F4 key to see a list of users who defined subscription sets.</p>
<b>JOBID</b>	<p>Specifies the name of the job description to use when submitting the Apply program.</p> <p><b>*LIBL/QZSNDPR (default)</b> The default job description provided with DB2 DataPropagator for System i.</p> <p><i>library-name/job-description-name</i> The name of the job description used for the Apply program.</p>

Tabella 64. STRDPRAPY command parameter definitions for System i (Continua)

Parameter	Definition and prompts
APYQUAL	<p>Specifies the Apply qualifier to be used by the Apply program. All subscriptions sets that are grouped together with this Apply qualifier are run by the Apply program.</p> <p><b>*USER (default)</b> The <b>USER</b> parameter value that you enter is used as the name of the Apply qualifier.</p> <p><i>apply-qualifier</i> The name used to group the subscription sets that are to be run by this Apply program. You can specify a maximum of 18 characters for the Apply qualifier name. This name follows the same naming conventions as a relational database name.</p> <p>When prompting on the STRDPRAPY command, you can press the F4 key to see a list of Apply qualifier names with existing subscription sets.</p>
CTLSVR	<p>Specifies the relational database name of the system that contains the Apply control tables.</p> <p><b>*LOCAL (default)</b> The Apply control tables reside locally (on the machine where you are running the STRDPRAPY command).</p> <p><i>rdb-name</i> The name of the relational database where the Apply control tables reside. You can use the Work with RDB Directory Entries (WRKRDBDIRE) command to find this name.</p> <p>When prompting on the STRDPRAPY command, you can press the F4 key to see a list of available RDB names.</p>
TRACE	<p>Specifies whether the Apply program should generate a trace. The Apply program writes the trace data to a spool file called QPZSNATRC.</p> <p><b>*NONE (default)</b> No trace is generated.</p> <p><b>*ERROR</b> The trace contains error information only.</p> <p><b>*ALL</b> The trace contains error and execution flow information.</p> <p><b>*PRF</b> The trace contains information that can be used to analyze performance at different stages of the Apply program execution.</p> <p><b>*REWORK</b> The trace contains information about rows that were reworked by the Apply program.</p>



Tabella 64. STRDPRAPY command parameter definitions for System i (Continua)

Parameter	Definition and prompts
<b>FULLREFPGM</b>	<p>Specifies whether the Apply program is to invoke an exit routine to initialize a target table. When the Apply program is ready to perform a full refresh of a target table, the Apply program invokes the specified exit routine rather than performing the full refresh itself.</p> <p>When a full-refresh exit routine is used by the Apply program, the value of the ASNLOAD column in the IBMSNAP_APPLYTRAIL table is Y.</p> <p><b>*NONE (default)</b> A full-refresh exit routine is not used.</p> <p><i>library-name/program-name</i> The qualified name of the program that is called by the Apply program performing a full refresh of a target table. For example, to call program ASNLOAD in library DATAPROP, the qualified name is DATAPROP/ASNLOAD.</p>
<b>SUBNFYPGM</b>	<p>Specifies whether the Apply program is to invoke an exit routine when the program finishes processing a subscription set. Input to the exit routine includes the subscription set name, Apply qualifier, completion status, and statistics with the number of rejects.</p> <p>The notify program allows you to examine the unit-of-work (UOW) table to determine when transactions have been rejected and when to take further actions, such as issuing a message or generating an event.</p> <p><b>*NONE (default)</b> An exit routine is not used.</p> <p><i>library-name/program-name</i> The qualified name of the exit routine program called by the Apply program when processing a subscription set. For example, to call program APPLYDONE in library DATAPROP, the qualified name is DATAPROP/APPLYDONE.</p>
<b>INACTMSG</b>	<p>Specifies whether the Apply program is to generate a message whenever the program completes its work and becomes inactive for a period of time.</p> <p><b>*YES (default)</b> The Apply program generates message ASN1044 before beginning a period of inactivity. Message ASN1044 indicates how long the Apply program remains inactive.</p> <p><b>*NO</b> No message is generated.</p>
<b>ALWINACT</b>	<p>Specifies whether the Apply program can run in an inactive (sleep) state.</p> <p><b>*YES (default)</b> The Apply program sleeps if there is nothing to process.</p> <p><b>*NO</b> If the Apply program has nothing to process, the job that submitted and started the Apply program ends.</p>

Tabella 64. STRDPRAPY command parameter definitions for System i (Continua)

Parameter	Definition and prompts
DELAY	<p>Specifies the delay time (in seconds) at the end of each Apply program cycle when continuous replication is used.</p> <p><b>6 (default)</b> The delay time is six seconds.</p> <p><i>delay-time</i> The delay time, entered as a number between 0 and 6 inclusive.</p>
RTYWAIT	<p>Specifies how long (in seconds) the Apply program is to wait after it encounters an error before it retries the operation that failed.</p> <p><b>300 (default)</b> The retry wait time is 300 seconds (five minutes).</p> <p><i>retry-wait-time</i> The wait time, entered as a number between 0 and 35000000 inclusive, before the Apply program retries the failed operation.</p>
COPYONCE	<p>Specifies whether the Apply program executes one copy cycle for each subscription set that is eligible at the time the Apply program is invoked. Then the Apply program terminates. An eligible subscription set meets the following criteria:</p> <ul style="list-style-type: none"> <li>• (ACTIVATE &gt; 0) in the IBMSNAP_SUBS_SET table. When the ACTIVATE column value is greater than zero, the subscription set is active indefinitely or is used for a one-time-only subscription processing.</li> <li>• (REFRESH_TYPE = R or B) or (REFRESH_TYPE = E and the specified event occurred). The REFRESH_TYPE column value is stored in the IBMSNAP_SUBS_SET table.</li> </ul> <p>The MAX_SYNCH_MINUTES limit from the IBMSNAP_SUBS_SET table and the END_OF_PERIOD timestamp from the IBMSNAP_SUBS_EVENT table are honored if specified.</p> <p><b>*NO (default)</b> The Apply program does not execute one copy cycle for each eligible subscription set.</p> <p><b>*YES</b> The Apply program executes one copy cycle for each eligible subscription set and then terminates.</p>
TRLREUSE	<p>Specifies whether the Apply program empties the IBMSNAP_APPLYTRAIL table when the Apply program starts.</p> <p><b>*NO (default)</b> The Apply program does not empty the IBMSNAP_APPLYTRAIL table during program startup.</p> <p><b>*YES</b> The Apply program empties the IBMSNAP_APPLYTRAIL table during program startup.</p>

Tabella 64. STRDPRAPY command parameter definitions for System i (Continua)

Parameter	Definition and prompts
OPTSNGSET	<p>Specifies whether the performance of the Apply program is optimized if only one subscription set is processed. This parameter does not pertain to replica target tables.</p> <p>If you set this parameter to *YES, the Apply program fetches the members and columns of a subscription set only once and reuses this fetched information when processing the same subscription set in two or more consecutive processing cycles.</p> <p><b>*NO (default)</b> The performance of the Apply program is not optimized if only one subscription set is processed.</p> <p><b>*YES</b> The performance of the Apply program is optimized if only one subscription set is processed. The Apply program reuses the subscription set information during subsequent processing cycles, requiring fewer CPU resources and improving throughput rates.</p>

## Usage notes

You can set up the system to start the subsystem automatically by adding the command that is referred to in the QSTRUPPGM value on your system. If you use the QDP4/QZSNDPR subsystem, it is started as part of the STRDPRAPY command processing.

If the relational database (RDB) specified by the **CTLSVR** parameter is a DB2 for i5/OS database, the tables on the server are found in the ASN library. If the RDB is not a DB2 for i5/OS database, you can access the tables by using ASN as the qualifier.

## Error conditions when starting the Apply program

The STRDPRAPY command issues an error message if any of the following conditions occur:

- If the user does not exist.
- If the user running the command is not authorized to the user profile specified on the command or the job description.
- If an instance of the Apply program is already active on the local system for this combination of Apply qualifier and control server.
- If the RDB name specified by the **CTLSVR** parameter is not in the relational database directory.
- If the control tables do not exist on the RDB specified by the **CTLSVR** parameter.
- If no subscription sets are defined for the Apply qualifier specified by the **APYQUAL** parameter.

An Apply program must be started for each unique Apply qualifier in every IBMSNAP\_SUBS\_SET table. You can start multiple Apply programs by specifying a different Apply qualifier each time that you issue the STRDPRAPY command. These Apply programs will run under the same user profile.

## Identifying Apply program jobs

Each Apply program is identified by using both the Apply qualifier and the control server names. When run, the job started for the Apply program does not have sufficient external attributes to correctly identify which Apply program is associated with a particular Apply qualifier and control server combination. Therefore, the job is identified in the following way:

- The job is started under the user profile associated with the **USER** parameter.
- The first 10 characters of the Apply qualifier are truncated and become the job name.
- DB2 DataPropagator for System i maintains an IBMSNAP\_APPLY\_JOB table named in the ASN library on the local system. The table maps the Apply qualifier and control server values to the correct Apply program job.
- You can view the job log. The Apply qualifier and control server names are used in the call to the Apply program.

In general, you can identify the correct Apply program job by looking at the list of jobs running in the QZSNDPR subsystem if both:

- The first 10 characters of the Apply qualifier name are unique.
- The Apply program is started for the local control server only.

## Examples for STRDPRAPY

The following examples illustrate how to use the STRDPRAPY command.

### Example 1:

To start the Apply program that uses the AQHR Apply qualifier and Apply control tables that reside locally and to generate a trace file that contains error and execution flow information:

```
STRDPRAPY APYQUAL(AQHR) CTLSVR(*LOCAL) TRACE(*ALL)
```

### Example 2:

To start an Apply program with Apply control tables that reside locally and to specify that the job that started this Apply program automatically end when the Apply program has nothing left to process:

```
STRDPRAPY APYQUAL(AQHR) CTLSVR(*LOCAL) ALWINACT(*NO)
```

### Example 3:

To start an Apply program that empties the IBMSNAP\_APPLYTRAIL table during program startup:

```
STRDPRAPY APYQUAL(AQHR) CTLSVR(*LOCAL) TRLREUSE(*YES)
```

### Example 4:

To start an Apply program with all default values:

```
STRDPRAPY
```

---

## STRDPRCAP: Starting Capture (System i)

System i

Use the Start DPR Capture (STRDPRCAP) command to start capturing changes to System i database tables on System i servers.

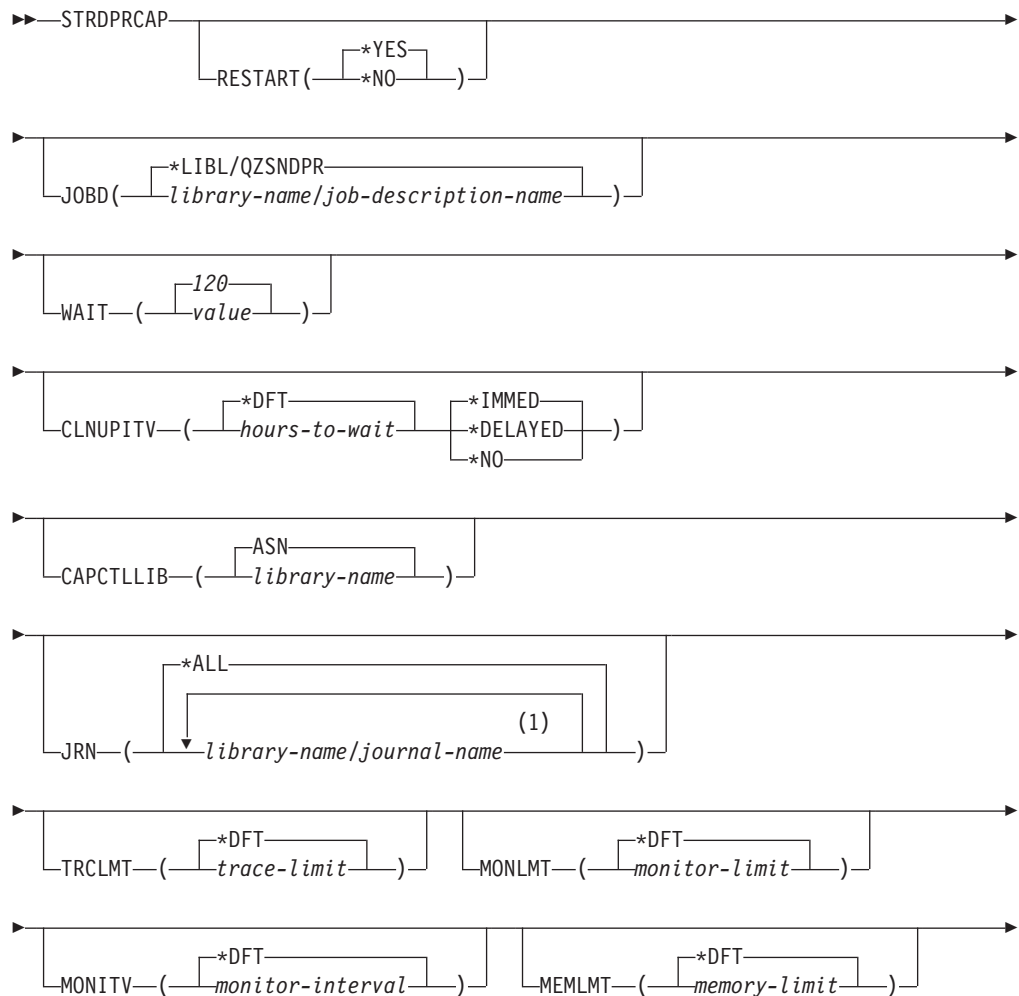
Because this command processes all replication sources in the IBMSNAP\_REGISTER table, make sure that you are running this command with the proper authority.

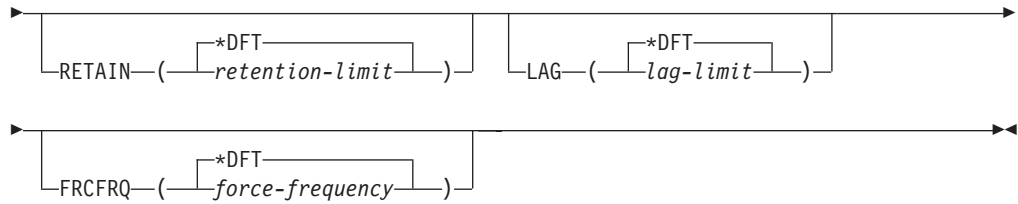
After you start the Capture program, it runs continuously until you stop it or it detects an unrecoverable error.

After you type the command name on the command line, you can press the F4 key to display the command syntax.

To display a complete description of this command and all of its parameters, move the cursor to the command at the top of the screen and press the F1 key. To display a description of a specific parameter, place the cursor on that parameter and press the F1 key.

## Syntax





**Note:**

- 1 You can specify up to 50 journals.

Tabella 65 lists the invocation parameters.

*Tabella 65. STRDPRCAP command parameter definitions for System i*

Parameter	Definition and prompts
RESTART	<p>Specifies how the Capture program handles warm and cold starts.</p> <p><b>*YES (default)</b> The Capture program continues processing the changes from the point where it was when it ended previously. This is also known as a <i>warm start</i> and is the normal mode of operation.</p> <p><b>*NO</b> The Capture program removes all information from the change-data (CD) tables. The Capture program also removes all information from the unit-of-work (UOW) table when you specify JRN(*ALL). All subscriptions for affected source tables are fully refreshed before change capturing resumes. This process is also known as a <i>cold start</i>. By specifying RESTART(*NO) and JRN(<i>library-name/journal-name</i>), you can cold start the Capture program for specified journals.</p>
JOBID	<p>Specifies the name of the job description to use when submitting the Capture program.</p> <p><b>*LIBL/QZSNDPR (default)</b> Specifies the default job description provided with DB2 DataPropagator for System i.</p> <p><i>library-name/job-description-name</i> The name of the job description used for the Capture program.</p>

Tabella 65. STRDPRCAP command parameter definitions for System i (Continua)

Parameter	Definition and prompts
WAIT	<p>Specifies the maximum number of seconds (60 to 6 000) to wait before the Capture program checks its status. You can use this value to tune the responsiveness of the Capture program.</p> <p>A low value reduces the time that the Capture program takes before ending or initializing, but can have a negative effect on system performance. A higher value increases the time that the Capture program takes before ending or initializing, but can improve system performance. A value that is too high can result in decreased responsiveness while the Capture program is performing periodic processing. The amount of the decrease in responsiveness depends on the amount of change activity to source tables and the amount of other work occurring on the system.</p> <p><b>120 (default)</b> The Capture program waits 120 seconds.</p> <p><i>value</i> The maximum number of seconds that the Capture program waits.</p>
CLNUPITV	<p>Specifies the maximum amount of time (in hours) before the Capture program prunes old records from the change-data (CD), unit-of-work (UOW), IBMSNAP_SIGNAL, IBMSNAP_CAPMON, IBMSNAP_CAPTRACE, and IBMSNAP_AUTHTKN tables.</p> <p>This parameter works with the <b>RETAIN</b> parameter to control pruning of the CD, UOW, IBMSNAP_SIGNAL, and IBMSNAP_AUTHTKN tables, with the <b>MONLMT</b> parameter to control pruning of the IBMSNAP_CAPMON table, and with the <b>TRCLMT</b> parameter to control pruning of the IBMSNAP_CAPTRACE table.</p> <p>(Use the <b>STRDPRCAP</b> command to set the <b>RETAIN</b>, <b>MONLMT</b>, and <b>TRCLMT</b> parameters for a Capture program. Use the <b>CHGDPRCAPA</b> or <b>OVRDPRCAPA</b> command to change these parameter settings.)</p> <p>There are two parts to the <b>CLNUPITV</b> parameter:</p> <p><b>*DFT (default)</b> The Capture program uses the value of the PRUNE_INTERVAL column from the IBMSNAP_CAPPARMS table.</p> <p><i>hours-to-wait</i> The pruning interval expressed as a specific number of hours (1 to 100).</p> <p><b>*IMMED (default)</b> The Capture program prunes old records at the beginning of the specified interval (or immediately), and at each interval thereafter.</p> <p><b>*DELAYED</b> The Capture program prunes old records at the end of the specified interval, and at each interval thereafter.</p> <p><b>*NO</b> The Capture program does not prune records.</p>

Tabella 65. STRDPRCAP command parameter definitions for System i (Continua)

Parameter	Definition and prompts
CAPCTLLIB	<p>Specifies the Capture schema, which is the name of the library in which the Capture control tables reside.</p> <p><b>ASN (default)</b> The default library in which the Capture control tables reside.</p> <p><i>library-name</i> The name of the library in which the Capture control tables reside.</p>
JRN	<p>Specifies a subset of up to 50 journals that you want the Capture program to work with. The Capture program starts processing all the source tables that are currently journaled to this journal.</p> <p><b>*ALL (default)</b> The Capture program starts working with all of the journals that have any source tables journaled to them.</p> <p><i>library-name/journal-name</i> The qualified name of the journal that you want the Capture program to work with. When entering multiple journals, separate the journals with spaces.</p>
TRCLMT	<p>Specifies the trace limit (in minutes). The Capture program prunes any IBMSNAP_CAPTRACE table rows that are older than the trace limit. The default is 10 080 minutes (seven days of trace entries).</p> <p><b>*DFT (default)</b> The Capture program uses the TRACE_LIMIT column value from the IBMSNAP_CAPPARMS table.</p> <p><i>trace-limit</i> The number of minutes of trace data kept in the IBMSNAP_CAPTRACE table after pruning.</p>
MONLMT	<p>Specifies the monitor limit (in minutes). The Capture programs prunes any IBMSNAP_CAPMON table rows that are older than the monitor limit. The default is 10 080 minutes (seven days of monitor entries).</p> <p><b>*DFT (default)</b> The Capture program uses the MONITOR_LIMIT column value from the IBMSNAP_CAPPARMS table.</p> <p><i>monitor-limit</i> The number of minutes of monitor data kept in the IBMSNAP_CAPMON table after pruning.</p>



Tabella 65. STRDPRCAP command parameter definitions for System i (Continua)

Parameter	Definition and prompts
<b>MONITV</b>	<p>Specifies how frequently (in seconds) the Capture program inserts rows into the IBMSNAP_CAPMON table. The default is 300 seconds (five minutes).</p> <p><b>*DFT (default)</b> The Capture program uses the MONITOR_INTERVAL column value from the IBMSNAP_CAPPARMS table.</p> <p><i>monitor-interval</i> The number of seconds between row insertion into the IBMSNAP_CAPMON table. The monitor interval must be at least 120 seconds (two minutes). If you type a number that is less than 120, this parameter value is set to 120.</p>
<b>MEMLMT</b>	<p>Specifies the maximum size (in megabytes) of memory that the Capture journal job can use. The default is 32 megabytes.</p> <p><b>*DFT (default)</b> The Capture program uses the MEMORY_LIMIT column value from the IBMSNAP_CAPPARMS table.</p> <p><i>memory-limit</i> The maximum number of megabytes for memory.</p>
<b>RETAIN</b>	<p>Specifies the new retention limit, which is the number of minutes that data is retained in the change-data (CD), unit-of-work (UOW), IBMSNAP_SIGNAL, and IBMSNAP_AUTHTKN tables before it is removed. This value works with the CLNUPITV parameter value. When the CLNUPITV value is reached, the CD, UOW, IBMSNAP_SIGNAL, and IBMSNAP_AUTHTKN data is removed if this data is older than the retention limit.</p> <p>Ensure that the Apply intervals are set to copy changed information before the data reaches this <b>RETAIN</b> parameter value to prevent inconsistent data in your tables. If the data becomes inconsistent, the Apply program performs a full refresh.</p> <p>The default is 10 080 minutes (seven days). The maximum is 3500000 minutes.</p> <p><b>*DFT (default)</b> The Capture program uses the RETENTION_LIMIT column value from the IBMSNAP_CAPPARMS table.</p> <p><i>retention-limit</i> The number of minutes that the CD, UOW, IBMSNAP_SIGNAL, and IBMSNAP_AUTHTKN data is retained.</p>

Tabella 65. STRDPRCAP command parameter definitions for System i (Continua)

Parameter	Definition and prompts
<b>LAG</b>	<p>Specifies the new lag limit, which is the number of minutes that the Capture program can fall behind in processing before restarting.</p> <p>When the lag limit is reached (that is, when the timestamp of the journal entry is older than the current timestamp minus the lag limit), the Capture program initiates a cold start for the tables that it is processing in that journal. The Apply program then performs a full refresh to provide the Capture program with a new starting point.</p> <p>The default is 10 080 minutes (seven days). The maximum is 35000000 minutes.</p> <p><b>*DFT (default)</b> The Capture program uses the LAG_LIMIT column value from the IBMSNAP_CAPPARMS table.</p> <p><i>lag-limit</i> The number of minutes that the Capture program is allowed to fall behind.</p>
<b>FRCFRQ</b>	<p>Specifies how often (30 to 600 seconds) that the Capture program writes changes to the change-data (CD) and unit-of-work (UOW) tables. The Capture program makes these changes available to the Apply program either when the buffers are filled or when the FRCFRQ time limit expires, whichever is sooner.</p> <p>Use this parameter to make changes more readily available to the Apply program on servers with a low rate of source table changes. The <b>FRCFRQ</b> parameter value is a global value used for all defined source tables. Setting the <b>FRCFRQ</b> value to a low number can affect system performance.</p> <p>The default is 30 seconds.</p> <p><b>*DFT (default)</b> The Capture program uses the COMMIT_INTERVAL column value from the IBMSNAP_CAPPARMS table.</p> <p><i>force-frequency</i> The number of seconds that the Capture program keeps CD and UOW table changes in buffer space before making these changes available to the Apply program.</p>

## Usage notes

The **CLNUPITV** parameter on the STRDPRCAP command specifies the maximum number of hours that the Capture program waits before pruning old records from the change-data (CD), unit-of-work (UOW), IBMSNAP\_SIGNAL, IBMSNAP\_CAPMON, IBMSNAP\_CAPTRACE, and IBMSNAP\_AUTHTKN tables.

You can run the STRDPRCAP command manually, or you can run the command automatically as a part of the initial program load (IPL startup program).

If the job description specified with the **JOB** parameter uses job queue QDP4/QZSNDPR and the DB2 DataPropagator for System i subsystem is not active, the STRDPRCAP command starts the subsystem. If the job description is

defined to use a different job queue and subsystem, you must start this subsystem manually with the Start Subsystem (STRSBS) command either before or after running the STRDPRCAP command:

```
STRSBS QDP4/QZSNDPR
```

You can set up the system to start the subsystem automatically by adding the STRSBS command to the program that is referred to in the QSTRUPPGM system value on your system.

## Restarting Capture by using warm or cold starts

The value of the **RESTART** parameter on the STRDPRCAP command controls how the Capture program handles warm and cold starts.

### Warm start process

Warm start information is saved in most cases. Occasionally, warm start information is not saved. In this case, the Capture program uses the CD tables, UOW table, or the IBMSNAP\_PRUNCNTL table to resynchronize to the time that it was stopped.

### Automatic cold starts

Sometimes the Capture program automatically switches to a cold start, even if you specified a warm start. On System i systems, cold starts work on a journal-by-journal basis. So, for example, if a journal exceeds the lag limit, all replication sources that use the journal are started in cold mode, whereas replication sources that use a different journal are not started in cold mode.

## Examples for STRDPRCAP

The following examples illustrate how to use the STRDPRCAP command.

### Example 1:

To initiate a warm start of a Capture program for two different journals:

```
STRDPRCAP RESTART(*YES) JRN(HR/QSQJRN ACCTS/QSQJRN)
```

### Example 2:

To start a Capture program for one specified journal:

```
STRDPRCAP CAPCTLLIB(BSN) JRN(MARKETING/QSQJRN)
```

The Capture control tables reside under a library named BSN.

### Example 3:

To start a Capture program without pruning for two journals:

```
STRDPRCAP RESTART(*YES) CLNUPITV(*DFT *NO) JRN(HR/QSQJRN ACCTS/QSQJRN)
```

### Example 4:

To start a Capture program for one specified journal under the default Capture control library and to change the default trace limit pruning, monitor limit pruning, IBMSNAP\_CAPMON table insertion, and memory limit parameters:

```
STRDPRCAP CAPCTLLIB(ASN) JRN(SALES/QSQJRN) TRCLMT(1440) MONLMT(1440)  
MONITV(3600) MEMLMT(64)
```

**Example 5:**

To initiate a cold start of a Capture program:

```
STRDPRCAP RESTART(*NO)
```

---

## WRKDPRTTRC: Using the DPR trace facility (System i)

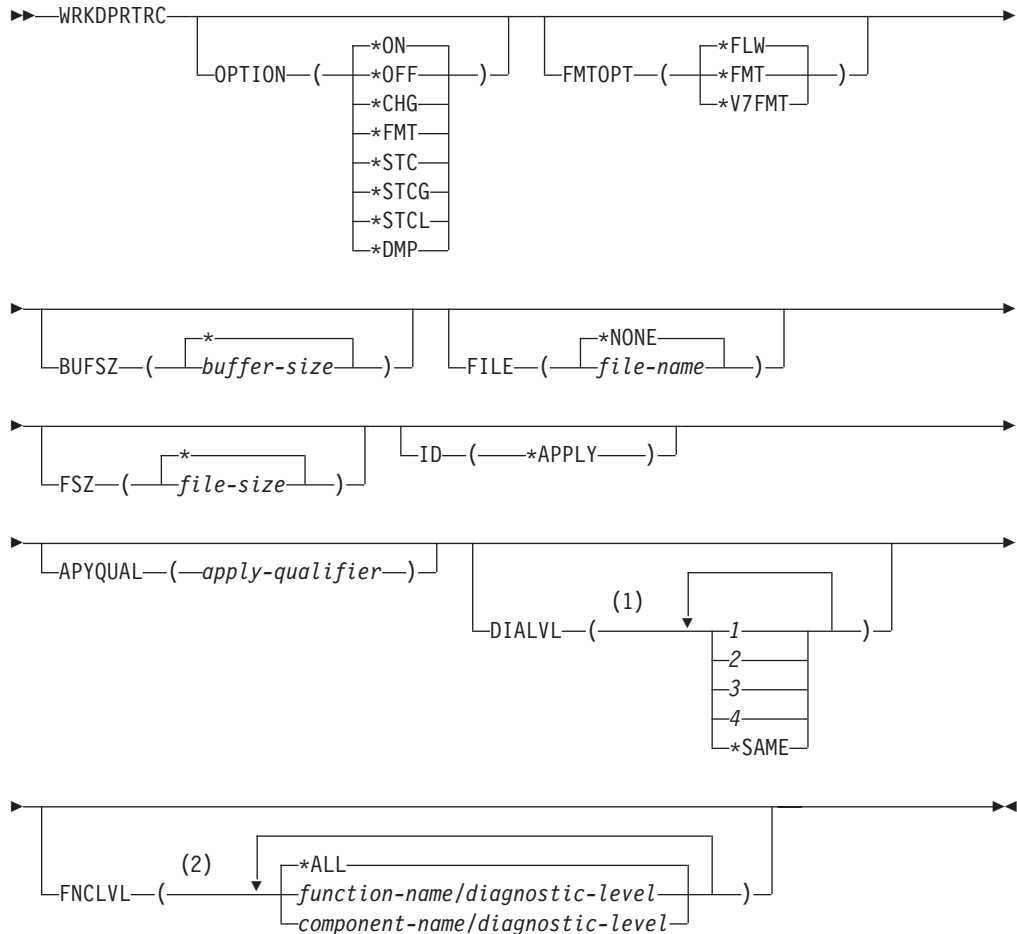
**System i**

Use the DPR trace (WRKDPRTTRC) command only if you are instructed to use the command by IBM software support. The command runs the trace facility to log program flow information for specified Apply programs.

After you type the command name on the command line, you can press the F4 key to display the command syntax.

To display a complete description of this command and all of its parameters, move the cursor to the command at the top of the screen and press the F1 key. To display a description of a specific parameter, place the cursor on that parameter and press the F1 key.

### Syntax



**Note:**

- 1 You can specify multiple values.
- 2 You can specify up to 20 functions or components.

Tabella 66 lists the invocation parameters.

*Tabella 66. WRKDPTRTC command parameter definitions for System i*

Parameter	Definition
OPTION	<p>Specify one trace facility function.</p> <p><b>*ON (default)</b> Turn the trace facility on. This option automatically creates a shared memory segment for tracing.</p> <p><b>*OFF</b> Turn the trace facility off.</p> <p><b>*CHG</b> Change the values of the trace facility parameters.</p> <p><b>*FMT</b> Format the trace facility output from shared memory.</p> <p><b>*STC</b> Display the status of a trace facility. This status information includes the trace version, application version, number of entries, buffer size, amount of buffer used, status code, and program timestamp.</p> <p>This parameter option is equivalent to the stat option of the asntrc command used on UNIX, Windows, and z/OS operating systems.</p> <p><b>*STCG</b> Display the status of a trace facility in Replication Center readable format.</p> <p><b>*STCL</b> Display the status of a trace facility with additional version level information. This additional information includes the service levels of each module in the application and appears as long strings of text.</p> <p>This parameter option is equivalent to the statlong option of the asntrc command used on UNIX, Windows, and z/OS operating systems.</p> <p><b>*DMP</b> Write the current contents of the trace buffer to a file.</p> <p>When prompting on the WRKDPTRTC command, you can press the F4 key to see a list of trace options.</p>

Tabella 66. WRKDPRTTC command parameter definitions for System i (Continua)

Parameter	Definition
FMTOPT	<p>Specifies the options of the format ID and is used with the <b>OPTION(*FMT)</b> parameter.</p> <p><b>*FLW (default)</b> Display the flow of the function calls.</p> <p><b>*FMT</b> Display the format of the trace buffer or trace file. Shows all the detailed data.</p> <p><b>*V7FMT</b> Format the trace buffer or trace file information in Version 7 format.</p> <p>When prompting on the WRKDPRTTC command, you can press the F4 key to see a list of format options.</p>
BUFSZ	<p>Specifies the size (in bytes) of the trace buffer. You can enter an M, K, or G after the number to indicate megabytes, kilobytes, or gigabytes, respectively.</p> <p>The default is two megabytes.</p> <p><b>* (default)</b> Use the two megabyte default size.</p> <p><i>buffer-size</i> The buffer size in bytes.</p>
FILE	<p>Specifies whether the trace output is written to a file.</p> <p><b>*NONE (default)</b> The trace output goes to shared memory only.</p> <p><i>file-name</i> The name of the output file. If you are using the <b>OPTION(*DMP)</b> parameter, this file name represents the name of a dump file.</p>
FSZ	<p>Specifies the size (in bytes) of the file where the trace data is stored. You can enter an M, K, or G after the number to indicate megabytes, kilobytes, or gigabytes, respectively.</p> <p>The default is two gigabytes.</p> <p><b>* (default)</b> Use the two gigabyte default size.</p> <p><i>file-size</i> The file size in bytes.</p>
ID	<p>Specifies the type of program to be traced.</p> <p><b>*APPLY (default)</b> An Apply program trace.</p>
APYQUAL	<p>Specifies the name of Apply program to be traced.</p> <p><i>apply-qualifier</i> The name of the Apply qualifier.</p>

Tabella 66. WRKDPRTTC command parameter definitions for System i (Continua)

Parameter	Definition
DIALVL	<p>Specifies the types of trace records to be recorded by the trace facility. Trace records are categorized by a diagnostic mask number:</p> <ul style="list-style-type: none"> <li>1 Flow data, which includes the entry and exit points of functions.</li> <li>2 Basic data, which includes all major events encountered by the trace facility.</li> <li>3 Detailed data, which includes the major events with descriptions.</li> <li>4 Performance data.</li> </ul> <p><b>*SAME</b> This command uses the diagnostic level settings from the previous trace facility.</p> <p>You can enter one or more diagnostic mask numbers. The numbers that you enter must be in ascending order. Do not type spaces between the numbers.</p> <p><b>Important:</b> The number levels are <i>not</i> inclusive.</p> <p>When you start the trace facility, the default is DIALVL(1234). When you subsequently invoke the trace facility, the default is *SAME.</p> <p>When prompting on the WRKDPRTTC command, you can press the F4 key to see a list of available diagnostic levels.</p>
FNCLVL	<p>Specifies if a particular function or component identifier is to be traced.</p> <p><b>*ALL (default)</b> All functions and components are included in the trace facility.</p> <p><i>function-name/diagnostic-level</i> The name of the function to be traced and the corresponding diagnostic mask numbers.</p> <p><i>component-name/diagnostic-level</i> The name of the component to be traced and the corresponding diagnostic mask numbers.</p> <p>You can enter up to 20 function or component names.</p>

## Examples for WRKDPRTTC

The following examples illustrate how to use the WRKDPRTTC command.

### Example 1:

To start an Apply trace on the Apply qualifier AQ1 for all functions and components with output written to a file called TRCFILE:

```
WRKDPRTTC OPTION(*ON) FILE(TRCFILE) ID(*APPLY) APYQUAL(AQ1)
```

### Example 2:

To end an Apply trace on the Apply qualifier AQ1:

```
WRKDPTRTC OPTION(*OFF) ID(*APPLY) APYQUAL(AQ1)
```

**Example 3:**

To change an Apply trace on the Apply qualifier AQ1 to diagnostic levels 3 and 4 (detailed and performance data) for all functions and components:

```
WRKDPTRTC OPTION(*CHG) ID(*APPLY) APYQUAL(AQ1) DIALVL(34)
```

**Example 4:**

To display the status of an Apply trace on the Apply qualifier AQ1:

```
WRKDPTRTC OPTION(*STC) ID(*APPLY) APYQUAL(AQ1)
```

**Example 5:**

To display the function calls on the Apply qualifier AQ1 at diagnostic levels 3 and 4:

```
WRKDPTRTC OPTION(*FMT) FMTOPT(*FLW) ID(*APPLY) APYQUAL(AQ1) DIALVL (34)
```

**Example 6:**

To write the Apply trace information of the Apply qualifier AQ1 to a dump file named DMPFILE:

```
WRKDPTRTC OPTION(*DMP) FILE(DMPFILE) ID(*APPLY) APYQUAL(AQ1)
```



---

## Capitolo 24. Strutture della tabella di replica SQL

Le tabelle database relazionale sono utilizzate per memorizzare le informazioni per il programma di replica in ogni server: il server di controllo Capture, il server di controllo Apply, il server di controllo Monitor e il server di destinazione. Queste tabelle sono chiamate *tabelle di controllo*.

---

### Tables at a glance

The following diagrams can be used for quick reference to the control tables for the Capture control server, Apply control server, and Monitor control server.

Figura 7 a pagina 408, Figura 8 a pagina 409, and Figura 9 a pagina 410 show the tables at the Capture control server, the columns in each table, and the indexes on each table. Figura 10 a pagina 411 and Figura 11 a pagina 412 show the tables at the Apply control server, the columns in each table, and the indexes on each table. Figura 12 a pagina 413 and Figura 13 a pagina 414 show the tables at the Monitor control server, the columns in each table, and the indexes on each table.

Tabelle di controllo utilizzate nel server di controllo Capture (immagine 1 di 2)

<p><i>Solo OS/400</i></p> <p><b>schema.IBMSNAP_AUTHTKN</b> (nessun indice univoco)</p> <p>APPLY_QUAL CHAR(18) NOT NULL            IBMSNAP_AUTHTKN CHAR(26) NOT NULL            JRN_LIB CHAR(10) NOT NULL            JRN_NAME CHAR(10) NOT NULL            IBMSNAP_LOGMARKER TIMESTAMP NOT NULL</p>	<p><b>ASN.IBMSNAP_CAPSCHEMAS</b> (CAP_SCHEMA_NAME)</p> <p>CAP_SCHEMA_NAME VARCHAR(30)</p> <p><i>OS/400</i></p> <p><b>ASN.IBMSNAP_CAPSCHEMAS</b> (CAP_SCHEMA_NAME)</p> <p>CAP_SCHEMA_NAME VARCHAR(30)            STATUS CHAR(1)</p>
<p><i>Solo UNIX, Windows e z/OS</i></p> <p><b>schema.IBMSNAP_CAPENQ</b> (nessun indice univoco)</p> <p>LOCKNAME CHAR(9)</p>	<p><b>schema.IBMSNAP_CAPTRACE</b> (TRACE_TIME)</p> <p>OPERATION CHAR(8) NOT NULL            TRACE_TIME TIMESTAMP NOT NULL            DESCRIPTION VARCHAR(1024) NOT NULL</p> <p><i>OS/400</i></p> <p><b>schema.IBMSNAP_CAPTRACE</b> (TRACE_TIME)</p> <p>OPERATION CHAR(8) NOT NULL            TRACE_TIME TIMESTAMP NOT NULL            JOB_NAME CHAR(26) NOT NULL            JOB_STR_TIME TIMESTAMP NOT NULL            DESCRIPTION VARCHAR(298) NOT NULL</p>
<p><b>schema.IBMSNAP_CAPMON</b> (MONITOR_TIME)</p> <p>MONITOR_TIME TIMESTAMP NOT NULL            RESTART_TIME TIMESTAMP NOT NULL            CURRENT_MEMORY INT NOT NULL            CD_ROWS_INSERTED INT NOT NULL            RECAP_ROWS_SKIPPED INT NOT NULL            TRIGR_ROWS_SKIPPED INT NOT NULL            CHG_ROWS_SKIPPED INT NOT NULL            TRANS_PROCESSED INT NOT NULL            TRANS_SPILLED INT NOT NULL            MAX_TRAN_SIZE INT NOT NULL            LOCKING_RETRIES INT NOT NULL            JRN_LIB CHAR(10)            JRN_NAME CHAR(10)            LOGREADLIMIT INT NOT NULL            CAPTURE_IDLE INT NOT NULL            SYNCHTIME TIMESTAMP NOT NULL</p>	<p><b>schema.IBMSNAP_PRUNCNTL</b> (SOURCE_OWNER, SOURCE_TABLE, SOURCE_VIEW_QUAL, APPLY_QUAL, SET_NAME, TARGET_SERVER, TARGET_TABLE, TARGET_OWNER, MAP_ID)</p> <p>TARGET_SERVER CHAR(18) NOT NULL            TARGET_OWNER VARCHAR(30) NOT NULL            TARGET_TABLE VARCHAR(128) NOT NULL            SYNCHTIME TIMESTAMP            SYNCHPOINT CHAR(10) FOR BIT DATA            SOURCE_OWNER VARCHAR(30) NOT NULL            SOURCE_TABLE VARCHAR(128) NOT NULL            SOURCE_VIEW_QUAL SMALLINT NOT NULL            APPLY_QUAL CHAR(18) NOT NULL            SET_NAME CHAR(18) NOT NULL            CNTL_SERVER CHAR(18) NOT NULL            TARGET_STRUCTURE SMALLINT NOT NULL            CNTL_ALIAS CHAR(8)            PHYS_CHANGE_OWNER VARCHAR(30)            PHYS_CHANGE_TABLE VARCHAR(128)            MAP_ID VARCHAR(10) NOT NULL</p>
<p><b>schema.IBMSNAP_CAPPARMS</b> (nessun indice univoco)</p> <p>RETENTION_LIMIT INT            LAG_LIMIT INT            COMMIT_INTERVAL INT            PRUNE_INTERVAL INT            TRACE_LIMIT INT            MONITOR_LIMIT INT            MONITOR_INTERVAL INT            MEMORY_LIMIT SMALLINT            REMOTE_SRC_SERVER CHAR(18)            AUTOPRUNE CHAR(1)            TERM CHAR(1)            AUTOSTOP CHAR(1)            LOGREUSE CHAR(1)            LOGSTDOUT CHAR(1)            SLEEPINTERVAL SMALLINT            CAPTURE_PATH VARCHAR(1040)            STARTMODE VARCHAR(10)</p>	<p><b>schema.IBMSNAP_PRUNE_LOCK</b> (nessun indice univoco)</p> <p>DUMMY CHAR(1)</p>

Figura 7. Tables used at the Capture control server. These tables are used by the Capture program, Apply program, and Capture triggers at the Capture control server. The columns that make up each table's main index are listed in parentheses under the table name.

Tabelle di controllo utilizzate nel server di controllo Capture (immagine 2 di 2)

<b>schema.IBMSNAP_PRUNE_SET</b> (TARGET_SERVER, APPLY_QUAL, SET_NAME)		<b>schema.IBMSNAP_REG_SYNCH</b> (TRIGGER_ME)	
TARGET_SERVER	CHAR(18) NOT NULL	TRIGGER_ME	CHAR(1) NOT NULL
APPLY_QUAL	CHAR(18) NOT NULL		
SET_NAME	CHAR(18) NOT NULL		
SYNCHTIME	TIMESTAMP		
SYNCHPOINT	CHAR(10) FOR BIT DATA NOT NULL		
<b>schema.IBMSNAP_REGISTER</b> (SOURCE_OWNER, SOURCE_TABLE, SOURCE_VIEW_QUAL)		<b>schema.IBMSNAP_RESTART</b> (nessun indice univoco)	
SOURCE_OWNER	VARCHAR(30) NOT NULL	MAX_COMMITSEQ	CHAR(10) FOR BIT DATA NOT NULL
SOURCE_TABLE	VARCHAR(128) NOT NULL	MAX_COMMIT_TIME	TIMESTAMP NOT NULL
SOURCE_VIEW_QUAL	VARCHAR(30)	MIN_INFLIGHTSEQ	CHAR(10) FOR BIT DATA NOT NULL
GLOBAL_RECORD	CHAR(1) NOT NULL	CURR_COMMIT_TIME	TIMESTAMP NOT NULL
SOURCE_STRUCTURE	SMALLINT NOT NULL	CAPTURE_FIRST_SEQ	CHAR(10) FOR BIT DATA NOT NULL
SOURCE_CONDENSED	CHAR(1) NOT NULL		
SOURCE_COMPLETE	CHAR(1) NOT NULL		
CD_OWNER	VARCHAR(30)		
CD_TABLE	VARCHAR(128)		
PHYS_CHANGE_OWNER	VARCHAR(30)		
PHYS_CHANGE_TABLE	VARCHAR(128)		
CD_OLD_SYNCHPOINT	CHAR(10) FOR BIT DATA		
CD_NEW_SYNCHPOINT	CHAR(10) FOR BIT DATA		
DISABLE_REFRESH	SMALLINT NOT NULL		
CCD_OWNER	VARCHAR(30)		
CCD_TABLE	VARCHAR(128)		
CCD_OLD_SYNCHPOINT	CHAR(10) FOR BIT DATA		
SYNCHPOINT	CHAR(10) FOR BIT DATA		
SYNCHTIME	TIMESTAMP		
CCD_CONDENSED	CHAR(1)		
CCD_COMPLETE	CHAR(1)		
ARCH_LEVEL	CHAR(4) NOT NULL		
DESCRIPTION	CHAR(254)		
BEFORE_IMG_PREFIX	VARCHAR(4)		
CONFLICT_LEVEL	CHAR(1)		
CHG_UPD_TO_DEL_INS	CHAR(1)		
CHGONLY	CHAR(1)		
RECAPTURE	CHAR(1)		
OPTION_FLAGS	CHAR(4) NOT NULL		
STOP_ON_ERROR	CHAR(1)		
STATE	CHAR(1)		
STATE_INFO	CHAR(8)		
Solo OS/400 <b>schema.IBMSNAP_REG_EXT</b> (VERSION, SOURCE_OWNER, SOURCE_TABLE, SOURCE_VIEW_QUAL)		<b>schema.IBMSNAP_SEQTABLE</b> (SEQ)	
VERSION	INT NOT NULL	SEQ	INTEGER NOT NULL
SOURCE_OWNER	VARCHAR(30) NOT NULL		
SOURCE_TABLE	VARCHAR(128) NOT NULL		
SOURCE_NAME	CHAR(10)		
SOURCE_MBR	CHAR(10)		
SOURCE_TABLE_RDB	CHAR(18)		
JRN_LIB	CHAR(10)		
JRN_NAME	CHAR(10)		
FR_START_TIME	TIMESTAMP		
SOURCE_VIEW_QUAL	SMALLINT NOT NULL		
CMT_BEHAVIOR_CASE	SMALLINT NOT NULL WITH DEFAULT		
MAX_ROWS_BTWN_CMTS	SMALLINT NOT NULL WITH DEFAULT		
		<b>schema.IBMSNAP_SIGNAL</b> (SIGNAL_TIME)	
		SIGNAL_TIME	TIMESTAMP NOT NULL WITH DEFAULT
		SIGNAL_TYPE	VARCHAR(30) NOT NULL
		SIGNAL_SUBTYPE	VARCHAR(30)
		SIGNAL_INPUT_IN	VARCHAR(500)
		SIGNAL_STATE	CHAR(1) NOT NULL
		SIGNAL_LSN	CHAR(10) FOR BIT DATA
		<b>schema.IBMSNAP_UOW</b> (IBMSNAP_COMMITSEQ, IBMSNAP_LOGMARKER)	
		IBMSNAP_UOWID	CHAR(10) FOR BIT DATA NOT NULL
		IBMSNAP_COMMITSEQ	CHAR(10) FOR BIT DATA NOT NULL
		IBMSNAP_LOGMARKER	TIMESTAMP NOT NULL
		IBMSNAP_AUTHTKN	VARCHAR(30) NOT NULL
		IBMSNAP_AUTHID	VARCHAR(30) NOT NULL
		IBMSNAP_REJ_CODE	CHAR(1) NOT NULL WITH DEFAULT
		IBMSNAP_APPLY_QUAL	CHAR(18) NOT NULL WITH DEFAULT

Figura 8. Tables used at the Capture control server (continued). These tables are used by the Capture program, Apply program, and Capture triggers at the Capture control server. The columns that make up each table's main index are listed in parentheses under the table name.

Tabelle di controllo utilizzate nel server di controllo **Monitor** (immagine 3 di 3)

**schema.IBMSNAP\_RESTART**

<i>Solo UNIX, Windows e z/OS</i> (nessun indice)	
MAX_COMMITSEQ	CHAR(10) FOR BIT DATA NOT NULL
MAX_COMMIT_TIME	TIMESTAMP NOT NULL
MIN_INFLIGHTSEQ	CHAR(10) FOR BIT DATA NOT NULL
CURR_COMMIT_TIME	TIMESTAMP NOT NULL
CAPTURE_FIRST_SEQ	CHAR(10) FOR BIT DATA NOT NULL
<i>Solo OS/400</i> (JRN_LIB, JRN_NAME)	
MAX_COMMITSEQ	CHAR(10) FOR BIT DATA NOT NULL
MAX_COMMIT_TIME	TIMESTAMP NOT NULL
MIN_INFLIGHTSEQ	CHAR(10) FOR BIT DATA NOT NULL
CURR_COMMIT_TIME	TIMESTAMP NOT NULL
CAPTURE_FIRST_SEQ	CHAR(10) FOR BIT DATA NOT NULL
UID	INTEGER NOT NULL
SEQNBR	BIGINT NOT NULL
JRN_LIB	CHAR(10) NOT NULL
JRN_NAME	CHAR(10) NOT NULL
STATUS	CHAR(1)

**schema.IBMSNAP\_SIGNAL**

SIGNAL_TIME	TIMESTAMP NOT NULL WITH DEFAULT
SIGNAL_TYPE	VARCHAR(30) NOT NULL
SIGNAL_SUBTYPE	VARCHAR(30)
SIGNAL_INPUT_IN	VARCHAR(500)
SIGNAL_STATE	CHAR(1) NOT NULL
SIGNAL_LSN	CHAR(10) FOR BIT DATA

**schema.IBMSNAP\_UOW**

(IBMSNAP_COMMITSEQ, IBMSNAP_LOGMARKER)	
IBMSNAP_UOWID	CHAR(10) FOR BIT DATA NOT NULL
IBMSNAP_COMMITSEQ	CHAR(10) FOR BIT DATA NOT NULL
IBMSNAP_LOGMARKER	TIMESTAMP NOT NULL
IBMSNAP_AUTHTKN	VARCHAR(30) NOT NULL
IBMSNAP_AUTHID	VARCHAR(30) <sup>1</sup> NOT NULL
IBMSNAP_REJ_CODE	CHAR(1) NOT NULL WITH DEFAULT
IBMSNAP_APPLY_QUAL	CHAR(18) NOT NULL WITH DEFAULT

**schema.IBMSNAP\_SEQTABLE**

<i>Solo Informix</i> (SEQ)	
SEQ	INTEGER NOT NULL

<sup>1</sup> VARCHAR(30) in modalità di compatibilità DB2 per z/OS V8 o precedenti  
<sup>2</sup> VARCHAR(128) in modalità nuova funzione DB2 per z/OS V8

*Figura 9. Tables used at the Capture control server (continued).* These tables are used by the Capture program, Apply program, and Capture triggers at the Capture control server. The columns that make up each table's main index are listed in parentheses under the table name.

Tabelle di controllo utilizzate nel server di controllo **Apply** (immagine 1 di 2)

<b>ASN.IBMSNAP_APPLYTRAIL</b> (nessun indice univoco)	
APPLY_QUAL	CHAR(18) NOT NULL
SET_NAME	CHAR(18) NOT NULL
SET_TYPE	CHAR(1) NOT NULL
WHOS_ON_FIRST	CHAR(1) NOT NULL
ASNLOAD	CHAR(1)
FULL_REFRESH	CHAR(1)
EFFECTIVE_MEMBERS	INT
SET_INSERTED	INT NOT NULL
SET_DELETED	INT NOT NULL
SET_UPDATED	INT NOT NULL
SET_REWORKED	INT NOT NULL
SET_REJECTED_TRXS	INT NOT NULL
STATUS	SMALLINT NOT NULL
LASTRUN	TIMESTAMP NOT NULL
LASTSUCCESS	TIMESTAMP
SYNCHPOINT	CHAR(10) FOR BIT DATA
SYNCHTIME	TIMESTAMP
SOURCE_SERVER	CHAR (18) NOT NULL
SOURCE_ALIAS	CHAR(8)
SOURCE_OWNER	VARCHAR(30)
SOURCE_TABLE	VARCHAR(128)
SOURCE_VIEW_QUAL	SMALLINT
TARGET_SERVER	CHAR(18) NOT NULL
TARGET_ALIAS	CHAR(8)
TARGET_OWNER	VARCHAR(30) NOT NULL
TARGET_TABLE	VARCHAR(128) NOT NULL
CAPTURE_SCHEMA	VARCHAR(30) NOT NULL
TGT_CAPTURE_SCHEMA	VARCHAR(30)
FEDERATED_SRC_SRVR	VARCHAR(18)
FEDERATED_TGT_SRVR	VARCHAR(18)
JRN_LIB	CHAR(10)
JRN_NAME	CHAR(10)
COMMIT_COUNT	SMALLINT
OPTION_FLAGS	CHAR(4) NOT NULL
EVENT_NAME	CHAR(18)
ENDTIME	TIMESTAMP NOT NULL WITH DEFAULT
SOURCE_CONN_TIME	TIMESTAMP
SQLSTATE	CHAR(5)
SQLCODE	INT
SQLERRP	CHAR(8)
SQLERRM	VARCHAR(70)
APPERRM	VARCHAR(760)

<b>ASN.IBMSNAP_APPENQ</b> (APPLY_QUAL)	
APPLY_QUAL	CHAR(18)

<i>Solo OS/400</i> <b>ASN.IBMSNAP_APPLY_JOB</b>	
APPLY_QUAL	CHAR(18) NOT NULL
CONTROL_SERVER	CHAR(18) NOT NULL
JOB_NAME	CHAR(10) NOT NULL
USER_NAME	CHAR(10) NOT NULL
JOB_NUMBER	CHAR(6) NOT NULL

<b>ASN.IBMSNAP_APPLYTRACE</b> (APPLY_QUAL, TRACE_TIME)	
APPLY_QUAL	CHAR(18) NOT NULL
TRACE_TIME	TIMESTAMP NOT NULL
OPERATION	CHAR(8) NOT NULL
DESCRIPTION	VARCHAR(1024) NOT NULL

Figura 10. Tables used at the Apply control server. These tables are used by the Apply program at the Apply control server. The columns that make up each table's main index are listed in parentheses under the table name.

Tabelle di controllo utilizzate nel server di controllo Apply (immagine 2 di 2)

<b>ASN.IBMSNAP_SUBS_COLS</b> (APPLY_QUAL, SET_NAME, WHOS_ON_FIRST, TARGET_OWNER, TARGET_TABLE, TARGET_NAME)		<b>ASN.IBMSNAP_SUBS_STMTS</b> (APPLY_QUAL, SET_NAME, WHOS_ON_FIRST, BEFORE_OR_AFTER, STMT_NUMBER)	
APPLY_QUAL	CHAR(18) NOT NULL	APPLY_QUAL	CHAR(18) NOT NULL
SET_NAME	CHAR(18) NOT NULL	SET_NAME	CHAR(18) NOT NULL
WHOS_ON_FIRST	CHAR(1) NOT NULL	WHOS_ON_FIRST	CHAR(1) NOT NULL
TARGET_OWNER	VARCHAR(30) NOT NULL	BEFORE_OR_AFTER	CHAR(1) NOT NULL
TARGET_TABLE	VARCHAR(128) NOT NULL	STMT_NUMBER	SMALLINT NOT NULL
COL_TYPE	CHAR(1) NOT NULL	EI_OR_CALL	CHAR(1) NOT NULL
TARGET_NAME	VARCHAR(30) NOT NULL	SQL_STMT	VARCHAR(1024)
IS_KEY	CHAR(1) NOT NULL	ACCEPT_SQLSTATES	VARCHAR(50)
COLNO	SMALLINT NOT NULL		
EXPRESSION	VARCHAR(254) NOT NULL		
<b>ASN.IBMSNAP_SUBS_EVENT</b> (EVENT_NAME, EVENT_TIME)		<b>ASN.IBMSNAP_SUBS_MEMBR</b> (APPLY_QUAL, SET_NAME, WHOS_ON_FIRST, SOURCE_OWNER, SOURCE_TABLE, SOURCE_VIEW_QUAL, TARGET_OWNER, TARGET_TABLE)	
EVENT_NAME	CHAR(18) NOT NULL	APPLY_QUAL	CHAR(18) NOT NULL
EVENT_TIME	TIMESTAMP NOT NULL	SET_NAME	CHAR(18) NOT NULL
END_SYNCHPOINT	CHAR(10) FOR BIT DATA	WHOS_ON_FIRST	CHAR(1) NOT NULL
END_OF_PERIOD	TIMESTAMP	SOURCE_OWNER	VARCHAR(30) NOT NULL
		SOURCE_TABLE	VARCHAR(128) NOT NULL
		SOURCE_VIEW_QUAL	SMALLINT NOT NULL
		TARGET_OWNER	VARCHAR(30) NOT NULL
		TARGET_TABLE	VARCHAR(128) NOT NULL
		TARGET_CONDENSED	CHAR(1) NOT NULL
		TARGET_COMPLETE	CHAR(1) NOT NULL
		TARGET_STRUCTURE	SMALLINT NOT NULL
		PREDICATES	VARCHAR(1024)
		MEMBER_STATE	CHAR(1)
		TARGET_KEY_CHG	CHAR(1) NOT NULL
		UOW_CD_PREDICATES	VARCHAR(1024)
		JOIN_UOW_CD	CHAR(1)
		LOADX_TYPE	SMALLINT
		LOADX_SRC_N_OWNER	VARCHAR(30)
		LOADX_SRC_N_TABLE	VARCHAR(128)
<b>ASN.IBMSNAP_SUBS_SET</b> (APPLY_QUAL, SET_NAME, WHOS_ON_FIRST)			
APPLY_QUAL	CHAR(18) NOT NULL		
SET_NAME	CHAR(18) NOT NULL		
SET_TYPE	CHAR(1) NOT NULL		
WHOS_ON_FIRST	CHAR(1) NOT NULL		
ACTIVATE	SMALLINT NOT NULL		
SOURCE_SERVER	CHAR(18) NOT NULL		
SOURCE_ALIAS	CHAR(8)		
TARGET_SERVER	CHAR(18) NOT NULL		
TARGET_ALIAS	CHAR(8)		
STATUS	SMALLINT NOT NULL		
LASTRUN	TIMESTAMP NOT NULL		
REFRESH_TYPE	CHAR(1) NOT NULL		
SLEEP_MINUTES	INT		
EVENT_NAME	CHAR(18)		
LASTSUCCESS	TIMESTAMP		
SYNCHPOINT	CHAR(10) FOR BIT DATA		
SYNCHTIME	TIMESTAMP		
CAPTURE_SCHEMA	VARCHAR(30) NOT NULL		
TGT_CAPTURE_SCHEMA	VARCHAR(30)		
FEDERATED_SRC_SRVR	VARCHAR(18)		
FEDERATED_TGT_SRVR	VARCHAR(18)		
JRN_LIB	CHAR(10)		
JRN_NAME	CHAR(10)		
OPTION_FLAGS	CHAR(4) NOT NULL		
COMMIT_COUNT	SMALLINT		
MAX_SYNCH_MINUTES	SMALLINT		
AUX_STMTS	SMALLINT NOT NULL		
ARCH_LEVEL	CHAR(4) NOT NULL		

Figura 11. Tables used at the Apply control server (continued). These tables are used by the Apply program at the Apply control server. The columns that make up each table's main index are listed in parentheses under the table name.



Tabelle di controllo utilizzate nel server di controllo **Monitor** (immagine 2 di 2)

ASN.IBMSNAP_MONSERVERS		ASN.IBMSNAP_MONTRAIL	
(MONITOR_QUAL, SERVER_NAME)		(nessun indice)	
MONITOR_QUAL	CHAR(18) NOT NULL	MONITOR_QUAL	CHAR(18) NOT NULL
SERVER_NAME	CHAR(18) NOT NULL	SERVER_NAME	CHAR(18) NOT NULL
SERVER_ALIAS	CHAR(8)	SERVER_ALIAS	CHAR(8)
LAST_MONITOR_TIME	TIMESTAMP NOT NULL	STATUS	SMALLINT NOT NULL
START_MONITOR_TIME	TIMESTAMP	LASTRUN	TIMESTAMP NOT NULL
END_MONITOR_TIME	TIMESTAMP	LASTSUCCESS	TIMESTAMP
LASTRUN	TIMESTAMP NOT NULL	ENDTIME	TIMESTAMP NOT NULL
LASTSUCCESS	TIMESTAMP		WITH DEFAULT
STATUS	SMALLINT NOT NULL	LAST_MONITOR_TIME	TIMESTAMP NOT NULL
		START_MONITOR_TIME	TIMESTAMP
		END_MONITOR_TIME	TIMESTAMP
		SQLCODE	INT
		SQLSTATE	CHAR(5)
		NUM_ALERTS	INT NOT NULL
		NUM_NOTIFICATIONS	INT NOT NULL
ASN.IBMSNAP_MONTRACE			
(MONITOR_QUAL, TRACE_TIME)			
MONITOR_QUAL	CHAR(18) NOT NULL		
TRACE_TIME	TIMESTAMP NOT NULL		
OPERATION	CHAR(8) NOT NULL		
DESCRIPTION	VARCHAR(1024) NOT NULL		

Figura 13. Tables used at the Monitor control server (continued). These tables are used by the Replication Alert Monitor program at the Monitor control server. The columns that make up each table's main index are listed in parentheses under the table name.

## Tabelle nel server di controllo Capture

Le tabelle memorizzate nel server di controllo Capture contengono informazioni riguardo le origini registrate e in che modo il programma Capture o i trigger elaborano le origini.

Per Linux, UNIX, Windows e z/OS, è possibile creare queste tabelle di controllo secondo le proprie specifiche utilizzando il programma riga di comando ASNCLP o il Centro di replica. Per i System i, tali tabelle di controllo sono create automaticamente nella libreria ASN quando si installa DataPropagator per System i. È possibile utilizzare i comandi System i per creare le tabelle di controllo Capture negli schemi di acquisizione alternativi.

Tabella 67 descrive le tabelle di controllo nel server Capture.

Tabella 67. Consultazione rapida per le tabelle utilizzate nel server di controllo Capture

Nome tabella	Descrizione
"Tabella IBMSNAP_CAPSCHEMAS"	Contiene i nomi di tutti gli schemi di Capture a pagina 423
Tabella IBMSNAP_AUTHTKN (System i)	Contiene informazioni di supporto per la replica update-anywhere.



Tabella 67. Consultazione rapida per le tabelle utilizzate nel server di controllo Capture (Continua)

Nome tabella	Descrizione
<p><b>z/OS</b></p> <p><b>Linux UNIX Windows</b> “Tabella IBMSNAP_CAPENQ (z/OS, Linux, UNIX, Windows)” a pagina 417</p>	<p>Per ogni schema Capture, questa tabella è utilizzata per garantire che:</p> <ul style="list-style-type: none"> <li>• <b>Linux UNIX Windows</b> Per DB2 per Linux, UNIX e Windows, solo un programma Capture è in esecuzione per database.</li> <li>• <b>z/OS</b> Per dati non condivisi DB2 per z/OS, solo un programma Capture è in esecuzione per sottosistema.</li> <li>• <b>z/OS</b> Per dati condivisi DB2 per z/OS, solo un programma Capture è in esecuzione per il gruppo di dati condivisi.</li> </ul>
“Tabella CD” a pagina 427	Contiene informazioni riguardo le modifiche che si verificano all’origine. Questa tabella non è creata fino a che non viene registrata un’origine di replica.
“Tabella CCD (non DB2)” a pagina 426	Contiene informazioni riguardo le modifiche che si verificano nell’origine e le colonne aggiuntive per identificate l’ordine sequenziale di queste modifiche.
“Tabella IBMSNAP_CAPMON” a pagina 418	Contiene statistiche operative che aiutano il controllo di avanzamento del programma Capture.
“Tabella IBMSNAP_CAPPARMS” a pagina 420	Contiene parametri che possono essere specificati per controllare le operazioni del programma Capture.
“Tabella IBMSNAP_CAPTRACE” a pagina 425	Contiene i messaggi del programma Capture.
“Tabella IBMQREP_IGNTRAN” a pagina 428	Può essere utilizzato per indicare al programma Capture le transazioni che non si desidera che vengano acquisite dalla registrazione di recupero DB2.
“Tabella IBMQREP_IGNTRANTRC” a pagina 429	Registra le informazioni sulle transazioni specificate come da ignorare.
“Tabella IBMSNAP_PARTITIONINFO” a pagina 430	Contiene le informazioni che abilitano il programma Capture ad effettuare il riavvio dai primi numeri di sequenze di registrazione richiesti.
“Tabella IBMSNAP_PRUNE_LOCK” a pagina 433	Utilizzato per serializzare l’accesso del programma Capture delle tabelle CD durante un avvio a freddo o durante l’eliminazione del limite di conservazione (eliminazione quando il limite di conservazione è raggiunto o superato).
“Tabella IBMSNAP_PRUNE_SET” a pagina 434	Coordina l’eliminazione delle tabelle CD.
“tabella IBMSNAP_PRUNCNTL” a pagina 431	Coordina gli aggiornamenti del punto di sincronizzazione fra i programmi Capture e Apply.
<p><b>System i</b></p> <p>IBMSNAP_REG_EXT (System i)</p>	Un’estensione della tabella di registro. Contiene informazioni aggiuntive riguardo le origini di replica, come il nome di giornale e il nome di voce database della tabella di origine remota.
“Tabella IBMSNAP_REGISTER” a pagina 436	Contiene informazione sulle origini di replica, come i nomi delle tabelle origine di replica, i loro attributi e i nomi delle loro tabelle CD e CCD corrispondenti.

Tabella 67. Consultazione rapida per le tabelle utilizzate nel server di controllo Capture (Continua)

Nome tabella	Descrizione
"Tabella IBMSNAP_REG_SYNCH (relazionale non DB2)" a pagina 443	Utilizzato quando si effettua la replica da un'origine di dati relazionale non-DB2. Un trigger di aggiornamento su questa tabella simula il programma Capture avviando un aggiornamento del valore di SYNCHPOINT per tutte le righe nella tabella di registro prima che il programma Apply legga le informazioni dalla tabella di registro.
"Tabella IBMSNAP_RESTART" a pagina 443	Contiene informazioni che abilitano il programma Capture a riprendere l'acquisizione dal punto corretto nella registrazione o nel giornale. Per gli ambienti System i, questa tabella è utilizzata anche per determinare l'ora di avvio del comando RCVJRNE (Ricezione voce di giornale).
"Tabella IBMSNAP_SEQTABLE (Informix)" a pagina 446	Contiene una sequenza di numeri univoci che la replica SQL utilizza come l'equivalente dei numeri di sequenza di registrazione per le tabelle Informix.
"Tabella IBMSNAP_SIGNAL" a pagina 446	Contiene tutti i segnali utilizzati per richiedere un programma Capture. Questi segnali possono essere inviati manualmente o dal programma Apply.
"Tabella IBMSNAP_UOW" a pagina 450	Fornisce informazioni aggiuntive riguardo le transazioni per le quali è stato eseguito il commit verso una tabella di origine.

## IBMSNAP\_AUTHTKN table (System i)

### System i

The IBMSNAP\_AUTHTKN table is used in the System i environment only. This table is used during update-anywhere replication to keep track of the transactions that have been processed by a particular Apply program. The Capture program prunes this table based on the retention limit that you set.

**Server:** Capture control server

**Default schema:** ASN

**Index:** JRN\_LIB, JRN\_NAME

**Important:** Use caution when you update this table by using SQL. Altering this table inappropriately can cause unexpected results and loss of data.

Tabella 68 provides a brief description of the columns in the IBMSNAP\_AUTHTKN table.

Tabella 68. Columns in the IBMSNAP\_AUTHTKN table

Column name	Description
APPLY_QUAL	<b>Data type:</b> CHAR(18); <b>Nullable:</b> No  The Apply qualifier that identifies which Apply program processed the transaction. This qualifier is used during update-anywhere replication to prevent the Apply program from replicating the same changes repeatedly.

Tabella 68. Columns in the IBMSNAP\_AUTHTKN table (Continua)

Column name	Description
IBMSNAP_AUTHTKN	<b>Data type:</b> CHAR(26); <b>Nullable:</b> No  The job name that is associated with the transaction. Capture for System i matches the name in this column with the name of the job that issued the transaction to determine whether the transaction was issued by either the Apply program or a user application. If the job names match, then Capture for System i copies the Apply qualifier that's in the APPLY_QUAL column of this table to the APPLY_QUAL column in the corresponding row of the UOW table. If the names do not match, then Capture for System i sets the APPLY_QUAL column of the UOW row to null. This column is not automatically copied to other tables; you must select it and copy it as a user data column.
JRN_LIB	<b>Data type:</b> CHAR(10); <b>Nullable:</b> No  The library name of the journal from which the transactions came.
JRN_NAME	<b>Data type:</b> CHAR(10); <b>Nullable:</b> No  The name of the journal from which the transactions came.
IBMSNAP_LOGMARKER	<b>Data type:</b> TIMESTAMP; <b>Nullable:</b> No  The approximate time that the transaction was committed at the Capture control server.

## Tabella IBMSNAP\_CAPENQ (z/OS, Linux, UNIX, Windows)

z/OS

Linux UNIX Windows

Per uno schema di Capture singolo, la tabella IBMSNAP\_CAPENQ garantisce che solo un programma Capture sia in esecuzione per database, sottosistema, o gruppo di condivisione dati.

**Server:** server di controllo Capture

**Schema predefinito:** ASN

**Indice:** Nessuno

**Importante:** Prestare attenzione quando si aggiorna questa tabella utilizzando SQL. La modifica non corretta di questa tabella può provocare risultati non previsti e perdita di dati.

La tabella IBMSNAP\_CAPENQ non è utilizzata sui sistemi relazionali non-DB2 o sui server System i.

Durante l'esecuzione, il programma Capture blocca esclusivamente questa tabella.

Tabella 69 fornisce una breve descrizione della colonna nella tabella IBMSNAP\_CAPENQ.

Tabella 69. Colonna nella tabella IBMSNAP\_CAPENQ

Nome colonna	Descrizione
LOCKNAME	<b>Tipo di dati:</b> CHAR(9); <b>Impostabile su null:</b> Sì  Questa colonna non contiene dati.

## Tabella IBMSNAP\_CAPMON

Il programma Capture inserisce una riga in una tabella IBMSNAP\_CAPMON dopo ogni intervallo per fornire statistiche operative. Il Centro di replica utilizza le informazioni contenute in questa tabella (e in altre tabelle) in modo che sia possibile controllare lo stato del programma Capture.

**Server:** server di controllo Capture

**Schema predefinito:** ASN

**Indice:** MONITOR\_TIME

Nella tabella IBMSNAP\_CAPPARMS, il valore specificato per il MONITOR\_INTERVAL indica la frequenza con cui il programma Capture effettua degli inserimenti all'interno della tabella di controllo Capture e il valore specificato per il MONITOR\_LIMIT indica il numero di minuti in cui le righe rimangono nella tabella prima di essere idonee per l'eliminazione.

Tabella 70 fornisce una breve descrizione sulle colonne nella tabella IBMSNAP\_CAPMON.

Tabella 70. Colonne nella tabella IBMSNAP\_CAPMON

Nome colonna	Descrizione
MONITOR_TIME	<b>Tipo di dati:</b> TIMESTAMP; <b>Impostabile su null:</b> no  La data/ora (nel server di controllo Capture) quando la riga è stata inserita in questa tabella.
RESTART_TIME	<b>Tipo di dati:</b> TIMESTAMP; <b>Impostabile su null:</b> no  La data/ora quando il richiamo corrente del programma Capture è stato riavviato.
CURRENT_MEMORY	<b>Tipo di dati:</b> INT; <b>Impostabile su null:</b> No  La quantità di memoria (in byte) utilizzata dal programma Capture.
CD_ROWS_INSERTED	<b>Tipo di dati:</b> INT; <b>Impostabile su null:</b> No  Il numero di righe che il programma Capture ha inserito nella tabella CD per tutte le tabelle origine.
RECAP_ROWS_SKIPPED	<b>Tipo di dati:</b> INT; <b>Impostabile su null:</b> No  Per le repliche update-anywhere, questo è il numero di righe che il programma Capture ha elaborato ma non ha inserito nella tabella CD. Le righe sono state saltate poiché la registrazione è stata definita affinché il programma Capture non acquisisca di nuovo le modifiche che sono state replicate su questa tabella che non è stata creata in questo server di origine.
TRIGR_ROWS_SKIPPED	<b>Tipo di dati:</b> INT; <b>Impostabile su null:</b> No  Il numero di righe che il programma Capture elabora ma non inserisce nella tabella CD. Le righe sono state saltate poiché durante la registrazione del programma Capture è stato definito un trigger che ignora alcune righe.
CHG_ROWS_SKIPPED	<b>Tipo di dati:</b> INT; <b>Impostabile su null:</b> No  Il numero di righe che il programma Capture elabora ma non inserisce nella tabella CD. Le righe sono state saltate poiché la registrazione è stata definita affinché il programma Capture acquisisca solamente le modifiche che si verificano nelle colonne registrate.

Tabella 70. Colonne nella tabella IBMSNAP\_CAPMON (Continua)


Nome colonna	Descrizione
TRANS_PROCESSED	<b>Tipo di dati:</b> INT; <b>Impostabile su null:</b> No  Il numero delle transazioni al sistema di origine elaborate dal programma Capture.
TRANS_SPILLED	<b>Tipo di dati:</b> INT; <b>Impostabile su null:</b> No  Il numero di transazioni al sistema di origine trasferite dal programma Capture al disco a causa delle restrizioni di memoria.
MAX_TRAN_SIZE	<b>Tipo di dati:</b> INT; <b>Impostabile su null:</b> No  La transazione di dimensioni più elevate che si è verificata nel sistema di origine. Conoscere la dimensione della transazione potrebbe influenzare l'utente nella modifica dei parametri della memoria.
LOCKING_RETRIES	<b>Tipo di dati:</b> INT; <b>Impostabile su null:</b> No  Il numero di volte che un deadlock ha provocato un'elaborazione.
JRN_LIB (System i)	<b>Tipo di dati:</b> CHAR(10); <b>Impostabile su null:</b> Sì   Il nome della libreria di giornale che il programma Capture sta elaborando.
JRN_NAME (System i)	<b>Tipo di dati:</b> CHAR(10); <b>Impostabile su null:</b> Sì  Il nome di giornale che il programma Capture sta processando.
LOGREADLIMIT	<b>Tipo di dati:</b> INT; <b>Impostabile su null:</b> No  Il numero di volte in cui il programma Capture è in pausa dalla lettura dei record di registrazione poiché sono stati letti 1000 record ma nessuna transazione completa è stata ancora incontrata fino a questi 1000 record.
CAPTURE_IDLE	<b>Tipo di dati:</b> INT; <b>Impostabile su null:</b> No  Il numero di volte in cui il programma Capture si disattiva poiché non c'è nessun lavoro da elaborare.
SYNCHTIME	<b>Tipo di dati:</b> TIMESTAMP; <b>Impostabile su null:</b> no  Il valore corrente di SYNCHTIME legge dalla riga globale della tabella di registro quando il record di controllo è stato inserito in questa tabella.
CURRENT_LOG_TIME	<b>Tipo di dati:</b> TIMESTAMP; <b>Impostabile su null:</b> no  La data/ora dell'ultimo commit di database a livello del server Capture visualizzato dal programma di lettura della registrazione Capture.
RESTART_SEQ	<b>Tipo di dati:</b> CHAR(10) PER DATI BIT; <b>Impostabile su null:</b> No  Il numero di sequenza della registrazione logica della registrazione di recupero da cui viene avviato il programma Capture durante un riavvio caldo. Questo valore rappresenta il primo numero di sequenza della registrazione che il programma Capture ha individuato, per il quale un commit record o un record di fine anomala non è ancora stato individuato.
CURRENT_SEQ	<b>Tipo di dati:</b> CHAR(10) PER DATI BIT; <b>Impostabile su null:</b> No  Il numero di sequenza della registrazione logica più recente della registrazione di recupero letta dal programma Capture.

Tabella 70. Colonne nella tabella IBMSNAP\_CAPMON (Continua)

Nome colonna	Descrizione
LAST_EOL_TIME	<b>Tipo di dati:</b> TIMESTAMP; <b>Impostabile su null:</b> Sì  L'ora del server Capture in cui il programma Capture ha raggiunto la fine della registrazione.

## Tabella IBMSNAP\_CAPPARMS

La tabella IBMSNAP\_CAPPARMS contiene i parametri che si possono modificare per controllare le operazioni del programma Capture. È possibile definire questi parametri per impostare valori come il periodo di tempo in cui il programma Capture mantiene i dati nelle tabelle CD e UOW prima dell'eliminazione e la quantità di tempo per la quale il programma Capture è abilitato a ritardare l'elaborazione dei record di registrazione. Se vengono apportate delle modifiche ai parametri in questa tabella, il programma Capture legge le modifiche solo durante l'avvio.

**Server:** server di controllo Capture

**Schema predefinito:** ASN

**Indice:** Nessuno

Questa tabella contiene le informazioni che è possibile aggiornare utilizzando SQL.

Tabella 71 fornisce una breve descrizione delle colonne nella tabella IBMSNAP\_CAPPARMS.

Tabella 71. Colonne nella tabella IBMSNAP\_CAPPARMS

Nome colonna	Descrizione
RETENTION_LIMIT	<b>Tipo di dati:</b> INT; <b>Impostabile su null:</b> Sì  Il periodo di tempo in cui le righe rimangono nelle tabelle CD, UOW e segnale prima di diventare idonee per l'eliminazione, in situazioni in cui non sono state eliminate secondo i normali criteri. Normalmente, le righe CD e UOW vengono eliminate dopo esser state applicate a tutte le destinazioni e le righe segnale vengono eliminate quando il loro ciclo è completo (SIGNAL_STATE = C).
LAG_LIMIT	<b>Tipo di dati:</b> INT; <b>Impostabile su null:</b> Sì  Il numero di minuti per cui è permesso al programma Capture di ritardare l'elaborazione prima che si chiuda. Durante periodi di alta frequenza di aggiornamento, un aggiornamento completo può essere più economico rispetto ad aggiornamenti singoli.
COMMIT_INTERVAL	<b>Tipo di dati:</b> INT; <b>Impostabile su null:</b> Sì  La frequenza, in secondi, con cui il programma Capture esegue il commit dei dati alle tabelle di controllo Capture, incluse le tabelle UOW e CD. Questo valore dovrebbe essere minore del valore di lockout DB2 per impedire conflitti tra i thread Capture ed eliminazione.

Tabella 71. Colonne nella tabella IBMSNAP\_CAPPARMS (Continua)

Nome colonna	Descrizione
PRUNE_INTERVAL	<p><b>Tipo di dati:</b> INT; <b>Impostabile su null:</b> Sì</p> <p>La frequenza, in secondi, con cui il programma Capture elimina automaticamente(AUTOPRUNE = Y) le righe nelle tabelle CD, UOW, segnale, traccia e controllo Capture che non sono più necessarie. Un intervallo di eliminazione minore risparmia spazio, ma aumenta i costi di elaborazione. Un intervallo di eliminazione maggiore richiede più tablespace CD e UOW, ma diminuisce i costi di elaborazione.</p>
TRACE_LIMIT	<p><b>Tipo di dati:</b> INT; <b>Impostabile su null:</b> Sì</p> <p>Il numero di minuti in cui le righe rimangono nella tabella IBMSNAP_CAPTRACE prima di essere idonee per l'eliminazione. Durante il processo di eliminazione, se il numero di minuti (data/ora corrente - l'ora in cui una riga è stata introdotta nella tabella di traccia Capture) supera il valore di TRACE_LIMIT, le righe nella tabella traccia di Capture vengono cancellate.</p>
MONITOR_LIMIT	<p><b>Tipo di dati:</b> INT; <b>Impostabile su null:</b> Sì</p> <p>Il numero di minuti in cui le righe rimangono nella tabella IBMSNAP_CAPMON prima che siano idonee per l'eliminazione. Durante il processo di eliminazione, le righe nella tabella di controllo Capture vengono eliminate se il valore dei minuti (data/ora corrente - MONITOR_TIME) supera il valore del MONITOR_LIMIT.</p>
MONITOR_INTERVAL	<p><b>Tipo di dati:</b> INT; <b>Impostabile su null:</b> Sì</p> <p>La frequenza, in secondi, con cui il thread di controllo aggiunge una riga alla tabella di controllo Capture IBMSNAP_CAPMON. Per Capture per System i, inserire un intervallo maggiore di 120 secondi.</p>
MEMORY_LIMIT	<p><b>Tipo di dati:</b> SMALLINT; <b>Impostabile su null:</b> Sì</p> <p>La quantità di memoria, in megabyte, che il programma Capture ha il permesso di utilizzare. Dopo aver utilizzato questa assegnazione, le transazioni in memoria verranno trasferite in un file.</p>
REMOTE_SRC_SERVER	<p><b>Tipo di dati:</b> CHAR(18); <b>Impostabile su null:</b> Sì</p> <p>Riservato per future opzioni della replica SQL. Attualmente questa colonna contiene il valore predefinito di null.</p>
AUTOPRUNE	<p><b>Tipo di dati:</b> CHAR(1); <b>Impostabile su null:</b> Sì</p> <p>Un indicatore segnala se il programma Capture elimina automaticamente le righe che non sono più necessarie dalle tabelle CD, UOW, segnale, traccia e controllo Capture:</p> <p>Y      L'eliminazione automatica è attiva.</p> <p>N      L'eliminazione automatica è disattivata.</p>
TERM	<p><b>Tipo di dati:</b> CHAR(1); <b>Impostabile su null:</b> Sì</p> <p>Un indicatore che segnala se il programma Capture termina quando DB2 si è arrestato o è in stato di sospensione:</p> <p>Y      Il programma Capture termina quando DB2 si arresta o è in stato di sospensione.</p> <p>N      Il programma Capture rimane attivo e attende che DB2 sia riavviato o esca dallo stato di sospensione.</p>

Tabella 71. Colonne nella tabella IBMSNAP\_CAPPARMS (Continua)

Nome colonna	Descrizione
AUTOSTOP	<p><b>Tipo di dati:</b> CHAR(1); <b>Impostabile su null:</b> Sì</p> <p>Un indicatore segnala se il programma Capture arresta l'acquisizione delle modifiche appena raggiunge la fine delle registrazioni attive:</p> <p><b>Y</b> Il programma Capture si arresta quando raggiunge la fine delle registrazioni attive.</p> <p><b>N</b> Il programma Capture continua l'esecuzione quando raggiunge la fine delle registrazioni attive.</p>
LOGREUSE	<p><b>Tipo di dati:</b> CHAR(1); <b>Impostabile su null:</b> Sì</p> <p>Un indicatore segnala se il programma Capture sovrascrive il file di log Capture oppure lo aggiunge.</p> <p><b>Y</b> Il programma Capture riutilizza il file di log innanzitutto eliminandolo e poi ricreandolo quando il programma Capture viene riavviato.</p> <p><b>N</b> Il programma Capture aggiunge una nuova informazione al file di log Capture.</p>
LOGSTDOUT	<p><b>Tipo di dati:</b> CHAR(1); <b>Impostabile su null:</b> Sì</p> <p>Un indicatore segnala dove il programma Capture indirizza i messaggi del file di log:</p> <p><b>Y</b> Il programma Capture indirizza i messaggi del file di log allo standard out (STDOUT) e al file di log.</p> <p><b>N</b> Il programma Capture indirizza la maggior parte dei messaggi del file di registrazione solamente al file di registrazione. I messaggi di inizializzazione vengono memorizzati sia nell'out standard (STDOUT) che nel file di log.</p>
<div style="background-color: #800000; color: white; padding: 2px; display: inline-block; margin-bottom: 5px;">z/OS</div> <div style="background-color: #800000; color: white; padding: 2px; display: inline-block; margin-bottom: 5px;">Linux UNIX Windows</div> SLEEP_INTERVAL (z/OS, Linux, UNIX, Windows)	<p><b>Tipo di dati:</b> SMALLINT; <b>Impostabile su null:</b> Sì</p> <p>Il numero di secondi in cui il programma Capture si disattiva quando raggiunge la fine delle registrazioni attive (in Linux, UNIX e Windows, oppure in ambienti dati non condivisi z/OS ), oppure quando viene restituita una quantità inefficiente di dati (in ambienti di dati condivisi z/OS).</p>
CAPTURE_PATH	<p><b>Tipo di dati:</b> VARCHAR(1040); <b>Impostabile su null:</b> Sì</p> <p>Il percorso dove è inviato l'output dal programma Capture.</p>



Tabella 71. Colonne nella tabella IBMSNAP\_CAPPARMS (Continua)

Nome colonna	Descrizione
STARTMODE	<p><b>Tipo di dati:</b> VARCHAR(10); <b>Impostabile su null:</b> Sì</p> <p>Il processo di elaborazione che il programma Capture utilizza quando viene avviato:</p> <p><b>a freddo</b>                      Il programma Capture elimina tutte le righe nelle proprie tabelle CD e nella tabella UOW durante l'inizializzazione. Tutte le richieste a queste origini di replica vengono completamente aggiornate durante il nuovo ciclo di elaborazione Apply (ovvero, tutti i dati vengono copiati dalle tabelle di origine alle tabelle di destinazione). Se il programma Capture tenta un avvio a freddo ma è stato disabilitato l'aggiornamento completo, si avvierà il programma Capture ma il programma Apply fallirà e immetterà un messaggio di errore.</p> <p><b>warmsi</b> Il programma Capture si avvia a caldo; se questa è la prima volta che si avvia il programma Capture, esso passerà all'avvio a freddo. La modalità di avvio warmsi garantisce che il programma Capture venga avviato a freddo solo quando viene avviato per la prima volta.</p> <p><b>warmns</b>                      Il programma Capture si avvia a caldo. Se non può essere avviato a caldo, esso non passa all'avvio a freddo. La modalità di avvio warmns impedisce che l'avvio a freddo sia effettuato in modo non previsto ed è utile quando sorgono problemi (come database non disponibili o tablespace) che richiedono una risoluzione e che impediscono un avvio a caldo. Quando il programma Capture viene avviato a caldo, riprende l'elaborazione dal punto in cui era terminata. In caso di errori dopo l'avvio del programma Capture, il programma termina e lascia tutte le tabelle intatte.</p>

## Tabella IBMSNAP\_CAPSCHEMAS

La tabella IBMSNAP\_CAPSCHEMAS contiene i nomi di tutti gli schemi Capture. Essa permette agli strumenti di gestione e ad altri programmi di utilità di trovare velocemente tutte le tabelle per un determinato server di controllo Capture. Viene inserita automaticamente una riga ogni volta che viene creato un nuovo schema Capture.

**Server:** server di controllo Capture

**Indice:** CAP\_SCHEMA\_NAME

**Importante:** Prestare attenzione quando si aggiorna questa tabella utilizzando SQL. La modifica non corretta di questa tabella può provocare risultati non previsti.

Le seguenti due tabelle mostrano i layout dei singoli sistemi operativi della tabella IBMSNAP\_CAPSCHEMAS.

Tabella 72. Colonne nella tabella IBMSNAP\_CAPSCHEMAS per tutti i sistemi operativi diversi da System i

Nome colonna	Descrizione
CAP_SCHEMA_NAME	<p><b>Tipo di dati:</b> VARCHAR(128); <b>Impostabile su null:</b> Sì</p> <p>Il nome dello schema Capture. Esiste una riga per ogni schema Capture.</p>

Tabella 73. Colonne nella tabella schemi Capture per System i

Nome colonna	Descrizione
CAP_SCHEMA_NAME	<b>Tipo di dati:</b> VARCHAR(30); <b>Impostabile su null:</b> Sì  Il nome dello schema Capture. Esiste una riga per ogni schema Capture.
STATUS	<b>Tipo di dati:</b> CHAR(1); <b>Impostabile su null:</b> Sì  Un indicatore segnala se il programma Capture che è identificato da questo schema Capture è in esecuzione:  Y      Il programma Capture è in esecuzione. N      Il programma Capture non è in esecuzione.

## Tabella IBMQREP\_COLVERSION (z/OS)

z/OS

La tabella IBMQREP\_COLVERSION viene creata sui sistemi operativi z/OS per consentire a Q Capture e ai programmi Capture di tenere traccia delle diverse versioni di una tabella di origine.

**Server:** server Q Capture

**Schema predefinito:** ASN

**Indice univoco:** LSN, TABLEID1, TABLEID2, POSITION

**Importante:** non modificare questa tabella utilizzando SQL. La modifica non corretta di questa tabella può provocare risultati non previsti e perdita di dati.

Il programma Q Capture o Capture inserisce le righe in questa tabella quando la registrazione o la sottoscrizione Q per una tabella origine viene attivata per prima e, quindi, ogni volta che la tabella di origine viene modificata.

ASN è l'unico schema consentito per questa tabella.

Tabella 74 fornisce una breve descrizione delle colonne nella tabella IBMQREP\_TRG\_COLS.

Tabella 74. Colonne nella tabella IBMQREP\_COLVERSION

Nome colonna	Descrizione
LSN	<b>Tipo di dati:</b> CHAR(10) FOR BIT DATA; <b>Impostabile su null:</b> no  Il punto nella registrazione di recupero DB2 dove il programma Q Capture o Capture ha individuato una nuova versione della tabella origine.
TABLEID1	<b>Tipo di dati:</b> SMALLINT; <b>Impostabile su null:</b> no  L'OBID (object identifier) in SYSIBM.SYSTABLES.
TABLEID2	<b>Tipo di dati:</b> SMALLINT; <b>Impostabile su null:</b> no  Il DBID (database identifier) in SYSIBM.SYSTABLES.
POSITION	<b>Tipo di dati:</b> SMALLINT; <b>Impostabile su null:</b> no  La posizione ordinale della colonna nella tabella, cominciando da 0 per la prima colonna nella tabella.

Tabella 74. Colonne nella tabella IBMQREP\_COLVERSION (Continua)

Nome colonna	Descrizione
NOME	<b>Tipo di dati:</b> VARCHAR(128); <b>Impostabile su null:</b> no Il nome della colonna.
TYPE	<b>Tipo di dati:</b> SMALLINT; <b>Impostabile su null:</b> no Un identificatore del tipo di dati interno per la colonna (SQLTYPE in SYSIBM.SYSCOLUMNS).
LENGTH	<b>Tipo di dati:</b> INTEGER; <b>Impostabile su null:</b> no La lunghezza massima dei dati per questa colonna.
NULLS	<b>Tipo di dati:</b> CHAR(1); <b>Impostabile su null:</b> no Un identificatore che segnala se la colonna consente i valori non validi: Y La colonna consente i valori non validi. N La colonna non consente i valori non validi.
DEFAULT	<b>Tipo di dati:</b> VARCHAR(1536); <b>Impostabile su null:</b> sì Il valore predefinito della colonna (DEFAULTVALUE) in SYSIBM.SYSCOLUMNS. Questa colonna è NULL se non presenta un valore predefinito.
CODEPAGE	<b>Tipo di dati:</b> INTEGER; <b>Impostabile su null:</b> sì La tabella origine utilizzata per i dati di questa colonna. Il valore è 0 se la colonna viene definita come FOR BIT DATA oppure non è un tipo di stringa. Valore predefinito: NULL
SCALE	<b>Tipo di dati:</b> INTEGER; <b>Impostabile su null:</b> sì La scala dei dati decimali nelle colonne a valori decimali. Il valore è 0 per le colonne a valori non decimali. Valore predefinito: NULL

## Tabella IBMSNAP\_CAPTRACE

La tabella traccia Capture contiene i messaggi del programma Capture.

**Server:** server di controllo Capture

**Schema predefinito:** ASN

**Indice:** TRACE\_TIME

Le seguenti due tabelle mostrano i layout dei singoli sistemi operativi della tabella IBMSNAP\_CAPTRACE. z/OS Linux UNIX Windows

Tabella 75. Colonne nella tabella IBMSNAP\_CAPTRACE per Linux, UNIX, Windows e z/OS

Nome colonna	Descrizione
OPERATION	<b>Tipo di dati:</b> CHAR(8); <b>Impostabile su null:</b> No Il tipo di operazione del programma Capture, ad esempio, inizializzazione, acquisizione, o condizione di errore.
TRACE_TIME	<b>Tipo di dati:</b> TIMESTAMP; <b>Impostabile su null:</b> No L'ora del server di controllo Capture in cui è stata inserita una riga nella tabella traccia Capture.

Tabella 75. Colonne nella tabella IBMSNAP\_CAPTRACE per Linux, UNIX, Windows e z/OS (Continua)

Nome colonna	Descrizione
DESCRIPTION	<p><b>Tipo di dati:</b> VARCHAR(1024); <b>Impostabile su null:</b> No</p> <p>L'ID del messaggio seguito dal testo del messaggio. Potrebbe essere un messaggio d'errore, un messaggio di avviso, o un messaggio d'informazione. Questa colonna solo testo in inglese.</p>

**System i**

Tabella 76. Colonne nella tabella di traccia Capture per System i

Nome colonna	Descrizione								
OPERATION	<p><b>Tipo di dati:</b> CHAR(8); <b>Impostabile su null:</b> No</p> <p>Il tipo di operazione eseguita dal programma Capture, ad esempio, l'inizializzazione, l'acquisizione, o la condizione di errore.</p>								
TRACE_TIME	<p><b>Tipo di dati:</b> TIMESTAMP; <b>Impostabile su null:</b> No</p> <p>L'ora in cui è stata inserita una riga nella tabella traccia Capture. Le righe TRACE_TIME che sono idonee per l'eliminazione del limite traccia saranno cancellate quando il programma Capture cancella le tabelle CD e UOW.</p>								
JOB_NAME	<p><b>Tipo di dati:</b> CHAR(26); <b>Impostabile su null:</b> No</p> <p>Il nome completo del lavoro scritto in questa voce di traccia.</p> <p><b>Posizione</b></p> <table border="0"> <thead> <tr> <th></th> <th>Descrizione</th> </tr> </thead> <tbody> <tr> <td>1-10</td> <td>Il nome dello schema Capture o il nome del lavoro giornale</td> </tr> <tr> <td>11-20</td> <td>L'ID dell'utente che ha avviato il programma Capture</td> </tr> <tr> <td>21-26</td> <td>Il numero del lavoro</td> </tr> </tbody> </table>		Descrizione	1-10	Il nome dello schema Capture o il nome del lavoro giornale	11-20	L'ID dell'utente che ha avviato il programma Capture	21-26	Il numero del lavoro
	Descrizione								
1-10	Il nome dello schema Capture o il nome del lavoro giornale								
11-20	L'ID dell'utente che ha avviato il programma Capture								
21-26	Il numero del lavoro								
JOB_STR_TIME	<p><b>Tipo di dati:</b> TIMESTAMP; <b>Impostabile su null:</b> No</p> <p>L'ora d'inizio del lavoro indica nella colonna JOB_NAME.</p>								
DESCRIPTION	<p><b>Tipo di dati:</b> VARCHAR(298); <b>Impostabile su null:</b> No</p> <p>L'ID del messaggio seguito dal testo del messaggio. L'ID del messaggio è rappresentato dai primi sette caratteri della colonna DESCRIPTION. Il testo del messaggio inizia dalla nona posizione della colonna DESCRIPTION.</p>								

## Tabella CCD (non DB2)

Le tabelle CCD (Consistent-change-data) sul server di controllo Capture sono tabelle che contengono informazioni sulle modifiche che si verificano nelle colonne non DB2 di origine e aggiuntive per identificare l'ordinamento sequenziale di queste modifiche. Una tabella CCD sul server di controllo Capture è una tabella popolata da un programma diverso dal programma Apply.

**Server:** server di controllo Capture

**Importante:** Prestare attenzione quando si aggiorna questa tabella utilizzando SQL. La modifica non corretta di questa tabella può provocare una perdita di dati.

Il server di controllo Capture può essere:

- Una tabella CCD interna per un'origine relazionale non DB2.

Per la replica di acquisizione delle modifiche, i trigger Capture inseriscono le modifiche in questa tabella come gli aggiornamenti si verificano nell'origine relazionale non DB2. Il nome di questo tipo di tabella CCD è memorizzato nella stessa riga nella tabella IBMSNAP\_REGISTER dell'origine di replica che da cui gestisce le modifiche. Questa tabella viene eliminata automaticamente dal trigger di eliminazione creato quando viene registrata un'origine relazionale non DB2.

- Una tabella CCD esterna per dati non relazionali e multivendor.

I programmi esterni possono cercare tabelle CCD che saranno utilizzate dalla replica SQL come origini di replica. Ad esempio, IMS DataPropagator acquisisce le modifiche IMS in una tabella CCD, affinché sia possibile ricreare copie di dati IMS in un database relazionale. I programmi esterni devono inizializzare, mantenere e fornire i valori corretti per le colonne di controllo. Se esistono tabelle CCD popolate esternamente che non sono mantenute da un programma come IMS DataPropagator o DataRefresher, è necessario che l'utente mantenga queste tabelle in modo che il programma Apply possa leggere le tabelle CCD come origini e funzionare correttamente.

Tabella 77 fornisce una breve descrizione delle colonne nella tabella CCD.

Tabella 77. Colonne nella tabella CCD

Nome colonna	Descrizione
IBMSNAP_INTENTSEQ	Un numero di sequenza che identifica una modifica in modo univoco. Questo valore è globalmente ascendente.
IBMSNAP_OPERATION	Un indicatore che segnala il tipo di operazione per un record: <b>I</b> Inserimento <b>U</b> Aggiornamento <b>D</b> Eliminazione
IBMSNAP_COMMITSEQ	Un numero di sequenza che fornisce l'ordine transazionale.
IBMSNAP_LOGMARKER	L'ora in cui è stato eseguito il commit dei dati.
<i>colonne chiave utente</i>	Se la tabella CCD è concentrata, questa colonna contiene le colonne che compongono la chiave di destinazione.
<i>colonna non chiave utente</i>	La colonna di dati non chiave dalla tabella di origine. I nomi delle colonne nella tabella di origine non devono corrispondere ai nomi di queste colonne, ma i tipi di dati devono essere compatibili.
<i>colonne calcolate utente</i>	Colonne definite dall'utente derivate da espressioni SQL. È possibile utilizzare colonne calcolate con funzioni SQL per convertire i tipi di dati di origine in tipi di dati di destinazione differenti.

## Tabella CD

Le tabelle CD (Change-data) registrano tutte le modifiche di cui è stato eseguito il commit apportate ad un'origine di replica. L'eliminazione della tabella CD è coordinata dalla tabella IBMSNAP\_PRUNE\_SET. A differenza delle altre tabelle di controllo Capture, le tabelle CD vengono create quando viene definita un'origine di replica; non vengono create automaticamente quando vengono create le tabelle di controllo per il server di controllo Capture.

**Server:** server di controllo Capture

**Importante:** Prestare attenzione quando si aggiorna questa tabella utilizzando SQL. La modifica non corretta di questa tabella può provocare una perdita di dati.

Tabella 78 fornisce una lista e una breve descrizione delle colonne nella tabella CD.

Tabella 78. Colonne nella tabella CD

Nome colonna	Descrizione
IBMSNAP_COMMITSEQ	Il numero della sequenza di registrazione acquisita della quale è stato effettuato il commit. Questa colonna, presente anche nella tabella UOW, è inclusa nella tabella CD in modo da permettere al programma Apply di elaborare le tabelle di destinazione copia utente senza che sia necessario unire la tabella CD con la tabella UOW. Nei casi in cui è necessaria l'integrazione fra la tabella CD e la tabella UOW, l'integrazione è eseguita utilizzando la colonna IBMSNAP_COMMITSEQ.
IBMSNAP_INTENTSEQ	Il numero della sequenza di registrazione del record di registrazione della modifica (inserisci, aggiorna, o cancella). Questo valore è globalmente ascendente. Se si sceglie che gli aggiornamenti vengano elaborati come coppie di elimina/inserisci, il valore IBMSNAP_INTENTSEQ per l'eliminazione riga è progettato per essere leggermente più piccolo del corrispondente valore d'inserimento riga.
IBMSNAP_OPERATION	Un indicatore che segnala il tipo di operazione per un record: <b>I</b> Inserimento <b>U</b> Aggiornamento <b>D</b> Eliminazione
<i>post-immagine colonna utente</i>	Nella maggior parte dei casi, la colonna post-immagine contiene il valore presente nella colonna d'origine dopo che la modifica si è verificata. Questa colonna ha lo stesso nome, lo stesso tipo di dati e gli stessi attributi null della colonna d'origine. Nel caso di un aggiornamento, questa colonna fa riferimento al nuovo valore dei dati che sono stati aggiornati. Nel caso di una eliminazione, questa colonna fa riferimento al valore dei dati che sono stati cancellati. Nel caso di un inserimento, questa colonna fa riferimento al valore dei dati che sono stati inseriti.
<i>pre-immagine colonna utente</i>	Questa colonna esiste solamente nella tabella CD se è stata registrata l'origine in modo da includere i valori di colonna pre-immagine. Nella maggior parte dei casi, la colonna pre-immagini contiene il valore presente nella colonna d'origine prima che si verificasse la modifica. Questa colonna ha lo stesso nome della colonna d'origine, prefissata dal valore presente nella colonna BEFORE_IMG_PREFIX nella tabella IBMSNAP_REGISTER. Inoltre, presenta lo stesso tipo di dati della colonna d'origine; tuttavia, permette sempre valori null per le operazioni indipendenti d'inserimento degli attributi null della colonna d'origine. Nel caso di un aggiornamento, questa colonna fa riferimento ai dati che sono stati aggiornati. Nel caso di una eliminazione, questa colonna fa riferimento ai dati che sono stati eliminati. Nel caso di un inserimento, questa colonna ha valore null.

## Tabella IBMQREP\_IGNTRAN

La tabella IBMQREP\_IGNTRAN può essere utilizzata per indicare a Q Capture o al programma Capture le transazioni che non si desidera che vengano acquisite dalla registrazione di recupero DB2. È possibile utilizzare SQL per inserire righe nella tabella che informa i programmi di ignorare le transazioni basate su un ID di autorizzazione, un token di autorizzazione (solo z/OS) o nome di piano (solo z/OS).

**Server:** server Q Capture, server di controllo Capture

**Schema predefinito:** ASN

Tabella 79 offre una breve descrizione delle colonne nella tabella IBMQREP\_IGNTRAN.

Tabella 79. Colonne nella tabella IBMQREP\_IGNTRAN

Nome colonna	Descrizione
AUTHID	<b>Tipo di dati:</b> CHAR(128); <b>Impostabile su null:</b> Sì  L'ID di autorizzazione principale relativo alla transazione che si desidera ignorare.
AUTHTOKEN	<b>Tipo di dati:</b> CHAR(30); <b>Impostabile su null:</b> Sì  <b>z/OS</b> Il token di autorizzazione (nome lavoro) relativo alla transazione che si desidera ignorare.
PLANNAME	<b>Tipo di dati:</b> CHAR(8); <b>Impostabile su null:</b> Sì.  <b>z/OS</b> Il nome piano relativo alla transazione che si desidera ignorare.
IGNTRANTRC	<b>Tipo di dati:</b> CHAR(1); <b>Impostabile su null:</b> No, con valore predefinito  Un indicatore dice al programma Q Capture o Capture se tracciare le transazioni ignorate in base al valore di AUTHID, AUTHTOKEN o PLANNAME specificato nella tabella IBMQREP_IGNTRAN:  <b>Y (valore predefinito)</b> La traccia è abilitata. Ogni volta che una transazione è ignorata, viene inserita una riga nella tabella IBMQREP_IGNTRANTRC e viene emesso un messaggio.  <b>N</b> La traccia è disabilitata.

## Tabella IBMQREP\_IGNTRANTRC

La tabella IBMQREP\_IGNTRANTRC registra le informazioni relative alle transazioni per le quali è stato specificato che devono essere ignorate.

**Server:** server Q Capture, server di controllo Capture

**Schema predefinito:** ASN

**Importante:** Non alterare questa tabella mediante SQL. La modifica non corretta di questa tabella può provocare risultati non previsti e perdita di dati.

Viene inserita una riga nella tabella IBMQREP\_IGNTRANTRC quando è ignorata una transazione nella registrazione di recupero DB2. Questa tabella è eliminata in base al parametro **trace\_limit** relativo al programma Q Capture o Capture.

Tabella 80 offre una breve descrizione delle colonne nella tabella IBMQREP\_IGNTRANTRC.

Tabella 80. Colonne nella tabella IBMQREP\_IGNTRANTRC

Nome colonna	Descrizione
IGNTRAN_TIME	<b>Tipo di dati:</b> TIMESTAMP; <b>Impostabile su null:</b> No, con valore predefinito  L'ora in cui la transazione è ignorata. Valore predefinito: data/ora corrente
AUTHID	<b>Tipo di dati:</b> CHAR(128); <b>Impostabile su null:</b> Sì  L'ID di autorizzazione primario della transazione ignorata.

Tabella 80. Colonne nella tabella IBMQREP\_IGNTRANTRC (Continua)

Nome colonna	Descrizione
AUTHTOKEN	<b>Tipo di dati:</b> CHAR(30); <b>Impostabile su null:</b> Sì  z/OS Il token di autorizzazione (nome lavoro) relativo alla transazione ignorata.
PLANNAME	<b>Tipo di dati:</b> CHAR(8); <b>Impostabile su null:</b> Sì.  z/OS Il nome piano relativo alla transazione ignorata.
TRANSID	<b>Tipo di dati:</b> CHAR(10) PER DATI BIT; <b>Impostabile su null:</b> No  L'identificatore della transazione relativo alla transazione ignorata.
COMMITLSN	<b>Tipo di dati:</b> CHAR(10) PER DATI BIT; <b>Impostabile su null:</b> No  Il numero di sequenza o la sequenza temporale della registrazione di commit relativi alla transazione ignorata.

## Tabella IBMSNAP\_PARTITIONINFO

La tabella IBMSNAP\_PARTITIONINFO aumenta la tabella IBMSNAP\_RESTART in un ambiente con più partizioni e contiene le informazioni che abilitano il programma Capture ad effettuare il riavvio dai primi numeri di sequenze di registrazione richieste senza ogni serie di partizioni dei file di log.

**Server:** server di controllo Capture

**Schema predefinito:** ASN

**Indice:** PARTITIONID, USAGE

**Importante:** Prestare attenzione quando si aggiorna questa tabella utilizzando SQL. La modifica non corretta di questa tabella può provocare risultati non previsti e perdita di dati. Se viene eliminata la riga da questa tabella, il programma Capture viene forzato ad eseguire l'avvio freddo.

In un ambiente con più partizioni, la tabella IBMSNAP\_PARTITIONINFO e la tabella IBMSNAP\_RESTART sostituiscono la tabella IBMSNAP\_WARM\_START dalla replica SQL nella Versione 7 e nelle versioni precedenti. Viene inserita una riga all'interno della tabella ogni volta che viene aggiunta una partizione. Il programma Capture inizierà la lettura del file di log di tutte le nuove partizioni dal primo numero di sequenza di registrazione che DB2 ha utilizzato prima che fosse immesso il primo database CONNECT.

Se il programma Capture non è stato mai avviato, quindi questa tabella è vuota, ed è necessario che il programma Capture effettui un avvio a freddo.

Tabella 81 fornisce una breve descrizione delle colonne nella tabella IBMSNAP\_PARTITIONINFO.

Tabella 81. Colonne nella tabella IBMSNAP\_PARTITIONINFO

Nome colonna	Descrizione
PARTITIONID	<b>Tipo di dati:</b> INT; <b>Impostabile su null:</b> No  L'ID partizione per ogni partizione valida.



Tabella 81. Colonne nella tabella IBMSNAP\_PARTITIONINFO (Continua)

Nome colonna	Descrizione
USAGE	<b>Tipo di dati:</b> CHAR(1); <b>Impostabile su null:</b> no  L'utilizzo dell'LSN (log sequence number). Una "R" in questa colonna indica che l'LSN è stato riavviato.
SEQUENCE	<b>Tipo di dati:</b> CHAR(10) per dati bit; <b>Impostabile su null:</b> No  LSN di riavvio per il nodo che presenta l'ID partizione.
STATUS	<b>Tipo di dati:</b> CHAR(1); <b>Impostabile su null:</b> Sì  Lo stato della partizione. Una A in questa colonna indica che la partizione è attiva. Questa colonna è riservato per un utilizzo futuro.
LAST_UPDATE	<b>Tipo di dati:</b> TIMESTAMP; <b>Impostabile su null:</b> Sì  Data/ora di ultimo aggiornamento dell'LSN di riavvio della partizione che dispone dell'ID partizione in questione.

## tabella IBMSNAP\_PRUNCNTL

La tabella di controllo eliminazione contiene informazioni dettagliate su tutti i membri di impostazione sottoscrizione che sono definiti come uno schema Capture. Questa tabella è utilizzata insieme alla tabella IBMSNAP\_PRUNE\_SET durante l'eliminazione. È inoltre utilizzata durante il processo d'inizializzazione handshake tra i programmi Apply e Capture.

**Server:** server di controllo Capture

**Schema predefinito:** ASN

**Indice:** SOURCE\_OWNER, SOURCE\_TABLE, SOURCE\_VIEW\_QUAL, APPLY\_QUAL, SET\_NAME, TARGET\_SERVER, TARGET\_TABLE, TARGET\_OWNER

**Importante:** Prestare attenzione quando si aggiorna questa tabella utilizzando SQL. La modifica non corretta di questa tabella può provocare risultati non previsti e perdita di dati.

Per le origini DB2, è possibile richiamare l'eliminazione immettendo il comando prune o impostandolo in automatico. Per le origini relazionali non-DB2, l'eliminazione viene effettuata dal trigger di eliminazione che è stato creato quando è stata registrata l'origine.

Tabella 82 fornisce una breve descrizione delle colonne nella tabella IBMSNAP\_PRUNCNTL.

Tabella 82. Colonne nella tabella IBMSNAP\_PRUNCNTL

Nome colonna	Descrizione
TARGET_SERVER	<b>Tipo di dati:</b> CHAR(18); <b>Impostabile su null:</b> no  Il nome del server nel quale risiede la tabella o la vista di destinazione per questo membro.

Tabella 82. Colonne nella tabella IBMSNAP\_PRUNCNTL (Continua)

Nome colonna	Descrizione
TARGET_OWNER	<p><b>Tipo di dati:</b> VARCHAR(30), VARCHAR(128) per sottosistemi in modalità nuova funzione DB2 UDB per z/OS Versione 8; <b>Impostabile su null:</b> No</p> <p>Il qualificatore di alto livello per la tabella o vista di destinazione per questo membro.</p>
TARGET_TABLE	<p><b>Tipo di dati:</b> VARCHAR(128), VARCHAR(18) per sottosistemi in modalità di compatibilità DB2 UDB per z/OS Versione 8 o precedenti; <b>Impostabile su null:</b> No</p> <p>Il nome della tabella o vista di destinazione per questo membro.</p>
SYNCHTIME	<p><b>Tipo di dati:</b> TIMESTAMP; <b>Impostabile su null:</b> Sì</p> <p>Il programma Capture imposta questa data/ora durante il processo d’inizializzazione handshake con il programma Apply. Il valore deriva dalla data/ora del record di registrazione commit associato con la transazione dell’inserimento del segnale CAPSTART. Esso sarà aggiornato di nuovo a meno che non si verifichi un successivo processo di inizializzazione.</p>
SYNCHPOINT	<p><b>Tipo di dati:</b> CHAR(10) per dati bit; <b>Impostabile su null:</b> Sì</p> <p>Il programma Capture imposta questo valore durante il processo d’inizializzazione handshake con il programma Apply. Il valore deriva dal numero della sequenza di registrazione del record di registrazione commit associato con la transazione dell’inserimento del segnale CAPSTART. Esso sarà aggiornato di nuovo a meno che non si verifichi un successivo processo di inizializzazione.</p>
SOURCE_OWNER	<p><b>Tipo di dati:</b> VARCHAR(30), VARCHAR(128) per sottosistemi in modalità nuova funzione DB2 UDB per z/OS Versione 8; <b>Impostabile su null:</b> No</p> <p>Il qualificatore di altro livello della tabella o della vista di origine per questo membro.</p>
SOURCE_TABLE	<p><b>Tipo di dati:</b> VARCHAR(128), VARCHAR(18) per sottosistemi in modalità di compatibilità DB2 UDB per z/OS Versione 8 o precedenti; <b>Impostabile su null:</b> No</p> <p>Il nome della tabella o vista di origine per questo membro.</p>
SOURCE_VIEW_QUAL	<p><b>Tipo di dati:</b> SMALLINT; <b>Impostabile su null:</b> no</p> <p>Questa colonna viene utilizzata per supportare più registrazioni per differenti viste di origine con identici valori di colonna SOURCE_OWNER e SOURCE_TABLE. Questo valore è impostato su 0 per le tabelle fisiche definite come origini ed è maggiore di 0 per le viste che sono definite come origini.</p>
APPLY_QUAL	<p><b>Tipo di dati:</b> CHAR(18); <b>Impostabile su null:</b> no</p> <p>IL qualificatore Apply che identifica quale programma Apply sta elaborando questo membro.</p>
SET_NAME	<p><b>Tipo di dati:</b> CHAR(18); <b>Impostabile su null:</b> no</p> <p>Il nome della serie di richieste alla quale appartiene questo membro della serie di richieste.</p>
CNTL_SERVER	<p><b>Tipo di dati:</b> CHAR(18); <b>Impostabile su null:</b> no</p> <p>IL nome del server nel quale risiedono le tabelle di controllo Apply per questo programma Apply, che è identificato dall’APPLY_QUAL.</p>

Tabella 82. Colonne nella tabella IBMSNAP\_PRUNCNTL (Continua)

Nome colonna	Descrizione																
TARGET_STRUCTURE	<p><b>Tipo di dati:</b> SMALLINT; <b>Impostabile su null:</b> no</p> <p>Un valore che identifica il tipo della tabella o della vista di destinazione:</p> <table> <tr> <td>1</td> <td>Tabella origine</td> </tr> <tr> <td>3</td> <td>Tabella CCD</td> </tr> <tr> <td>4</td> <td>Tabella oraria</td> </tr> <tr> <td>5</td> <td>Tabella aggregati di base</td> </tr> <tr> <td>6</td> <td>Tabella aggregati di modifica</td> </tr> <tr> <td>7</td> <td>Tabella di replica</td> </tr> <tr> <td>8</td> <td>Tabella copia utente</td> </tr> <tr> <td>9</td> <td>Tabella CCD senza un'unione delle tabelle IBMSNAP_UOW e CD</td> </tr> </table>	1	Tabella origine	3	Tabella CCD	4	Tabella oraria	5	Tabella aggregati di base	6	Tabella aggregati di modifica	7	Tabella di replica	8	Tabella copia utente	9	Tabella CCD senza un'unione delle tabelle IBMSNAP_UOW e CD
1	Tabella origine																
3	Tabella CCD																
4	Tabella oraria																
5	Tabella aggregati di base																
6	Tabella aggregati di modifica																
7	Tabella di replica																
8	Tabella copia utente																
9	Tabella CCD senza un'unione delle tabelle IBMSNAP_UOW e CD																
CNTL_ALIAS	<p><b>Tipo di dati:</b> CHAR(8); <b>Impostabile su null:</b> Sì</p> <p>L'alias DB2 corrispondente al server di controllo Apply indicato nella colonna CNTL_SERVER.</p>																
PHYS_CHANGE_OWNER	<p><b>Tipo di dati:</b> VARCHAR(30), VARCHAR(128) per sottosistemi in modalità nuova funzione DB2 UDB per z/OS Versione 8; <b>Impostabile su null:</b> Sì</p> <p>Il valore nella colonna PHYS_CHANGE_OWNER dalla tabella IBMSNAP_REGISTER associata con l'origine di questo particolare membro della serie di richieste.</p>																
PHYS_CHANGE_TABLE	<p><b>Tipo di dati:</b> VARCHAR(128), VARCHAR(18) per sottosistemi in modalità di compatibilità DB2 UDB per z/OS Versione 8 o precedente; <b>Impostabile su null:</b> Sì</p> <p>Il valore della colonna PHYS_CHANGE_TABLE dalla tabella IBMSNAP_REGISTER associata con l'origine di questo particolare membro della serie di richieste.</p>																
MAP_ID	<p><b>Tipo di dati:</b> VARCHAR(10); <b>Impostabile su null:</b> No</p> <p>Un fattore univoco che fornisce un indice più breve, più facile da utilizzare in questa tabella. È inoltre utilizzato per associare gli inserimenti CAPSTART nella tabella segnale con la riga appropriata nella tabella di controllo eliminazione.</p>																

## Tabella IBMSNAP\_PRUNE\_LOCK

La tabella IBMSNAP\_PRUNE\_LOCK è utilizzata per serializzare l'accesso alle tabelle CD durante un avvio a freddo o una cancellazione del limite di conservazione. Questa tabella garantisce che il programma Apply non acceda alla tabella CD durante queste fasi critiche. Non ci sono righe in questa tabella.

**Server:** server di controllo Capture

**Schema predefinito:** ASN

**Indice:** Nessuno

**Importante:** Prestare attenzione quando si aggiorna questa tabella utilizzando SQL. La modifica non corretta di questa tabella può provocare risultati non previsti e perdita di dati.

## Tabella IBMSNAP\_PRUNE\_SET

La tabella IBMSNAP\_PRUNE\_SET tiene traccia dell'avanzamento dei programmi Capture e Apply per ogni serie di richieste in modo da aiutare la coordinazione dell'eliminazione delle tabelle CD e UOW. A differenza della tabella IBMSNAP\_PRUNCNTL, che presenta una riga per ogni associazione dall'origine alla destinazione, la tabella IBMSNAP\_PRUNE\_SET presenta una riga per ogni serie di sottoscrizioni.

**Server:** server di controllo Capture

**Schema predefinito:** ASN

**Indice:** TARGET\_SERVER, APPLY\_QUAL, SET\_NAME

**Importante:** Prestare attenzione quando si aggiorna questa tabella utilizzando SQL. La modifica non corretta di questa tabella può provocare risultati non previsti e perdita di dati.

Tabella 83 fornisce una breve descrizione delle colonne nella tabella IBMSNAP\_PRUNE\_SET.

Tabella 83. Colonne nella tabella IBMSNAP\_PRUNE\_SET

Nome colonna	Descrizione
TARGET_SERVER	<b>Tipo di dati:</b> CHAR(18); <b>Impostabile su null:</b> no Il nome del server nel quale risiedono le tabelle o le viste di destinazione per questa serie.
APPLY_QUAL	<b>Tipo di dati:</b> CHAR(18); <b>Impostabile su null:</b> no Il qualificatore Apply che identifica quale programma Apply sta elaborando questa serie.
SET_NAME	<b>Tipo di dati:</b> CHAR(18); <b>Impostabile su null:</b> no Il nome della serie di richieste.
SYNCHTIME	<b>Tipo di dati:</b> TIMESTAMP; <b>Impostabile su null:</b> Sì IL programma Apply utilizza questa colonna per registrare il proprio avanzamento, segnalando che ha elaborato i dati fino a questa data/ora per la serie di richieste.
SYNCHPOINT	<b>Tipo di dati:</b> CHAR(10) per dati bit; <b>Impostabile su null:</b> No Il programma Apply utilizza questa colonna per registrare il proprio avanzamento, segnalando che ha elaborato i dati fino a questo valore synchpoint per la serie di richieste.

## IBMSNAP\_REG\_EXT (System i)

System i

The IBMSNAP\_REG\_EXT table is a System i-specific table that provides supplemental information for the IBMSNAP\_REGISTER table. For every row in the IBMSNAP\_REGISTER table, there is a matching row in the IBMSNAP\_REG\_EXT table that contains additional System i-specific columns.

**Server:** Capture control server

**Default schema:** ASN

**Index:** VERSION, SOURCE\_OWNER, SOURCE\_TABLE, SOURCE\_VIEW\_QUAL

**Important:** Use caution when you update this table by using SQL. Altering this table inappropriately can cause unexpected results and loss of data.

This table is maintained by a trigger program (program QZSNJLV8 in library QDP4) on the IBMSNAP\_REGISTER table. The trigger is defined at the time the IBMSNAP\_REGISTER table is created.

The information from this table is used to track where and how you defined your replication sources on an System i server.

Tabella 84 provides a brief description of the columns in the IBMSNAP\_REG\_EXT table.

*Tabella 84. Columns in the IBMSNAP\_REG\_EXT table*

Column name	Description
VERSION	<b>Data type:</b> INT; <b>Nullable:</b> No The version of DB2 DataPropagator for System i that you used to register the source.
SOURCE_OWNER	<b>Data type:</b> VARCHAR(30); <b>Nullable:</b> No The high-level qualifier of the source table or view that you registered.
SOURCE_TABLE	<b>Data type:</b> VARCHAR(128); <b>Nullable:</b> No The name of the source table or view that you registered.
SOURCE_NAME	<b>Data type:</b> CHAR(10); <b>Nullable:</b> Yes A ten-character system name of the source table or view that you used to issue the commands.
SOURCE_MBR	<b>Data type:</b> CHAR(10); <b>Nullable:</b> Yes The name of the source table member, which is used for issuing Receive Journal Entry (RCVJRNE) commands and ALIAS support.
SOURCE_TABLE_RDB	<b>Data type:</b> CHAR(18); <b>Nullable:</b> Yes When you use remote journals, this column contains the database name of the system where the source table actually resides. For local journals, this column is null.
JRN_LIB	<b>Data type:</b> CHAR(10); <b>Nullable:</b> Yes The library name of the journal that the source table uses.
JRN_NAME	<b>Data type:</b> CHAR(10); <b>Nullable:</b> Yes The name of the journal that is used by a source table. An asterisk followed by nine blanks in this column means that the source table is currently not in a journal, and it is not possible for the Capture program to capture data for this source.
FR_START_TIME	<b>Data type:</b> TIMESTAMP; <b>Nullable:</b> Yes The time when the Apply program began to perform a full refresh.

Tabella 84. Columns in the IBMSNAP\_REG\_EXT table (Continua)

Column name	Description
SOURCE_VIEW_QUAL	<p><b>Data type:</b> SMALLINT; <b>Nullable:</b> No</p> <p>Supports the view of subscriptions by matching the similar column in the register table. This value is set to equal 0 for physical tables that are defined as a source and is greater than 0 for views that are defined as sources. You must have this column to support multiple subscriptions for different source views containing identical SOURCE_OWNER and SOURCE_TABLE column values.</p>
CMT_BEHAVIOR_CASE	<p><b>Data type:</b> SMALLINT; <b>Nullable:</b> No, with default; <b>Default:</b> 0</p> <p>An integer that represents how the application programs that are updating the source table use commitment control. The Capture program uses this value to manage its memory usage for CD rows that it has constructed but is not yet ready to write to the CD tables.</p> <ul style="list-style-type: none"> <li>-1 The commitment control pattern is not yet established for the applications. This is the initial value in the column.</li> <li>0 None of the applications that update the source uses commitment control.</li> <li>1 All of the applications that update the source use commitment control. Therefore, two different applications never update the same source table under commitment control at the same time.</li> <li>2 For concurrent applications that update the source, some use commitment control and others do not. It is possible that two applications are updating the source table by using commitment control concurrently.</li> </ul>
MAX_ROWS_BTWN_CMTS	<p><b>Data type:</b> SMALLINT; <b>Nullable:</b> No, with default; <b>Default:</b> 0</p> <p>The maximum number of rows that the Capture program can process before it commits data to the CD table.</p>

## Tabella IBMSNAP\_REGISTER

La tabella IBMSNAP\_REGISTER contiene informazioni sulle origini della replica, come ad esempio i nomi delle tabelle di origine della replica, i relativi attributi e i nomi delle tabelle CD e CCD associate ad esse. Viene inserita automaticamente una riga in questa tabella ogni volta che viene definita una nuova tabella di origine della replica o una vista che il programma Capture deve elaborare.

**Server:** server di controllo Capture

**Schema predefinito:** ASN

**Indice:** SOURCE\_OWNER, SOURCE\_TABLE, SOURCE\_VIEW\_QUAL

**Importante:** Prestare attenzione quando si aggiorna questa tabella utilizzando SQL. La modifica non corretta di questa tabella può provocare risultati non previsti e perdita di dati.

La tabella di registro è il luogo che l'utente deve visualizzare nel caso in cui sia necessario sapere come ha definito le proprie origini della replica .

Tabella 85 a pagina 437 fornisce una breve descrizione delle colonne nella tabella IBMSNAP\_REGISTER.

Tabella 85. Colonne nella tabella IBMSNAP\_REGISTER

Nome colonna	Descrizione																
SOURCE_OWNER	<p><b>Tipo di dati:</b> VARCHAR(30), VARCHAR(128) per sottosistemi in modalità nuova funzione DB2 UDB per z/OS Versione 8; <b>Impostabile su null:</b> No</p> <p>Il qualificatore di alto livello della tabella di origine o vista registrata.</p>																
SOURCE_TABLE	<p><b>Tipo di dati:</b> VARCHAR(128), VARCHAR(18) per sottosistemi in modalità di compatibilità DB2 UDB per z/OS Versione 8 o precedenti; <b>Impostabile su null:</b> No</p> <p>Il nome della tabella o vista di origine registrata.</p>																
SOURCE_VIEW_QUAL	<p><b>Tipo di dati:</b> SMALLINT; <b>Impostabile su null:</b> no</p> <p>Questa colonna viene utilizzata per supportare più registrazioni per differenti viste di origine con identici valori di colonna SOURCE_OWNER e SOURCE_TABLE. Questo valore è impostato su 0 per tabelle fisiche definite come origini ed è maggiore di 0 per viste definite come origini.</p>																
GLOBAL_RECORD	<p><b>Tipo di dati:</b> CHAR(1); <b>Impostabile su null:</b> No</p>																
SOURCE_STRUCTURE	<p><b>Tipo di dati:</b> SMALLINT; <b>Impostabile su null:</b> no</p> <p>Un valore che identifica la struttura della tabella o vista di origine:</p> <table> <tr> <td>1</td> <td>Tabella utente</td> </tr> <tr> <td>3</td> <td>Tabella CCD</td> </tr> <tr> <td>4</td> <td>Tabella oraria</td> </tr> <tr> <td>5</td> <td>Tabella aggregati di base</td> </tr> <tr> <td>6</td> <td>Tabella aggregati di modifica</td> </tr> <tr> <td>7</td> <td>Tabella di replica</td> </tr> <tr> <td>8</td> <td>Tabella copia utente</td> </tr> <tr> <td>9</td> <td>Tabella CCD senza un'unione delle tabelle IBMSNAP_UOW e CD</td> </tr> </table>	1	Tabella utente	3	Tabella CCD	4	Tabella oraria	5	Tabella aggregati di base	6	Tabella aggregati di modifica	7	Tabella di replica	8	Tabella copia utente	9	Tabella CCD senza un'unione delle tabelle IBMSNAP_UOW e CD
1	Tabella utente																
3	Tabella CCD																
4	Tabella oraria																
5	Tabella aggregati di base																
6	Tabella aggregati di modifica																
7	Tabella di replica																
8	Tabella copia utente																
9	Tabella CCD senza un'unione delle tabelle IBMSNAP_UOW e CD																
SOURCE_CONDENSED	<p><b>Tipo di dati:</b> CHAR(1); <b>Impostabile su null:</b> no</p> <p>Un indicatore che segnala se la tabella di origine è una tabella concentrata, e quindi che tutte le righe con la stessa chiave sono concentrate in una riga:</p> <table> <tr> <td>Y</td> <td>L'origine è concentrata.</td> </tr> <tr> <td>N</td> <td>L'origine non è concentrata.</td> </tr> <tr> <td>A</td> <td>L'origine è una tabella aggregati di base o aggregati di modifica.</td> </tr> </table>	Y	L'origine è concentrata.	N	L'origine non è concentrata.	A	L'origine è una tabella aggregati di base o aggregati di modifica.										
Y	L'origine è concentrata.																
N	L'origine non è concentrata.																
A	L'origine è una tabella aggregati di base o aggregati di modifica.																
SOURCE_COMPLETE	<p><b>Tipo di dati:</b> CHAR(1); <b>Impostabile su null:</b> no</p> <p>Un indicatore che segnala in che modo la tabella di origine memorizza righe di valori di chiave primari:</p> <table> <tr> <td>Y</td> <td>La tabella di origine contiene una riga per ogni valore chiave primario di interesse.</td> </tr> <tr> <td>N</td> <td>La tabella di origine contiene una serie secondaria di righe di valori chiave primari.</td> </tr> </table>	Y	La tabella di origine contiene una riga per ogni valore chiave primario di interesse.	N	La tabella di origine contiene una serie secondaria di righe di valori chiave primari.												
Y	La tabella di origine contiene una riga per ogni valore chiave primario di interesse.																
N	La tabella di origine contiene una serie secondaria di righe di valori chiave primari.																

Tabella 85. Colonne nella tabella IBMSNAP\_REGISTER (Continua)

Nome colonna	Descrizione
CD_OWNER	<p><b>Tipo di dati:</b> VARCHAR(30), sottosistemi in modalità nuova funzione DB2 UDB per z/OS Versione 8; <b>Impostabile su null:</b> Sì</p> <p>Il qualificatore di alto livello della tabella CD di origine.</p> <p><b>Per tabelle come origini</b> Per tutte le tabelle di origine registrate che non sono tabelle CCD esterne, questa colonna contiene il qualificatore di alto livello della tabella CD associato a quella tabella di origine.</p> <p><b>Per viste come origini</b> Questa colonna contiene il qualificatore di alto livello della vista CD.</p> <p><b>Per tabelle CCD esterne come origini</b> Questa colonna è nulla.</p>
CD_TABLE	<p><b>Tipo di dati:</b> VARCHAR(128), VARCHAR(18) per sottosistemi in modalità compatibilità DB2 UDB per z/OS Versione 8 o precedenti; <b>Impostabile su null:</b> Sì</p> <p>Il nome della tabella CD di origine.</p> <p><b>Per tabelle come origini</b> Per tutte le tabelle di origine registrate che non sono tabelle CCD esterne, questa colonna contiene il nome della tabella CD che conserva aggiornamenti acquisiti della tabella di origine.</p> <p><b>Per viste come origini</b> Questa colonna contiene il nome della vista CD.</p> <p><b>Per tabelle CCD esterne come origini</b> Questa colonna è nulla.</p>
PHYS_CHANGE_OWNER	<p><b>Tipo di dati:</b> VARCHAR(30), VARCHAR(128) per sottosistemi in modalità nuova funzione DB2 UDB per z/OS Versione 8; <b>Impostabile su null:</b> Sì</p> <p>il qualificatore di alto livello della tabella o vista utilizzata dal programma Apply per la replica di modifica e acquisizione:</p> <p><b>Per tabelle come origini</b> Per tutte le tabelle di origine registrate che non sono tabelle CCD esterne, questa colonna contiene il qualificatore di alto livello della tabella fisica CD associata a quella tabella di origine.</p> <p><b>Per viste come origini</b> Questa colonna contiene il qualificatore di alto livello della tabella fisica CD associata a quella vista di origine.</p> <p><b>Per tabelle CCD esterne come origini</b> Questa colonna contiene il qualificatore di alto livello della tabella esterna CCD.</p>



Tabella 85. Colonne nella tabella IBMSNAP\_REGISTER (Continua)

Nome colonna	Descrizione
PHYS_CHANGE_TABLE	<p><b>Tipo di dati:</b> VARCHAR(128), VARCHAR(18) per sottosistemi in modalità compatibilità DB2 UDB per z/OS Versione 8 o precedenti; <b>Impostabile su null:</b> Sì</p> <p>Il nome della tabella o vista utilizzata dal programma Apply per la replica di modifica e acquisizione:</p> <p><b>Per tabelle come origini</b> Per tutte le tabelle di origine registrate che non sono tabelle CCD esterne, questa colonna contiene il nome della tabella CD fisica associata alla tabella di origine.</p> <p><b>Per viste come origini</b> Questa colonna contiene il nome della tabella fisica CD associata a quella vista di origine.</p> <p><b>Per tabelle CCD esterne come origini</b> Questa colonna contiene il nome della tabella CCD esterna.</p>
CD_OLD_SYNCHPOINT	<p><b>Tipo di dati:</b> CHAR(10) per dati bit; <b>Impostabile su null:</b> Sì</p> <p>Questa colonna viene utilizzata per l'handshake iniziale tra il programma Apply e il programma Capture. Il programma Capture, quindi, inizia l'acquisizione dei dati da questo numero di sequenza della registrazione nella registrazione origine. Questa colonna viene utilizzata anche per mostrare che l'eliminazione del limite di conservazione si è verificata per una tabella CD. Se questo valore è null, la registrazione è inattiva.</p>
CD_NEW_SYNCHPOINT	<p><b>Tipo di dati:</b> CHAR(10) per dati bit; <b>Impostabile su null:</b> Sì</p> <p>Il programma Capture fa avanzare questa colonna quando inserisce nuove righe nella tabella CD. IL programma Apply utilizza questa colonna per verificare se ci sono nuove modifiche che devono essere replicate.</p>
DISABLE_REFRESH	<p><b>Tipo di dati:</b> SMALLINT; <b>Impostabile su null:</b> Sì</p> <p>Un indicatore che segnala se sono consentiti aggiornamenti completi:</p> <p><b>0</b> Sono consentiti aggiornamenti completi.</p> <p><b>1</b> Sono impediti aggiornamenti completi.</p>
CCD_OWNER	<p><b>Tipo di dati:</b> VARCHAR(30), VARCHAR(128) per sottosistemi in modalità nuova funzione DB2 per z/OS Versione 8; <b>Impostabile su null:</b> Sì</p> <p>Per un'origine che ha una tabella interna CCD associata ad essa, questa colonna contiene il qualificatore di alto livello della tabella interna CCD. Per una tabella esterna CCD, questa colonna è nulla.</p>
CCD_TABLE	<p><b>Tipi di dati:</b> VARCHAR(128), VARCHAR(18) per sottosistemi in modalità compatibilità DB2 per z/OS Versione 8 o precedenti; <b>Impostabile su null:</b> Sì</p> <p>Per un'origine che ha una tabella interna CCD associata ad essa, questa colonna contiene il nome della tabella interna CCD. Per una tabella esterna CCD, questa colonna è nulla.</p>

Tabella 85. Colonne nella tabella IBMSNAP\_REGISTER (Continua)

Nome colonna	Descrizione
CCD_OLD_SYNCHPOINT	<p><b>Tipo di dati:</b> CHAR(10) per dati bit; <b>Impostabile su null:</b> Sì</p> <p>Il numero di sequenza della registrazione quando la tabella CCD viene inizializzata nuovamente. Questa colonna è correlata all'elaborazione dell'aggiornamento completo delle tabelle CCD. Il valore in questa colonna deve essere modificato solo quando la tabella CCD è inizialmente o successivamente aggiornata in modo completo. Questo valore può essere più obsoleto di ogni riga che rimane nella tabella CCD. Se questa colonna non è gestita, il programma Apply che utilizza la tabella CCD come origine della replica non sa che la tabella CCD è stata inizializzata nuovamente, quindi la nuova inizializzazione delle copie complete dell'origine CCD ha esito negativo.</p>
SYNCHPOINT	<p><b>Tipo di dati:</b> CHAR(10) per dati bit; <b>Impostabile su null:</b> Sì</p> <p>Nella riga globale (dove GLOBAL_RECORD = Y), il punto di sincronizzazione rappresenta il numero di sequenza del log dell'ultimo record di log o giornale elaborato dal programma Capture. In ogni riga della tabella IBMSNAP_REGISTER contenente informazioni di registrazione su una tabella CCD (interna o esterna), il valore del punto di sincronizzazione verrà aumentato dal programma di manutenzione della tabella CCD per indicare la presenza di nuovi dati disponibili in tale tabella CCD.</p>
SYNCHTIME	<p><b>Tipo di dati:</b> TIMESTAMP; <b>Impostabile su null:</b> Sì</p> <p>Nella riga globale (dove GLOBAL_RECORD = Y), il synchtime rappresenta la data/ora dall'ultimo record giornale o della registrazione elaborato dal programma Capture. Se il programma Capture ha raggiunto il termine della registrazione DB2, il synchtime viene aggiornato alla data/ora corrente DB2. In ogni riga nella tabella IBMSNAP_REGISTER che contiene informazioni di registrazione sulla tabella CCD (interna o esterna), il valore synchtime viene aumentato dal programma che mantiene la tabella CCD per indicare la ricorrenza dei dati disponibili in questa tabella CCD.</p>
CCD_CONDENSED	<p><b>Tipo di dati:</b> CHAR(1); <b>Impostabile su null:</b> Sì</p> <p>Un indicatore che segnala se la tabella interna CCD associata con questa origine è concentrata, e quindi tutte le righe con la stessa chiave sono concentrate in una riga:</p> <p><b>Y</b> La tabella interna CCD è concentrata.</p> <p><b>N</b> La tabella interna CCD non è concentrata.</p> <p><b>NULL</b> Nessuna tabella interna CCD è definita per questa origine.</p>
CCD_COMPLETE	<p><b>Tipo di dati:</b> CHAR(1); <b>Impostabile su null:</b> Sì</p> <p>Un indicatore che segnala se la tabella interna CCD associata con questa origine è completa, e quindi inizialmente conteneva tutte le righe dalla tabella di origine:</p> <p><b>N</b> La tabella interna CCD non è completa.</p> <p><b>NULL</b> Nessuna tabella interna CCD è definita per questa origine.</p>
ARCH_LEVEL	<p><b>Tipo di dati:</b> CHAR(4); <b>Impostabile su null:</b> No</p> <p>Il livello strutturale delle tabelle di controllo della replica:</p> <p><b>0801</b> Versione 8 Replica SQL</p> <p><b>0803</b> Versione 8 Replica SQL con supporto avanzato per origini Oracle</p> <p><b>0805</b> Versione 8 Replica SQL con supporto per modalità nuova funzione DB2 per z/OS</p>

Tabella 85. Colonne nella tabella IBMSNAP\_REGISTER (Continua)

Nome colonna	Descrizione
DESCRIPTION	<p><b>Tipo di dati:</b> CHAR(254); <b>Impostabile su null:</b> Sì</p> <p>Una descrizione dell'origine di replica.</p>
BEFORE_IMG_PREFIX	<p><b>Tipo di dati:</b> VARCHAR(4); <b>Impostabile su null:</b> Sì</p> <p>Il prefisso di un carattere che identifica i nomi delle colonne pre-immagine nella tabella CD. La combinazione del prefisso colonne pre-immagine e del nome della colonna CD non deve essere ambigua, e quindi un nome di colonna CD con prefisso non può essere uguale al nome di colonna pre-immagine corrente o potenziale. La lunghezza in byte di BEFORE_IMG_PREFIX è:</p> <p>1 Per un carattere del prefisso ASCII o EBCDIC formato da un solo byte.</p> <p>2 Per un carattere di un prefisso ASCII formato da due byte.</p> <p>4 Per un carattere del prefisso EBCDIC DBCS. Questa lunghezza è consentita per caratteri shift-in e shift-out.</p>
CONFLICT_LEVEL	<p><b>Tipo di dati:</b> CHAR(1); <b>Impostabile su null:</b> Sì</p> <p>Un indicatore che segnala il livello di rilevazione dei conflitti per questa origine:</p> <p>0 Il programma Apply non ricerca conflitti. La consistenza dei dati deve essere forzata dall'applicazione dell'utente per permettere l'aggiornamento di conflitti potenziali.</p> <p>1 Rilevazione standard con rifiuto della transazione a cascata. Il programma Apply ricerca conflitti basati sulle modifiche acquisite da questo punto. Il programma Apply invertirà ogni transazione in conflitto alla replica e anche ogni transazione con dipendenze con transazioni in conflitto. Modifiche acquisite dopo che il programma Apply inizia la rilevazione dei conflitti, non saranno verificate durante questo ciclo di Apply.</p> <p>2 Rilevazione avanzata con rifiuto della transazione a cascata. Il programma Apply attende fino a quando il programma Capture acquisisce tutte le modifiche dalla registrazione o dal giornale (fare riferimento alla descrizione della colonna SYNCHTIME), quindi effettua una rilevazione di conflitti standard, come quando è impostato su 1. Durante il tempo di attesa, il programma Apply conserva un LOCK nella tabella di origine per garantire che non vengano effettuate modifiche durante il processo di rilevazione dei conflitti.</p>
CHG_UPD_TO_DEL_INS	<p><b>Tipo di dati:</b> CHAR(1); <b>Impostabile su null:</b> Sì</p> <p>Un indicatore che segnala in che modo il programma Capture memorizza aggiornamenti nella tabella CD.</p> <p>Y Il programma Capture memorizza aggiornamenti utilizzando due righe nella tabella CD, una per l'eliminazione e una per l'inserimento. Il programma Apply elabora prima l'eliminazione e poi l'inserimento. Quando questo indicatore Y è impostato, ogni aggiornamento ad un'origine della replica viene memorizzato nella tabella CD utilizzando due righe. Questo indicatore garantisce che gli aggiornamenti, effettuati alle colonne di partizione o alle colonne a cui si riferisce un predicato della serie di richieste, siano eseguiti correttamente.</p> <p>N Ogni aggiornamento alla tabella di origine viene memorizzato in una singola riga nella tabella CD.</p>

Tabella 85. Colonne nella tabella IBMSNAP\_REGISTER (Continua)

Nome colonna	Descrizione
CHGONLY	<p><b>Tipo di dati:</b> CHAR(1); <b>Impostabile su null:</b> Sì</p> <p>Un indicatore che segnala se il programma Capture acquisisce tutte le modifiche che si verificano all'origine o solo le modifiche che si verificano nelle colonne registrate. Di solito è necessario che l'opzione sia impostata su Y per ridurre al minimo il numero di righe che il programma Capture inserisce nella tabella CD, ma è possibile che si desideri impostare questa opzione su N in modo da tenere traccia esatta di quali righe nella tabella di origine sono state aggiornate. Ad esempio, è possibile che si stiano acquisendo i valori della colonna chiave primaria per verificare quali righe sono state modificate in una tabella di origine.</p> <p><b>Y</b> Il programma Capture acquisisce solo le modifiche che si verificano in colonne registrate nella tabella di origine.</p> <p><b>N</b> Il programma Capture acquisisce le modifiche da tutte le colonne nella tabella di origine.</p>
RECAPTURE	<p><b>Tipo di dati:</b> CHAR(1); <b>Impostabile su null:</b> Sì</p> <p>Questa colonna è per la replica update-anywhere e contiene un indicatore che segnala se le modifiche che hanno origine da una tabella o una vista sono acquisite di nuovo e spedite ad altre tabelle o viste.</p> <p>Per tabelle nel sito principale:</p> <p><b>N</b> Aggiornamenti al sito principale, che sono stati applicati da una replica, non sono acquisiti di nuovo e non saranno replicati in altre repliche.</p> <p><b>Y</b> Aggiornamenti al sito principale che sono stati applicati da una replica e saranno replicati in altre repliche.</p> <p>Per tabelle nel sito di replica:</p> <p><b>Y</b> Aggiornamenti al sito di replica che sono stati applicati dal sito principale sono acquisiti di nuovo e sono disponibili per essere replicati in un'altra tabella che utilizza il sito di replica come propria origine.</p> <p><b>N</b> Aggiornamenti al sito di replica che sono stati applicati dal sito principale non sono acquisiti di nuovo.</p>
OPTION_FLAGS	<p><b>Tipo di dati:</b> CHAR(4); <b>Impostabile su null:</b> No</p> <p>Riservato per future opzioni della replica SQL. Attualmente questa colonna contiene il valore predefinito di NNNN.</p>
STOP_ON_ERROR	<p><b>Tipo di dati:</b> CHAR(1); <b>Impostabile su null:</b> Sì, con valore predefinito; <b>Valore predefinito:</b> Y.</p> <p>Un indicatore che segnala se il programma Capture terminerà o arresterà solamente l'elaborazione della registrazione se rileva errori mentre tenta di avviare, inizializzare, inizializzare di nuovo o inserire una riga nella tabella CD:</p> <p><b>Y</b> Il programma Capture termina quando si verifica un errore mentre sta tentando di avviare, inizializzare, inizializzare di nuovo o inserire una riga nella tabella CD.</p> <p><b>N</b> Il programma Capture arresta la registrazione ma non termina quando si verifica un errore mentre sta tentando di avviare, inizializzare, inizializzare di nuovo o inserire una riga nella tabella CD; continua ad elaborare altre registrazioni.</p>

Tabella 85. Colonne nella tabella IBMSNAP\_REGISTER (Continua)

Nome colonna	Descrizione
STATE	<p><b>Tipo di dati:</b> CHAR(1); <b>Impostabile su null:</b> Sì, con valore predefinito; <b>Valore predefinito:</b> I.</p> <p>Un indicatore che segnala in quale stato è la registrazione:</p> <p><b>S</b> Il programma Capture ha arrestato l'elaborazione di questa registrazione. Il programma Apply non utilizzerà questa registrazione fino a quando la registrazione non sarà corretta e inserita nello stato I (inattivo).</p> <p><b>A</b> La registrazione è attiva.</p> <p><b>I</b> La registrazione è inattiva.</p>
STATE_INFO	<p><b>Tipo di dati:</b> CHAR(8); <b>Impostabile su null:</b> Sì;</p> <p>Se il programma Capture arresta l'elaborazione della registrazione, questa colonna contiene il messaggio di errore che è stato emesso riguardo al malfunzionamento.</p>

## Tabella IBMSNAP\_REG\_SYNC (relazionale non DB2)

La tabella IBMSNAP\_REG\_SYNC utilizza un trigger di aggiornamento per avviare un aggiornamento del valore SYNCHPOINT per tutte le righe nella tabella IBMSNAP\_REGISTER quando il programma Apply si sta preparando per estrarre i dati da un'origine dati relazionale non DB2.

**Server:** server di controllo Capture

**Schema predefinito:** ASN

**Indice:** TRIGGER\_ME

**Importante:** Prestare attenzione quando si aggiorna questa tabella utilizzando SQL. La modifica non corretta di questa tabella può provocare risultati non previsti e perdita di dati.

Tabella 86 fornisce una breve descrizione delle colonne nella tabella IBMSNAP\_REG\_SYNC.

Tabella 86. Colonne della tabella IBMSNAP\_REG\_SYNC

Nome colonna	Descrizione
TRIGGER_ME	<p><b>Tipo di dati:</b> CHAR(1); <b>Impostabile su null:</b> no</p> <p>Un indicatore di Y segnala se un trigger ha iniziato l'aggiornamento del valore SYNCHPOINT per tutte le righe nella tabella di registro.</p>
TIMESTAMP	<p>Per Server SQL e origini Sybase Microsoft, questa colonna contiene il numero univoco generato dal sistema quando si verifica un aggiornamento su una colonna data/ora nella tabella. Questo valore viene utilizzato per ottenere il valore SYNCHPOINT registrato nella tabella IBMSNAP_REGISTER.</p>

## Tabella IBMSNAP\_RESTART

La tabella IBMSNAP\_RESTART contiene informazioni che abilitano il programma Capture al riavvio dal primo record giornale o di registrazione richiesto. Questa tabella sostituisce la tabella IBMSNAP\_WARM\_START dalla replica SQL Versione 7

e versioni precedenti. Essa contiene una riga, che viene aggiornata ad ogni punto di commit; quindi, il programma Capture può essere riavviato esattamente dalla giusta posizione senza acquisire di nuovo informazioni che ha già elaborato e inserito nelle tabelle CD e UOW.

**Server:** server di controllo Capture

**Schema predefinito:** ASN

**Indice:** Nessuno

**Importante:** Prestare attenzione quando si aggiorna questa tabella utilizzando SQL. La modifica non corretta di questa tabella può provocare risultati non previsti e perdita di dati. Se viene eliminata la riga da questa tabella, il programma Capture viene forzato ad eseguire l'avvio freddo.

Se non è mai stato avviato il programma Capture, questa tabella è vuota e il programma Capture deve eseguire un avvio freddo.

Le seguenti due sezioni mostrano i layout specifici del sistema operativo della tabella IBMSNAP\_RESTART.

z/OS
Linux UNIX Windows  
**z/OS, Linux, UNIX, Windows**

*Tabella 87. Colonne della tabella IBMSNAP\_RESTART per z/OS, Linux, UNIX, e Windows*

Nome colonna	Descrizione
MAX_COMMITSEQ	<p><b>Tipo di dati:</b> CHAR(10) per dati bit; <b>Impostabile su null:</b> No</p> <p>Il valore massimo del numero di sequenza della registrazione logica (IBMSNAP_COMMITSEQ) per cui il programma Capture ha eseguito il commit alle tabelle CD e UOW.</p>
MAX_COMMIT_TIME	<p><b>Tipo di dati:</b> TIMESTAMP; <b>Impostabile su null:</b> no</p> <p>La data/ora associata al numero di sequenza della registrazione nella colonna MAX_COMMITSEQ.</p>
MIN_INFLIGHTSEQ	<p><b>Tipo di dati:</b> CHAR(10) per dati bit; <b>Impostabile su null:</b> No</p> <p>Il numero di sequenza della registrazione logica con cui viene avviato il programma Capture durante un riavvio caldo. Questo valore rappresenta il primo numero di sequenza della registrazione che il programma Capture ha individuato, per il quale un commit record o un record di fine anomala non è ancora stato individuato.</p>
CURR_COMMIT_TIME	<p><b>Tipo di dati:</b> TIMESTAMP; <b>Impostabile su null:</b> no</p> <p>La data/ora locale corrente nel momento in cui questa tabella è stata aggiornata dal programma Capture.</p>
CAPTURE_FIRST_SEQ	<p><b>Tipo di dati:</b> CHAR(10) per dati bit; <b>Impostabile su null:</b> No</p> <p>Il numero di sequenza della registrazione logica associato con la registrazione di recupero che il programma Capture ha avviato dal momento dell'ultimo avvio freddo eseguito dal programma Capture. Questo valore è utilizzato per rilevare se si è verificato un database RESTORE, il quale può richiedere al programma Capture di eseguire un avvio freddo perché il programma di gestione della registrazione del database può utilizzare di nuovo i numeri di sequenza della registrazione durante alcune operazioni RESTORE.</p>

## System i

Per System i, la tabella IBMSNAP\_RESTART è utilizzata per determinare il momento di avvio del comando RCVJRNE (Receive Journal Entry). Viene inserita una riga nella tabella di riavvio per ogni giornale utilizzato dall'origine di replica o da un gruppo di origini di replica.

**Indice:** JRN\_LIB, JRN\_NAME

Tabella 88. Colonne della tabella IBMSNAP\_RESTART per System i

Nome colonna	Descrizione
MAX_COMMITSEQ	<b>Tipo di dati:</b> CHAR(10) per dati bit; <b>Impostabile su null:</b> No  Il numero del record giornale del commit più recente dalla tabella UOW.
MAX_COMMIT_TIME	<b>Tipo di dati:</b> TIMESTAMP; <b>Impostabile su null:</b> no  La data/ora associata con il numero del record giornale nella colonna MAX_COMMITSEQ, o la data/ora corrente se il programma Capture ha raggiunto le registrazioni e non ha lavoro da eseguire.
MIN_INFLIGHTSEQ	<b>Tipo di dati:</b> CHAR(10) per dati bit; <b>Impostabile su null:</b> No  Il numero di sequenza della registrazione logica da cui il programma Capture viene avviato durante un riavvio caldo.
CURR_COMMIT_TIME	<b>Tipo di dati:</b> TIMESTAMP; <b>Impostabile su null:</b> no  La data/ora corrente nel punto in cui la tabella viene aggiornata.
CAPTURE_FIRST_SEQ	<b>Tipo di dati:</b> CHAR(10) per dati bit; <b>Impostabile su null:</b> No  Il numero del record giornale che il programma Capture avvia dopo un avvio freddo.
UID	<b>Tipo di dati:</b> INTEGER; <b>Impostabile su null:</b> No  Un unico numero utilizzato come prefisso per il contenuto della colonna IBMSNAP_UOWID ubicata nella tabella UOW.
SEQNBR	<b>Tipo di dati:</b> BIGINT; <b>Impostabile su null:</b> No  Il numero di sequenza dell'ultima voce giornale che il programma Capture ha elaborato.
JRN_LIB	<b>Tipo di dati:</b> CHAR(10); <b>Impostabile su null:</b> No  Il nome della libreria del giornale che il programma Capture sta elaborando.
JRN_NAME	<b>Tipo di dati:</b> CHAR(10); <b>Impostabile su null:</b> No  Il nome del giornale che il programma Capture sta elaborando.
STATUS	<b>Tipo di dati:</b> CHAR(1); <b>Impostabile su null:</b> Sì  Un indicatore segnala se il programma Capture sta elaborando un particolare lavoro giornale:  Y      Il programma Capture sta elaborando il lavoro giornale. N      Il programma Capture non sta elaborando il lavoro giornale.

## Tabella IBMSNAP\_SEQTABLE (Informix)

La tabella IBMSNAP\_SEQTABLE contiene una sequenza di numeri univoci che la replica SQL utilizza come equivalenti di numeri di sequenza della registrazione per le tabelle Informix. Questi identificatori univoci vengono utilizzati nella tabella IBMSNAP\_REGISTER al posto dei valori dei punti di sincronizzazione affinché il programma Capture, il programma Apply e Replication Alert Monitor siano in grado di comunicare il punto in cui sono stati interrotti durante il loro ultimo ciclo.

**Server:** server di controllo Capture

**Schema predefinito:** ASN

**Indice univoco:** SEQ

**Importante:** Prestare attenzione quando si aggiorna questa tabella utilizzando SQL. La modifica non corretta di questa tabella può provocare risultati non previsti e perdita di dati.

Tabella 89 fornisce una breve descrizione della colonna nella tabella IBMSNAP\_SEQTABLE.

Tabella 89. Colonna nella tabella IBMSNAP\_SEQTABLE

Nome colonna	Descrizione
SEQ	<b>Tipo di dati:</b> INTEGER; <b>Impostabile su null:</b> No  Un numero univoco utilizzato come identificatore di log o giornale (punti di sincronizzazione) per le tabelle Informix.

## Tabella IBMSNAP\_SIGNAL

La tabella dei segnali memorizza segnali che richiedo al programma Capture di eseguire certe azioni. I segnali sono immessi sia dall'utente che dal programma Apply.

**Server:** server di controllo Capture

**Schema predefinito:** ASN

Questa tabella contiene le informazioni che si possono aggiornare utilizzando SQL.

La tabella IBMSNAP\_SIGNAL è creata con l'attributo DATA CAPTURE CHANGES, quindi tutti gli inserimenti, gli aggiornamenti e le eliminazioni di operazioni eseguiti su questa tabella sono visibili dal programma Capture come record di registrazione letti dalla registrazione di recupero DB2. Il programma Capture ignora tutti gli aggiornamenti e le eliminazioni dei record di registrazione per la tabella IBMSNAP\_SIGNAL, ma riconosce tutti i record di registrazione di inserimenti di segnale, di cui è stato eseguito il commit e che sono stati creati in modo valido, come "segnali" che richiedono la sua attenzione. Le azioni che il programma Capture esegue per un lungo record da un inserimento di segnale dipendono da cosa è specificato nella tabella IBMSNAP\_SIGNAL per quell'inserimento. I valori nella tabella IBMSNAP\_SIGNAL forniscono le istruzioni al programma Capture riguardo l'azione desiderata.



I record in questa tabella con un valore SIGNAL\_STATE C per completo o i record con una data/ora idonea per l'eliminazione del limite di conservazione vengono eliminati quando il programma Capture esegue l'eliminazione.

Tabella 90 fornisce una breve descrizione delle colonne nella tabella IBMSNAP\_SIGNAL.

Tabella 90. Colonne nella tabella IBMSNAP\_SIGNAL

Nome colonna	Descrizione
SIGNAL_TIME	<p><b>Tipo di dati:</b> TIMESTAMP; <b>Impostabile su null:</b> No, con valore predefinito; <b>valore predefinito:</b> data/ora corrente.</p> <p>La data/ora utilizzata per identificare in modo univoco la riga. Il programma Capture utilizza questo valore univoco per trovare la riga corretta nella tabella dei segnali per indicare quando ha completato l'elaborazione del segnale Capture. Questa colonna data/ora è creata come NOT NULL WITH DEFAULT, e quindi un segnale Capture può generalmente essere inserito in modo che DB2 fornisca la data/ora corrente come il valore SIGNAL_TIME.</p>
SIGNAL_TYPE	<p><b>Tipo di dati:</b> VARCHAR(30); <b>Impostabile su null:</b> No</p> <p>Un indicatore che segnala il tipo di segnale inviato:</p> <p><b>CMD</b> Un segnale inviato dall'utente, dal programma Apply o da un'altra applicazione, che è un comando di sistema o segnale. Fare riferimento alla colonna SIGNAL_SUBTYPE per questa tabella per un elenco dei sottotipi di segnali disponibili.</p> <p><b>USER</b> Un segnale inviato dall'utente o da un altro utente. Il programma Capture aggiorna il valore nella colonna SIGNAL_LSN con l'LSN dalla registrazione di quando il segnale è stato inserito e aggiorna il valore nella colonna SIGNAL_STATE da P (pending) a R (received).</p>

Tabella 90. Colonne nella tabella IBMSNAP\_SIGNAL (Continua)

Nome colonna	Descrizione
SIGNAL_SUBTYPE	<p><b>Tipo di dati:</b> VARCHAR(30); <b>Impostabile su null:</b> Sì</p> <p>L'azione eseguita dal programma Capture quando si verifica un segnale da un comando di sistema (SIGNAL_TYPE = CMD).</p> <p><b>CAPSTART</b>                      Il programma Capture avvia l'acquisizione di modifiche all'origine registrata per un particolare membro della serie di richieste, identificato da MAP_ID (dalla tabella IBMSNAP_PRUNCNTL) nella colonna SIGNAL_INPUT_IN. Ad esempio, il programma Apply immette questo segnale prima di eseguire un aggiornamento completo su tutte le tabelle di destinazione nella serie per indicare al programma che la serie è pronto per iniziare la replica di acquisizione di modifiche. Il programma Apply invia questo segnale.</p> <p><b>STOP</b>                      Il programma Capture arresta l'acquisizione di modifiche e termina. Questo comando può essere immesso solo dall'utente, non dal programma Apply.</p> <p><b>CAPSTOP</b>                      Il programma Capture arresta l'acquisizione di modifiche per una particolare origine registrata, identificata da <i>source_owner.source_table</i> nella colonna SIGNAL_INPUT_IN. Questo comando può essere immesso solo dall'utente, non dal programma Apply.</p> <p><b>UPDANY</b>                      Il programma Apply (identificato dal qualificatore Apply nella colonna SIGNAL_INPUT_IN) indica al programma Capture che sta lavorando con due programmi Capture in una configurazione update-anywhere. Il programma Apply invia questo segnale.                      Quando il tipo di segnale è USER, il sottotipo di segnale non è utilizzato o riconosciuto dal programma Capture e quindi non è un campo richiesto. È possibile impostarlo su qualsiasi valore.</p>
SIGNAL_INPUT_IN	<p><b>Tipo di dati:</b> VARCHAR(500); <b>Impostabile su null:</b> Sì</p> <p>Con il SIGNAL_TYPE=USER, questa colonna contiene l'input definito dall'utente. Se il SIGNAL_TYPE = CMD, il significato di questo valore dipende dal SIGNAL_SUBTYPE per questo segnale:</p> <p><b>CMD + CAPSTART</b>                      L'identificatore dell'associazione. Poiché i trigger Capture e non il programma Capture elaborano origini relazionali non-DB2, esiste un trigger chiamato SIGNAL_TRIGGER, che si avvia dopo che la tabella IBMSNAP_SIGNAL viene aggiornata, e aggiorna la tabella IBMSNAP_PRUNCNTL con il prossimo valore nella sequenza.</p> <p><b>CMD + UPDANY</b>                      Il qualificativo Apply che identifica il programma Apply nella configurazione update-anywhere.</p> <p><b>CMD + CAPSTOP</b>                      Il nome del proprietario dell'origine e della tabella di origine per cui il programma Capture deve arrestare l'acquisizione delle modifiche (<i>source_owner.source_table</i>).</p>

Tabella 90. Colonne nella tabella IBMSNAP\_SIGNAL (Continua)

Nome colonna	Descrizione
SIGNAL_STATE	<p><b>Tipo di dati:</b> CHAR(1); <b>Impostabile su null:</b> no</p> <p>Un indicatore che segnala lo stato del segnale:</p> <p><b>P</b> Il segnale è sospeso: il programma Capture non lo ha ancora ricevuto. Quando viene inviato un segnale, impostare il SIGNAL_STATE su P.</p> <p><b>R</b> Il programma Capture ha ricevuto il segnale. Il programma Capture imposta la serie SIGNAL_STATE su R (invece di modificarla in C per completo) quando riceve un segnale in cui SIGNAL_TYPE = USER, o uno in cui SIGNAL_TYPE = CMD e SIGNAL_SUBTYPE = STOP.</p> <p><b>C</b> Il programma Capture ha completato l'elaborazione del segnale. Il programma Capture imposta questo valore su C quando SIGNAL_TYPE = CMD per tutti i valori SIGNAL_SUBTYPE tranne STOP.</p>
SIGNAL_LSN	<p><b>Tipo di dati:</b> CHAR(10) per dati bit; <b>Impostabile su null:</b> Sì</p> <p>Il numero di sequenza della registrazione del commit record. Questo valore è impostato solo dal programma Capture.</p>

**System i** Su System i, è associata una tabella dei segnali a ogni giornale utilizzato per le tabelle di origine. Queste tabelle sono chiamate tabelle dei segnali giornale ed hanno la stessa struttura della tabella globale IBMSNAP\_SIGNAL. Il nome della tabella dei segnali giornale è *schema*.IBMSNAP\_SIGNAL\_XXXX\_yyyy, dove *xxxx* è la libreria giornale e *yyyy* è il nome giornale. Questa tabella è creata automaticamente e registrata nel giornale di origine sul server di origine.

## Tabella IBMQREP\_TABVERSION (z/OS)

La tabella ASN.IBMQREP\_TABVERSION viene creata sui sistemi operativi z/OS per consentire a Q Capture e ai programmi Capture di tenere traccia delle diverse versioni di una tabella origine. Il programma Q Capture o Capture inserisce le righe in questa tabella quando la registrazione o la sottoscrizione Q per una tabella origine viene attivata per prima e, quindi, ogni volta che la tabella di origine viene modificata.

**Server:** server Q Capture

**Schema predefinito:** ASN

**Indice univoco:** LSN, TABLEID1, TABLEID2, VERSION

**Importante:** non modificare questa tabella utilizzando SQL. La modifica non corretta di questa tabella può provocare risultati non previsti e perdita di dati.

Tabella 91 fornisce una breve descrizione delle colonne nella tabella IBMQREP\_TABVERSION.

Tabella 91. Colonne nella tabella IBMQREP\_TABVERSION

Nome colonna	Descrizione
LSN	<p><b>Tipo di dati:</b> CHAR(10) FOR BIT DATA; <b>Impostabile su null:</b> no</p> <p>Il punto nella registrazione di recupero DB2 dove il programma Q Capture o Capture ha individuato una nuova versione della tabella origine.</p>

Tabella 91. Colonne nella tabella IBMQREP\_TABVERSION (Continua)

Nome colonna	Descrizione
TABLEID1	<b>Tipo di dati:</b> SMALLINT; <b>Impostabile su null:</b> no  L'OBID (object identifier) in SYSIBM.SYSTABLES.
TABLEID2	<b>Tipo di dati:</b> SMALLINT; <b>Impostabile su null:</b> no  Il DBID (database identifier) in SYSIBM.SYSTABLES.
VERSION	<b>Tipo di dati:</b> INTEGER; <b>Impostabile su null:</b> no  Un numero generato da Q Capture o dal programma Capture per tenere traccia delle versioni diverse di una tabella origine.
SOURCE_OWNER	<b>Tipo di dati:</b> VARCHAR(128); <b>Impostabile su null:</b> no  Lo schema o il qualificatore di alto livello della tabella origine.
SOURCE_NAME	<b>Tipo di dati:</b> VARCHAR(128); <b>Impostabile su null:</b> no  Il nome della tabella origine.

## Tabella IBMSNAP\_UOW

La tabella IBMSNAP\_UOW fornisce informazioni aggiuntive sulle transazioni che sono state sottoposte a commit a una tabella di origine. Per tutti i tipi di tabelle di destinazione diversi da copia utente e CCD tipo 9, il programma Apply collega le tabelle IBMSNAP\_UOW e CD (change data) in base ai valori IBMSNAP\_COMMITSEQ corrispondenti, quando applica le modifiche alle tabelle di destinazione. Se si avvia a freddo il programma Capture, tutte queste voci in questa tabella vengono eliminate.

**Server:** server di controllo Capture

**Schema predefinito:** ASN

**Indice:** IBMSNAP\_COMMITSEQ, IBMSNAP\_LOGMARKER

**Importante:** Prestare attenzione quando si aggiorna questa tabella utilizzando SQL. La modifica non corretta di questa tabella può provocare risultati non previsti e perdita di dati.

### System i

- Poiché Capture per System i può iniziare l'acquisizione dei dati per una serie secondaria di origini della replica, non elimina tutte le righe nella tabella IBMSNAP\_UOW se si effettua un avvio freddo parziale.
- Esistono alcuni programmi dell'utente che non utilizzano il controllo di sincronizzazione. In questi casi, Capture per System i inserisce arbitrariamente una nuova riga UOW dopo che un numero di righe viene scritto nella tabella CD. Questo limite di sincronizzazione artificiale è utile per ridurre le dimensioni della tabella UOW.
- La tabella UOW è eliminata dai limiti di conservazione, non da informazioni dalla tabella IBMSNAP\_PRUNE\_SET.

Il programma Capture richiede che esista una tabella IBMSNAP\_UOW per ogni schema Capture. Il programma Capture inserisce una nuova riga in questa tabella per ogni record di registrazione o giornale che è sottoposto a commit all'origine della replica.

Il programma Capture inoltre elimina la tabella UOW basata su informazioni che il programma Apply inserisce nella tabella IBMSNAP\_PRUNE\_SET.

Tabella 92 fornisce una breve descrizione delle colonne nella tabella IBMSNAP\_UOW.

Tabella 92. Colonne nella tabella IBMSNAP\_UOW

Nome colonna	Descrizione
IBMSNAP_UOWID	<p><b>Tipo di dati:</b> CHAR(10) per dati bit; <b>Impostabile su null:</b> No</p> <p>L'identificatore UOW (unit-of-work) dall'intestazione del record della registrazione per questa unità di lavoro. È possibile decidere che questa colonna sia parte di una tabella CCD di destinazione non completa.</p>
IBMSNAP_COMMITSEQ	<p><b>Tipo di dati:</b> CHAR(10) per dati bit; <b>Impostabile su null:</b> No</p> <p>Il numero di sequenza del record di registrazione dell'istruzione commit acquisita. Per tutti i tipi di tabelle di destinazione diversi da copia utente, il programma Apply collega le tabelle UOW e CD basate sui valori in questa colonna quando applica le modifiche alla tabella di destinazione.</p>
IBMSNAP_LOGMARKER	<p><b>Tipo di dati:</b> TIMESTAMP; <b>Impostabile su null:</b> no</p> <p>L'ora (sul server di controllo Capture) del commit dei dati.</p>
IBMSNAP_AUTHTKN	<p><b>Tipo di dati:</b> VARCHAR(30); <b>Impostabile su null:</b> No</p> <p>Il token di autorizzazione associato alla transazione. Questo ID è utile per il controllo del database. Per DB2 per z/OS, questa colonna è l'ID di correlazione. Per DB2 per i5/OS, questa colonna è il nome lavoro del lavoro che ha provocato una transazione. Questa colonna non è copiata automaticamente su altre tabelle; è necessario selezionarla e copiarla come colonna di dati dell'utente. È possibile decidere che questa colonna sia parte di una tabella CCD di destinazione non completa.</p>
IBMSNAP_AUTHID	<p><b>Tipo di dati:</b> VARCHAR(30), VARCHAR(128) per sottosistemi in modalità nuova funzione DB2 UDB per z/OS Versione 8; <b>Impostabile su null:</b> No</p> <p>L'ID di autorizzazione associato alla transazione. È utile per il controllo del database. Per DB2 per z/OS, questa colonna è l'ID di autorizzazione principale. Per DB2 per i5/OS, questa colonna ha il nome dell'ID del profilo utente che ha eseguito l'applicazione che ha provocato la transazione. Questa colonna conserva l'ID di dieci caratteri riempito da spazi vuoti. Questa colonna non è copiata automaticamente su altre tabelle; è necessario selezionarla e copiarla come colonna di dati dell'utente. È possibile decidere che questa colonna sia parte di una tabella CCD di destinazione non completa.</p>

Tabella 92. Colonne nella tabella IBMSNAP\_UOW (Continua)

Nome colonna	Descrizione
IBMSNAP_REJ_CODE	<p><b>Tipo di dati:</b> CHAR(1); <b>Impostabile su null:</b> No, con valore predefinito; <b>valore predefinito:</b> 0.</p> <p>Un indicatore segnala se tutte le righe sono state rifiutate e sottoposte a rollback. Questo valore viene impostato solo durante la replica update-anywhere se la rilevazione di conflitti è specificata come standard o avanzata quando l'utente ha definito la propria origine di replica. È possibile decidere che questa colonna sia parte di una tabella CCD di destinazione non completa.</p> <p><b>0</b> Non si è verificato alcun conflitto conosciuto nella transazione.</p> <p><b>1</b> Si è verificato un conflitto perché è stata aggiornata la stessa riga nella tabella principale e nella replica. La transazione nella replica è stata rifiutata e sottoposta a rollback.</p> <p><b>2</b> La transazione è stata rifiutata e sottoposta a rollback perché era dipendente da una transazione precedente che è stata rifiutata. La transazione precedente è stata rifiutata perché è stata aggiornata la stessa riga nella tabella principale e nella replica e la transazione nella replica è stata rifiutata e sottoposta a rollback.</p> <p><b>3</b> La transazione è stata rifiutata e sottoposta a rollback perché conteneva almeno una violazione dei vincoli di integrità referenziale. Poiché questa transazione viola i vincoli referenziali definiti nella tabella di origine, il programma Apply contrassegnerà questa serie di richieste come non riuscita. Gli aggiornamenti non possono essere copiati fino a quando non vengono corrette le definizioni di integrità referenziale.</p> <p><b>4</b> La transazione è stata rifiutata e sottoposta a rollback perché era dipendente da una transazione precedente che è stata rifiutata. La transazione precedente è stata rifiutata perché conteneva almeno una violazione dei vincoli di integrità referenziale.</p>
IBMSNAP_APPLY_QUAL	<p><b>Tipo di dati:</b> CHAR(18); <b>Impostabile su null:</b> No, con valore predefinito; <b>Valore predefinito:</b> nome utente corrente.</p> <p>Il qualificativo Apply che identifica quale programma Apply applicava le modifiche. È possibile decidere che questa colonna sia parte di una tabella CCD di destinazione non completa.</p>

## Tabelle nel server di controllo Apply

Queste tabelle, memorizzate nel server di controllo Apply, contengono informazioni riguardo le proprie definizioni di sottoscrizione. Per Linux, UNIX, Windows e z/OS, è possibile creare queste tabelle di controllo secondo le proprie specifiche utilizzando il programma riga di comando ASNCLP o il Centro di replica. Per System i, queste tabelle di controllo vengono create automaticamente quando viene installato DataPropagator per System i.

Tabella 93 descrive le tabelle di controllo nel server Apply.

Tabella 93. Tabelle di controllo nel server Apply

Nome tabella	Descrizione
"Tabella ASN.IBMSNAP_APPENQ"	Utilizzato per garantire che sia in esecuzione solo un programma Apply per ogni qualificatore Apply

Tabella 93. Tabelle di controllo nel server Apply (Continua)

Nome tabella	Descrizione
<b>System i</b> Tabella ASN.IBMSNAP_APPLY_JOB (System i)	Utilizzato per garantire l'esistenza di un solo qualificatore Apply per ogni istanza del programma Apply in esecuzione su un server di controllo Apply.
"Tabella ASN.IBMSNAP_APPLYTRACE" a pagina 458	Contiene messaggi importanti dal programma Apply.
"Tabella ASN.IBMSNAP_APPLYTRAIL" a pagina 459	Contiene l'informazione del tracciato di controllo del programma Apply.
"Tabella ASN.IBMSNAP_APPPARMS" a pagina 455	Contiene parametri che è possibile modificare per controllare le operazioni del programma Apply.
"Tabella ASN.IBMSNAP_SUBS_COLS" a pagina 465	Associa le colonne nella tabella di destinazione o nella vista alle corrispondenti colonne nella tabella di origine o nella vista.
"Tabella ASN.IBMSNAP_SUBS_EVENT" a pagina 467	Contiene eventi che vengono definiti per il controllo quando il programma Apply elabora una serie di richieste.
"tabella ASN.IBMSNAP_SUBS_MEMBR" a pagina 468	Identifica una coppia di tabelle origine e destinazione e specifica l'informazione di elaborazione per quella coppia.
"tabella ASN.IBMSNAP_SUBS_SET" a pagina 472	Contiene l'informazione di elaborazione per ogni serie di membri di impostazione sottoscrizione che il programma Apply elabora come un gruppo.
"Tabella ASN.IBMSNAP_SUBS_STMTS" a pagina 479	Contiene le istruzioni SQL o i richiami del processo memorizzato che sono stati definiti per una serie di richieste. Queste vengono richiamate prima o dopo che il programma Apply elabora la serie.

## Tabella ASN.IBMSNAP\_APPENQ

La tabella di accodamento Apply è utilizzata per garantire che sia in esecuzione solo un programma Apply per ogni qualificatore Apply. Il programma Apply blocca esclusivamente una riga fino a che il programma Apply non viene terminato. Questa tabella non è utilizzata su System i.

**Server:** server di controllo Apply

**Indice:** APPLY\_QUAL

**Importante:** Prestare attenzione quando si aggiorna questa tabella utilizzando SQL. La modifica non corretta di questa tabella può provocare risultati non previsti e perdita di dati.

Tabella 94 a pagina 454 fornisce una breve descrizione della colonna nella tabella BMSNAP\_APPENQ.

Tabella 94. Colonna nella tabella IBMSNAP\_APPENQ

Nome colonna	Descrizione
APPLY_QUAL	<p><b>Tipo di dati:</b> CHAR(18); <b>Impostabile su null:</b> Sì</p> <p>Identifica in modo univoco un gruppo di serie di richieste che vengono elaborate dallo stesso programma Apply. Questo valore non è sensibile al maiuscolo/minuscolo. È necessario specificare questo valore quando si definisce una serie di richieste.</p>

## ASN.IBMSNAP\_APPLY\_JOB (System i)

### System i

The IBMSNAP\_APPLY\_JOB table, which is System i-specific, is used to guarantee a unique APPLY\_QUAL value for all instances of the Apply program running at the Apply control server. A row is added to this table every time an instance of the Apply program is started. If you start a new instance of the Apply program with an APPLY\_QUAL value that already exists, the start command fails.

**Server:** Apply control server

**Index:** None

**Important:** Use caution when you update this table by using SQL. Altering this table inappropriately can cause unexpected results and loss of data.

Tabella 95 provides a brief description of the columns in the IBMSNAP\_APPLY\_JOB table.

Tabella 95. Columns in the IBMSNAP\_APPLY\_JOB table

Column name	Description
APPLY_QUAL	<p><b>Data type:</b> CHAR(18); <b>Nullable:</b> No</p> <p>A unique identifier for a group of subscription sets. This value is supplied by the user when defining a subscription set. Each instance of the Apply program is started with an APPLY_QUAL value. This value is used during update-anywhere replication to prevent circular replication of the changes made by the Apply program.</p>
CONTROL_SERVER	<p><b>Data type:</b> CHAR(18); <b>Nullable:</b> No</p> <p>The name of the database where the Apply control tables and view are defined.</p>
JOB_NAME	<p><b>Data type:</b> CHAR(10); <b>Nullable:</b> No</p> <p>The fully qualified name of the job that wrote this trace entry:</p> <p><b>Position 1–10</b> APPLY_QUAL</p> <p><b>Position 11–20</b> The ID of the user who started the Apply program</p> <p><b>Position 21–26</b> The job number</p>
USER_NAME	<p><b>Data type:</b> CHAR(10); <b>Nullable:</b> No</p> <p>The name of the user who started a new instance of the Apply program.</p>



Tabella 95. Columns in the IBMSNAP\_APPLY\_JOB table (Continua)

Column name	Description
JOB_NUMBER	<b>Data type:</b> CHAR(6); <b>Nullable:</b> No  The job number of the current job for a particular journal. If the journal is not active, this column contains the job number of the last job that was processed.

## Tabella ASN.IBMSNAP\_APPPARMS

La tabella IBMSNAP\_APPPARMS contiene parametri che possono essere modificati per controllare le operazioni del programma Apply. È possibile definire questi parametri per impostare il valore come il nome del server di controllo Apply nel quale risiedono le definizioni di sottoscrizione e le tabelle di controllo del programma Apply. Se vengono apportate delle modifiche ai parametri in questa tabella, il programma Apply legge le modifiche solo durante l'avvio.

**Server:** server di controllo Apply

**Indice:** APPLY\_QUAL

Questa tabella contiene le informazioni che si possono aggiornare utilizzando SQL.

Tabella 96 fornisce una breve descrizione delle colonne nella tabella IBMSNAP\_APPPARMS.

Tabella 96. Colonne nella tabella IBMSNAP\_APPPARMS

Nome colonna	Descrizione
APPLY_QUAL	<b>Tipo di dati:</b> CHAR(18); <b>Impostabile su null:</b> no  Il qualificatore Apply confronta i parametri con il programma Apply al quale applica questi parametri.
APPLY_PATH	<b>Tipo di dati:</b> VARCHAR(1040); <b>Impostabile su null:</b> Sì  L'ubicazione dei file di lavoro utilizzati dal programma Apply. La directory predefinita nella quale è avviato il programma.
CAF	<b>Tipo di dati:</b> CHAR(1); <b>Impostabile su null:</b> Sì <b>Valore predefinito:</b> Y  Un indicatore che specifica se il programma Apply utilizza il CAF (Call Attach Facility) di connessione. <b>Y (valore predefinito)</b> Il programma Apply sovrascrive l'RSS (Recoverable Resource Manager Services) di connessione e viene eseguito con CAF (Call Attach Facility) di connessione. <b>N</b> Il programma Apply utilizza l'RRS (Recoverable Resource Manager Services) di connessione.

Tabella 96. Colonne nella tabella IBMSNAP\_APPPARMS (Continua)

Nome colonna	Descrizione
COPYONCE	<p><b>Tipo di dati:</b> CHAR(1); <b>Impostabile su null:</b> Sì, con valore predefinito; <b>Valore predefinito:</b> N</p> <p>Un indicatore che segnala se il programma Apply esegue un ciclo di copia per ogni serie di richieste che è idonea nel momento in cui il programma Apply viene richiamato.</p> <p>Y Il programma Apply esegue un ciclo di copia per ogni serie di richieste idonea.</p> <p>N Il programma Apply non esegue un ciclo di copia per ogni serie di richieste idonea.</p>
DELAY	<p><b>Tipo di dati:</b> INT; <b>Impostabile su null:</b> Sì, con valore predefinito; <b>Valore predefinito:</b> 6</p> <p>Il tempo di ritardo (in secondi) alla fine di ogni ciclo di Apply quando è utilizzata una replica continua. Questo parametro viene ignorato se viene specificato <b>copyonce</b>.</p>
ERRWAIT	<p><b>Tipo di dati:</b> INT; <b>Impostabile su null:</b> Sì, con valore predefinito; <b>Valore predefinito:</b> 300</p> <p>Il numero di secondi (da 1 a 300) che il programma Apply attende prima di un nuovo tentativo dopo che il programma ha rilevato una condizione di errore. Questo parametro è ignorato se è specificato <b>copyonce</b>.</p>
INAMSG	<p><b>Tipo di dati:</b> CHAR(1); <b>Impostabile su null:</b> Sì, con valore predefinito; <b>Valore predefinito:</b> Y</p> <p>Un indicatore segnala se il programma Apply immette un messaggio quando è inattivo.</p> <p>S Il programma Apply emette un messaggio quando è inattivo.</p> <p>N Il programma Apply non emette un messaggio quando è inattivo.</p>
LOADXIT	<p><b>Tipo di dati:</b> CHAR(1); <b>Impostabile su null:</b> Sì, con valore predefinito; <b>Valore predefinito:</b> N</p> <p>Un indicatore segnala se il programma Apply richiama la routine di uscita fornita da IBM (ASNLOAD) che utilizza le utilità di esportazione e caricamento per aggiornare le tabelle di destinazione.</p> <p>S Il programma Apply richiama ASNLOAD.</p> <p>N Il programma Apply non richiama ASNLOAD.</p>
LOGREUSE	<p><b>Tipo di dati:</b> CHAR(1); <b>Impostabile su null:</b> Sì, con valore predefinito; <b>Valore predefinito:</b> N</p> <p>Un indicatore che segnala se il programma Apply sovrascrive il file di log Apply o se lo aggiunge.</p> <p>Y Il programma Apply riutilizza il file di registrazione innanzitutto eliminandolo e poi ricreandolo quando il programma Apply viene riavviato.</p> <p>N Il programma Apply aggiunge una nuova informazione al file di log Apply.</p>

Tabella 96. Colonne nella tabella IBMSNAP\_APPPARMS (Continua)

Nome colonna	Descrizione
LOGSTDOUT	<p><b>Tipo di dati:</b> CHAR(1); <b>Impostabile su null:</b> Sì, con valore predefinito; <b>Valore predefinito:</b> N</p> <p>Un indicatore segnala dove il programma Apply indirizza i messaggi di registrazione:</p> <p><b>Y</b> Il programma Apply indirizza i messaggi del file di log allo standard out (STDOUT) e al file di log.</p> <p><b>N</b> Il programma Apply indirizza la maggior parte dei messaggi dei file di log solo al file di log. I messaggi di inizializzazione vengono memorizzati sia nell'out standard (STDOUT) che nel file di log.</p>
NOTIFY	<p><b>Tipo di dati:</b> CHAR(1); <b>Impostabile su null:</b> Sì, con valore predefinito; <b>Valore predefinito:</b> N</p> <p>Un indicatore che segnala se il programma Apply deve richiamare la routine d'uscita (ASNDONE) che restituisce il controllo all'utente dopo che il programma Apply finisce di copiare una serie di richieste.</p> <p><b>S</b> Il programma Apply richiama ASNDONE.</p> <p><b>N</b> Il programma Apply non richiama ASNDONE.</p>
OPT4ONE	<p><b>Tipo di dati:</b> CHAR(1); <b>Impostabile su null:</b> Sì, con valore predefinito; <b>Valore predefinito:</b> N</p> <p>Un indicatore segnala se le prestazioni del programma Apply vengono ottimizzate se è definita una sola serie di richieste per il programma Apply.</p> <p><b>Y</b> Le prestazioni del programma Apply sono ottimizzate per una serie di richieste.</p> <p><b>N</b> Le prestazioni del programma Apply non sono ottimizzate per una serie di richieste.</p> <p>Questo parametro è ignorato se è specificato <b>copyonce</b>.</p>
SLEEP	<p><b>Tipo di dati:</b> CHAR(1); <b>Impostabile su null:</b> Sì, con valore predefinito; <b>Valore predefinito:</b> Y</p> <p>Un indicatore segnala come si comporta il programma Apply se nessuna serie di richieste è idonea per l'elaborazione:</p> <p><b>S</b> Il programma Apply si disattiva.</p> <p><b>N</b> Il programma Apply si arresta.</p> <p>Questo parametro è ignorato se è specificato <b>copyonce</b>.</p>
SQLERRCONTINUE	<p><b>Tipo di dati:</b> CHAR(1); <b>Impostabile su null:</b> Sì, con valore predefinito; <b>Valore predefinito:</b> N</p> <p>Un indicatore segnala se il programma Apply continua l'elaborazione dopo aver ricercato eventuali errori nel file SQLSTATE.</p> <p><b>S</b> Il programma Apply controlla gli errori SQL nel file SQLSTATE durante l'elaborazione. Se viene trovato un errore, Apply arresta l'elaborazione.</p> <p><b>N</b> Il programma Apply non controlla il file SQLSTATE e continua l'elaborazione.</p>

Tabella 96. Colonne nella tabella IBMSNAP\_APPPARMS (Continua)

Nome colonna	Descrizione
SPILLFILE	<p><b>Tipo di dati:</b> VARCHAR(10); <b>Impostabile su null:</b> Sì, con valore predefinito.</p> <p>Un indicatore segnala dove vengono memorizzate le risposte inserite.</p> <p><b>z/OS</b> I valori validi sono:</p> <p><b>mem (predefinito)</b> Un file memoria.</p> <p><b>disk</b> Un file di disco.</p> <p><b>Linux UNIX Windows</b> I valori validi sono:</p> <p><b>disk (predefinito)</b> Un file di disco.</p>
TERM	<p><b>Tipo di dati:</b> CHAR(1); <b>Impostabile su null:</b> Sì, con valore predefinito; <b>Valore predefinito:</b> Y</p> <p>Un indicatore che segnala se il programma Apply termina se non riesce a connettersi al proprio server di controllo.</p> <p><b>Y (valore predefinito)</b> Per impostazione predefinita, il programma Apply termina se non riesce a connettersi al proprio server di controllo.</p> <p><b>N</b> Il programma Apply non termina. Apply invece registra un errore, attende per la quantità di tempo impostata dal parametro <b>errwait</b>, quindi ritenta la connessione.</p> <p>Questo parametro è ignorato se è specificato <b>copyonce</b>.</p>
TRLREUSE	<p><b>Tipo di dati:</b> CHAR(1); <b>Impostabile su null:</b> Sì, con valore predefinito; <b>Valore predefinito:</b> N</p> <p>Un indicatore segnala se il programma Apply richiama la routine di uscita fornita da IBM (ASNLOAD) che utilizza i programmi di utilità di esportazione e di caricamento per aggiornare le tabelle di destinazione:</p> <p><b>S</b> Il programma Apply richiama ASNLOAD.</p> <p><b>y</b> Il programma Apply non richiama ASNLOAD.</p>

## Tabella ASN.IBMSNAP\_APPLYTRACE

La tabella traccia IBMSNAP\_APPLYTRACE contiene i messaggi del programma Apply. Il programma Apply non elimina automaticamente questa tabella, ma è possibile automatizzare l'eliminazione aggiungendo un'istruzione SQL che si avvia dopo una serie di richieste.

**Server:** server di controllo Apply

**Indice:** APPLY\_QUAL, TRACE\_TIME

Tabella 97 a pagina 459 fornisce una breve descrizione della colonna nella tabella IBMSNAP\_APPLYTRACE.

Tabella 97. Colonne nella tabella IBMSNAP\_APPLYTRACE

Nome colonna	Descrizione
APPLY_QUAL	<b>Tipo di dati:</b> CHAR(18); <b>Impostabile su null:</b> no  Identifica in modo univoco quale programma Apply ha inserito il messaggio.
TRACE_TIME	<b>Tipo di dati:</b> TIMESTAMP; <b>Impostabile su null:</b> no  L'ore del server di controllo Apply quando la riga è stata inserita in questa tabella.
OPERATION	<b>Tipo di dati:</b> CHAR(8); <b>Impostabile su null:</b> No  Il tipo di operazione del programma Apply, ad esempio, inizializzazione, applicazione, o una condizione di errore.
DESCRIPTION	<b>Tipo di dati:</b> VARCHAR(1024); <b>Impostabile su null:</b> No  L'ID del messaggio seguito dal testo del messaggio. L'ID del messaggio è rappresentato dai primi sette caratteri della colonna DESCRIPTION. Il testo del messaggio inizia dalla nona posizione della colonna DESCRIPTION.

## Tabella ASN.IBMSNAP\_APPLYTRAIL

La tabella IBMSNAP\_APPLYTRAIL contiene le informazioni del tracciato di controllo di tutti i cicli d'impostazione sottoscrizione eseguiti dal programma Apply. Questa tabella registra una cronologia degli aggiornamenti che vengono eseguiti rispetto alle richieste. È un repository di statistiche di diagnostica e prestazioni. La tabella tracciato di Apply è uno degli elementi più importanti da controllare se s'incontrano problemi con il programma Apply. Il programma Apply non elimina automaticamente questa tabella, ma è possibile automatizzare facilmente l'eliminazione aggiungendo una successiva istruzione SQL ad una delle serie di richieste.

**Server:** server di controllo Apply

**Indice:** LASTRUN, APPLY\_QUAL

Tabella 98 fornisce una breve descrizione della colonna nella tabella IBMSNAP\_APPLYTRAIL.

Tabella 98. Le colonne nella tabella IBMSNAP\_APPLYTRAIL

Nome colonna	Descrizione
APPLY_QUAL	<b>Tipo di dati:</b> CHAR(18); <b>Impostabile su null:</b> no  Identifica in modo univoco quale programma stava elaborando la serie di richieste.
SET_NAME	<b>Tipo di dati:</b> CHAR(18); <b>Impostabile su null:</b> no  Il nome della serie di richieste che il programma Apply stava elaborando.
SET_TYPE	<b>Tipo di dati:</b> CHAR(1); <b>Impostabile su null:</b> no  Il valore che appare nella colonna SET_TYPE della tabella IBMSNAP_SUBS_SET dopo il più recente ciclo di Apply.

Tabella 98. Le colonne nella tabella IBMSNAP\_APPLYTRAIL (Continua)

Nome colonna	Descrizione
WHOS_ON_FIRST	<p><b>Tipo di dati:</b> CHAR(1); <b>Impostabile su null:</b> no</p> <p>I valori seguenti sono utilizzati per controllare l'ordine di elaborazione negli scenari di replica update-anywhere.</p> <p><b>F</b> (primo) La tabella di origine è la replica e la tabella di destinazione è la principale. In caso di conflitti di aggiornamento tra la tabella di replica e la tabella principale, le transazioni in conflitto della replica saranno rifiutate. F non è utilizzato per richieste di sola lettura; è utilizzato per update-anywhere.</p> <p><b>S</b> (secondo) La tabella di origine è la tabella principale o un'altra origine e la tabella di destinazione è la replica o un'altra copia. In caso di conflitti di aggiornamento tra la tabella principale e la tabella di replica, le transazioni in conflitto della replica saranno rifiutate. S è utilizzato per tutte le richieste di sola lettura.</p>
ASNLOAD	<p><b>Tipo di dati:</b> CHAR(1); <b>Impostabile su null:</b> Sì</p> <p>Il valore utilizzato per avviare il programma Apply:</p> <p><b>S</b> Indica che il programma Apply è stato avviato con il parametro <b>loadxit=y</b> provocando l'aggiornamento completo su una serie di richieste da parte della routine di uscita utente ASNLOAD.</p> <p><b>N</b> Indica che la routine di uscita ASNLOAD non è stata richiamata poiché non è stato necessario un aggiornamento completo oppure il programma Apply non è stato avviato con il parametro <b>loadxit</b>.</p> <p><b>NULL</b> Indica che si è verificato un errore del programma Apply prima che il programma Apply potesse determinare la necessità di richiamare la routine di uscita ASNLOAD.</p>
FULL_REFRESH	<p><b>Tipo di dati:</b> CHAR(1); <b>Impostabile su null:</b> Sì</p> <p>Un indicatore segnala se si è verificato un aggiornamento completo:</p> <p><b>Y</b> Indica che è stato eseguito un aggiornamento completo ad una serie di richieste.</p> <p><b>N</b> Indica che non è stato eseguito un aggiornamento completo ad una serie di richieste.</p> <p><b>NULL</b> Indica che si è verificato un errore prima che il programma Apply potesse determinare la necessità di un aggiornamento completo.</p>
EFFECTIVE_MEMBERS	<p><b>Tipo di dati:</b> INT; <b>Impostabile su null:</b> Sì</p> <p>Il numero di membri della serie di richieste modificati durante un ciclo di Apply da un aggiornamento completo o da una replica di inserimento, aggiornamento, ed eliminazione. Questo numero è compreso fra zero e il numero di membri della serie di richieste.</p>
SET_INSERTED	<p><b>Tipo di dati:</b> INT; <b>Impostabile su null:</b> No</p> <p>Il numero totale di righe inserite nei membri della serie di richieste durante il ciclo di sottoscrizione.</p>
SET_DELETED	<p><b>Tipo di dati:</b> INT; <b>Impostabile su null:</b> No</p> <p>Il numero totale di righe eliminate fra i membri della serie di richieste durante il ciclo di sottoscrizione.</p>

Tabella 98. Le colonne nella tabella IBMSNAP\_APPLYTRAIL (Continua)

Nome colonna	Descrizione
SET_UPDATED	<p><b>Tipo di dati:</b> INT; <b>Impostabile su null:</b> No</p> <p>Il numero totale di righe aggiornate nei membri della serie di richieste durante il ciclo di sottoscrizione.</p>
SET_REWORKED	<p><b>Tipo di dati:</b> INT; <b>Impostabile su null:</b> No</p> <p>Il numero totale di righe che il programma Apply ha rielaborato durante l'ultimo ciclo. Il programma Apply rielabora le modifiche secondo le seguenti condizioni:</p> <ul style="list-style-type: none"><li>• Se un inserimento ha esito negativo poiché le righe sono già esistenti nella tabella di destinazione, il programma Apply converte gli inserimenti in un aggiornamento della riga esistente.</li><li>• Se l'aggiornamento ha esito negativo poiché le righe non esistono nella tabella di destinazione, il programma Apply converte gli aggiornamenti in un inserimento.</li></ul>
SET_REJECTED_TRXS	<p><b>Tipo di dati:</b> INT; <b>Impostabile su null:</b> No</p> <p>Il numero totale delle transazioni che sono state rifiutate a causa di un conflitto update-anywhere. Questa colonna è utilizzata solamente per serie di richieste update-anywhere nelle quali la rilevazione del conflitto è definita come standard o avanzato.</p>

Tabella 98. Le colonne nella tabella IBMSNAP\_APPLYTRAIL (Continua)

Nome colonna	Descrizione
STATUS	<p><b>Tipo di dati:</b> SMALLINT; <b>Impostabile su null:</b> no</p> <p>Un valore che rappresenta lo stato di lavoro per il programma Apply dopo un determinato ciclo:</p> <p><b>-1</b> La replica non è riuscita. Il programma Apply ha perso l'intera serie di righe applicate in precedenza e non è stato eseguito il commit di alcun dato. Se il parametro di avvio SQLERRCONTINUE = Y, SQLSTATE restituito al programma Apply durante l'ultimo ciclo <i>non</i> è uno degli errori accettabili segnalati nel file di input per SQLERRCONTINUE (<i>apply_qualifier.SQS</i>).</p> <p><b>0</b> Il programma Apply ha elaborato la serie di richieste con esito positivo. Se il parametro di avvio SQLERRCONTINUE = Y, il programma Apply non rileva alcun errore SQL segnalato per il parametro di avvio SQLERRCONTINUE (in <i>apply_qualifier.SQS</i>) e non rifiuta alcuna riga.</p> <p><b>2</b> Il programma Apply sta elaborando la serie di richieste in più cicli. Ha elaborato con esito positivo una sottoscrizione logica singola che è stata divisa in base alla colonna di controllo MAX_SYNC_MINUTES.</p> <p><b>16</b> Il programma Apply ha elaborato con esito la serie di richieste e ha restituito uno stato 0; tuttavia, ha rilevato alcuni errori SQL che erano stati segnalati per il parametro di avvio SQLERRCONTINUE (in <i>apply_qualifier.SQS</i>) e ha rifiutato alcune righe. Fare riferimento al file <i>apply_qualifier.ERR</i> per dettagli sulle righe che hanno avuto esito negativo.</p> <p><b>Esempio:</b> Viene selezionato SQLERRCONTINUE = Y e viene indicato che lo stato SQL consentito è 23502 (codice SQL -407). Si verifica un errore 23502, ma non si verifica nessun altro errore. Il programma Apply termina l'elaborazione della serie di richieste e imposta lo stato a 16. Durante la successiva esecuzione, si verifica un errore 23502, ma anche un errore 07006 (codice SQL -301). Ora il programma Apply arresta l'elaborazione della serie di richieste, perde tutte l'intera serie di righe applicate e imposta lo stato a -1 (poiché non è stato eseguito il commit di alcun dato).</p> <p><b>18</b> Il programma Apply sta elaborando una serie di richieste in più cicli e restituisce uno stato con valore 2, che indica che il programma ha elaborato con successo una sottoscrizione logica singola che è stata divisa in base alla colonna di controllo MAX_SYNC_MINUTES. Tuttavia, rileva alcuni errori SQL che erano stati segnalati per il parametro di avvio SQLERRCONTINUE (in <i>apply_qualifier.SQS</i>) e ha rifiutato alcune righe. Fare riferimento al file <i>apply_qualifier.ERR</i> per dettagli sulle righe che hanno avuto esito negativo.</p>
LASTRUN	<p><b>Tipo di dati:</b> TIMESTAMP; <b>Impostabile su null:</b> no</p> <p>Il tempo stimato in cui è iniziata l'ultima sottoscrizione. Il programma Apply imposta il valore LASTRUN ogni volta che viene elaborata una serie di richieste. È l'ora approssimativa del server di controllo Apply alla quale il programma Apply inizia ad elaborare la serie di richieste.</p>
LASTSUCCESS	<p><b>Tipo di dati:</b> TIMESTAMP; <b>Impostabile su null:</b> Sì</p> <p>La data/ora del server di controllo Apply per l'inizio dell'ultima elaborazione con esito positivo di una serie di richieste.</p>



Tabella 98. Le colonne nella tabella IBMSNAP\_APPLYTRAIL (Continua)

Nome colonna	Descrizione
SYNCHPOINT	<p><b>Tipo di dati:</b> CHAR(10) per dati bit; <b>Impostabile su null:</b> Sì</p> <p>Il programma Apply utilizza questa colonna per registrare il proprio avanzamento, indicando di aver provveduto all'elaborazione dei dati fino a tale valore di punto di sincronizzazione per quanto riguarda la serie di sottoscrizioni.</p>
SYNCHTIME	<p><b>Tipo di dati:</b> TIMESTAMP; <b>Impostabile su null:</b> Sì</p> <p>IL programma Apply utilizza questa colonna per registrare il proprio avanzamento, segnalando che ha elaborato i dati fino a questa data/ora per la serie di richieste.</p>
SOURCE_SERVER	<p><b>Tipo di dati:</b> CHAR(18); <b>Impostabile su null:</b> no</p> <p>Il nome del database DB2 dove vengono definite le tabelle di origine e le viste.</p>
SOURCE_ALIAS	<p><b>Tipo di dati:</b> CHAR(8); <b>Impostabile su null:</b> Sì</p> <p>L'alias DB2 che corrisponde al server d'origine indicato nella colonna SOURCE_SERVER.</p>
SOURCE_OWNER	<p><b>Tipo di dati:</b> VARCHAR(30), VARCHAR(128) per sottosistemi in modalità nuova funzione DB2 UDB per z/OS Versione 8; <b>Impostabile su null:</b> Sì</p> <p>Il qualificatore di altro livello della tabella o della vista di origine che il programma Apply stava elaborando. Questo valore è impostato solamente quando il ciclo Apply ha esito negativo.</p>
SOURCE_TABLE	<p><b>Tipo di dati:</b> VARCHAR(128), VARCHAR(18) per sottosistemi in modalità di compatibilità DB2 UDB z/OS Versione 8 o precedenti; <b>Impostabile su null:</b> Sì</p> <p>Il nome della tabella o della vista di origine che il programma Apply stava elaborando. Questo valore è impostato solamente quando il ciclo Apply ha esito negativo.</p>
SOURCE_VIEW_QUAL	<p><b>Tipo di dati:</b> SMALLINT; <b>Impostabile su null:</b> Sì</p> <p>Il valore del qualificatore della vista d'origine per la tabella o per la vista di origine che il programma Apply stava elaborando. Questo valore è impostato solamente quando il ciclo Apply ha esito negativo.</p>
TARGET_SERVER	<p><b>Tipo di dati:</b> CHAR(18); <b>Impostabile su null:</b> no</p> <p>Il nome database del server in cui le tabelle o viste di destinazione sono memorizzate.</p>
TARGET_ALIAS	<p><b>Tipo di dati:</b> CHAR(8); <b>Impostabile su null:</b> Sì</p> <p>L'alias DB2 corrispondente al server di destinazione denominato nella colonna TARGET_SERVER.</p>
TARGET_OWNER	<p><b>Tipo di dati:</b> VARCHAR(30), VARCHAR(128) per sottosistemi in modalità nuova funzione DB2 UDB per z/OS Versione 8; <b>Impostabile su null:</b> No</p> <p>Il qualificatore di alto livello della tabella destinazione che il programma Apply stava processando. Questo valore è impostato solamente quando il ciclo Apply ha esito negativo.</p>
TARGET_TABLE	<p><b>Tipo di dati:</b> VARCHAR(128), VARCHAR(18) per sottosistemi in modalità di compatibilità DB2 UDB per z/OS Versione 8 o precedenti; <b>Impostabile su null:</b> No</p> <p>In nome della tabella di destinazione che il programma Apply stava processando. Questo valore è impostato solamente quando il ciclo Apply ha esito negativo.</p>

Tabella 98. Le colonne nella tabella IBMSNAP\_APPLYTRAIL (Continua)

Nome colonna	Descrizione
CAPTURE_SCHEMA	<p><b>Tipo di dati:</b> VARCHAR(30), VARCHAR(128) per sottosistemi in modalità nuova funzione DB2 UDB per z/OS Versione 8; <b>Impostabile su null:</b> No</p> <p>Il nome dello schema delle tabelle del server Capture per questa serie di richieste.</p>
TGT_CAPTURE_SCHEMA	<p><b>Tipo di dati:</b> VARCHAR(30), VARCHAR(128) per sottosistemi in modalità nuova funzione DB2 UDB per z/OS Versione 8; <b>Impostabile su null:</b> Sì</p> <p>Se la tabella destinazione è anche l'origine di un'altra serie di richieste (come una tabella CCD in una configurazione multi-livello o in una tabella di replica in una configurazione update-anywhere), questa colonna contiene lo schema Capture che sarà utilizzato quando la tabella si comporta come un'origine.</p>
FEDERATED_SRC_SRVR	<p><b>Tipo di dati:</b> VARCHAR(18); <b>Impostabile su null:</b> Sì</p> <p>Il nome del server remoto federato che è l'origine per la serie di richieste, che si applica solo a origini relazionali non-DB2.</p>
FEDERATED_TGT_SRVR	<p><b>Tipo di dati:</b> VARCHAR(18); <b>Impostabile su null:</b> Sì</p> <p>Il nome del server remoto federato che è la destinazione della serie di richieste, il quale si applica solo ai server di destinazione relazionali non-DB2.</p>
JRN_LIB	<p><b>Tipo di dati:</b> CHAR(10); <b>Impostabile su null:</b> Sì</p> <p><b>System i</b> Questa colonna, che si applica solamente ai server Capture System i, è il nome della libreria di giornale utilizzata dalla tabella di origine.</p>
JRN_NAME	<p><b>Tipo di dati:</b> CHAR(10); <b>Impostabile su null:</b> Sì</p> <p><b>System i</b> Questa colonna, che si applica solamente ai server Capture System i, è il nome del giornale utilizzato da una tabella origine. Un asterisco seguito da nove spazi bianchi in questa colonna indica che la colonna di origine attualmente non è in un giornale, in questo caso non è possibile acquisire dati per questa tabella di origine.</p>
COMMIT_COUNT	<p><b>Tipo di dati:</b> SMALLINT; <b>Impostabile su null:</b> Sì</p> <p>Il valore di COMMIT_COUNT dall'ultimo ciclo Apply, registrato nella tabella IBMSNAP_SUBS_SET.</p>
OPTION_FLAGS	<p><b>Tipo di dati:</b> CHAR(4); <b>Impostabile su null:</b> No</p> <p>Riservato per future opzioni della replica SQL. Attualmente questa colonna contiene il valore predefinito di NNNN.</p>
EVENT_NAME	<p><b>Tipo di dati:</b> CHAR(18); <b>Impostabile su null:</b> Sì</p> <p>Un carattere univoco utilizzato per rappresentare l'evento che ha attivato la serie da elaborare.</p>
ENDTIME	<p><b>Tipo di dati:</b> TIMESTAMP; <b>Impostabile su null:</b> No, con valore predefinito; <b>valore predefinito:</b> data/ora corrente.</p> <p>La data/ora del server di controllo Apply quando il programma Apply ha terminato di elaborare la serie di richieste. Per individuare quanto tempo è durata l'elaborazione, sottrae LASTRUN da ENDTIME.</p>
SOURCE_CONN_TIME	<p><b>Tipo di dati:</b> TIMESTAMP; <b>Impostabile su null:</b> Sì</p> <p>La data/ora del server di controllo Capture quando il programma Apply esegue per la prima volta il collegamento per inserire i dati di origine.</p>

Tabella 98. Le colonne nella tabella IBMSNAP\_APPLYTRAIL (Continua)

Nome colonna	Descrizione
SQLSTATE	<b>Tipo di dati:</b> CHAR(5); <b>Impostabile su null:</b> Sì Il codice di stato SQL per una esecuzione con esito negativo. Altrimenti, NULL.
SQLCODE	<b>Tipo di dati:</b> INT; <b>Impostabile su null:</b> Sì Il codice di errore SQL per una esecuzione con esito negativo. Altrimenti, NULL.
SQLERRP	<b>Tipo di dati:</b> CHAR(8); <b>Impostabile su null:</b> Sì L'identificatore del prodotto database del server nel quale si è verificato l'errore SQL che ha provocato l'esito negativo dell'esecuzione. Altrimenti, NULL.
SQLERRM	<b>Tipo di dati:</b> VARCHAR(70); <b>Impostabile su null:</b> Sì L'informazione di errore SQL per una esecuzione con esito negativo.
APPERRM	<b>Tipo di dati:</b> VARCHAR(760); <b>Impostabile su null:</b> Sì L'ID del messaggio di errore Apply e il testo per una esecuzione con esito negativo.

## Tabella ASN.IBMSNAP\_SUBS\_COLS

La tabella IBMSNAP\_SUBS\_COLS contiene informazioni sulle colonne dei membri della serie di richieste copiate in una serie di richieste. Le righe vengono inserite automaticamente in questa tabella o eliminate da questa tabella quando le informazioni vengono modificate in una o più colonne di una coppia di tabelle di origine e destinazione. Utilizzare questa tabella se sono necessarie informazioni riguardo colonne specifiche in un membro della serie di richieste.

**Server:** server di controllo Apply

**Indice:** APPLY\_QUAL, SET\_NAME, WHOS\_ON\_FIRST, TARGET\_OWNER, TARGET\_TABLE, TARGET\_NAME

**Importante:** Prestare attenzione quando si aggiorna questa tabella utilizzando SQL. La modifica non corretta di questa tabella può provocare risultati non previsti e perdita di dati.

Tabella 99 fornisce una breve descrizione delle colonne nella tabella IBMSNAP\_SUBS\_COLS.

Tabella 99. Colonne nella tabella IBMSNAP\_SUBS\_COLS

Nome colonna	Descrizione
APPLY_QUAL	<b>Tipo di dati:</b> CHAR(18); <b>Impostabile su null:</b> no Identifica in modo univoco quale programma Apply elabora questo membro della serie di richieste.
SET_NAME	<b>Tipo di dati:</b> CHAR(18); <b>Impostabile su null:</b> no Il nome di una serie di richieste a cui appartiene questo membro.

Tabella 99. Colonne nella tabella IBMSNAP\_SUBS\_COLS (Continua)

Nome colonna	Descrizione
WHOS_ON_FIRST	<p><b>Tipo di dati:</b> CHAR(1); <b>Impostabile su null:</b> no</p> <p>I valori seguenti sono utilizzati per controllare l'ordine di elaborazione negli scenari di replica update-anywhere.</p> <p><b>F</b> (primo) La tabella di origine è la replica e la tabella di destinazione è la principale. In caso di conflitti di aggiornamento tra la tabella di replica e la tabella principale, le transazioni in conflitto della replica saranno rifiutate. F non è utilizzato per richieste di sola lettura; è utilizzato per update-anywhere.</p> <p><b>S</b> (secondo) La tabella di origine è la tabella principale o un'altra origine e la tabella di destinazione è la replica o un'altra copia. In caso di conflitti di aggiornamento tra la tabella principale e la tabella di replica, le transazioni in conflitto della replica saranno rifiutate. S è utilizzato per tutte le richieste di sola lettura.</p>
TARGET_OWNER	<p><b>Tipo di dati:</b> VARCHAR(30), VARCHAR(128) per sottosistemi in modalità nuova funzione DB2 UDB per z/OS Versione 8; <b>Impostabile su null:</b> No</p> <p>Il qualificatore di alto livello per una tabella o vista di destinazione.</p>
TARGET_TABLE	<p><b>Tipo di dati:</b> VARCHAR(128); VARCHAR(18) per sottosistemi in modalità compatibilità DB2 UDB per z/OS Versione 8 o precedenti; <b>Impostabile su null:</b> No</p> <p>La tabella o vista a cui i dati sono stati applicati.</p>
COL_TYPE	<p><b>Tipo di dati:</b> CHAR(1); <b>Impostabile su null:</b> no</p> <p>Un indicatore che segnala il tipo di colonna.</p> <p><b>A</b> Una colonna post-immagine.</p> <p><b>B</b> Una colonna pre-immagine.</p> <p><b>C</b> Una colonna calcolata o un'espressione SQL che utilizza le funzioni scalari.</p> <p><b>F</b> Una colonna calcolata che utilizza funzioni di colonne.</p> <p><b>L</b> Un valore di indicatore LOB.</p> <p><b>P</b> Una colonna predicato pre-immagine.</p> <p><b>R</b> Una colonna RRN (relative record number), fornita dal sistema e utilizzata come colonna chiave primaria. Utilizzata solo da DB2 DataPropagator per System i.</p>
TARGET_NAME	<p><b>Tipo di dati:</b> VARCHAR(30); <b>Impostabile su null:</b> No</p> <p>Il nome della colonna della tabella o vista di destinazione. Non è necessario che corrisponda al nome della colonna di origine.</p> <p>I nomi delle colonne CCD interne non possono essere rinominati. Devono corrispondere ai nomi della colonna della tabella di origine.</p>
IS_KEY	<p><b>Tipo di dati:</b> CHAR(1); <b>Impostabile su null:</b> no</p> <p>Un indicatore che segnala se la colonna è parte della chiave di destinazione, che può essere sia un indice univoco che una chiave primaria di una tabella di destinazione concentrata:</p> <p><b>Y</b> La colonna è tutta o parte della chiave di destinazione.</p> <p><b>N</b> La colonna non è parte della chiave di destinazione.</p>

Tabella 99. Colonne nella tabella IBMSNAP\_SUBS\_COLS (Continua)

Nome colonna	Descrizione
COLNO	<b>Tipo di dati:</b> SMALLINT; <b>Impostabile su null:</b> no  L'ubicazione numerica della colonna nell'origine, da conservare vicino ad altre colonne utente in visualizzazioni e richieste.
EXPRESSION	<b>Tipo di dati:</b> VARCHAR(254); <b>Impostabile su null:</b> No  Il nome della colonna di origine o un'espressione SQL utilizzata per creare il contenuto della colonna di destinazione.

## Tabella ASN.IBMSNAP\_SUBS\_EVENT

La tabella IBMSNAP\_SUBS\_EVENT contiene informazioni riguardo i trigger degli eventi associati con una serie di richieste. Contiene inoltre nomi e data/ora associati con i nomi degli eventi.

**Server:** server di controllo Apply

**Indice:** EVENT\_NAME, EVENT\_TIME

Questa tabella contiene le informazioni che si possono aggiornare utilizzando SQL.

Si inserisce una riga in questa tabella quando si crea un nuovo evento per avviare un programma Apply.

Tabella 100 fornisce una breve descrizione delle colonne nella tabella IBMSNAP\_SUBS\_EVENT.

Tabella 100. Colonne nella tabella IBMSNAP\_SUBS\_EVENT

Nome colonna	Descrizione
EVENT_NAME	<b>Tipo di dati:</b> CHAR(18); <b>Impostabile su null:</b> no  L'identificatore univoco di un evento. Questo identificatore viene utilizzato per attivare la replica per una serie di richieste.
EVENT_TIME	<b>Tipo di dati:</b> TIMESTAMP; <b>Impostabile su null:</b> no  Una data/ora del server di controllo Apply di un tempo di registrazione corrente o futuro. Le applicazioni dell'utente agli eventi di replica di segnale forniscono i valori in questa colonna.
END_SYNCHPOINT	<b>Tipo di dati:</b> CHAR(10) per dati bit; <b>Impostabile su null:</b> Sì  Un numero di sequenza di registrazione che indica al programma Apply di applicare solo i dati che sono stati acquisiti fino a questo punto. È possibile trovare l'esatto END_SYNCHPOINT che si desidera utilizzare facendo riferimento alla tabella dei segnali e trovando il numero di sequenza di registrazione preciso associato a una data/ora. Tutte le transazioni che sono sottoposte a commit dopo questo punto nella registrazione non sono replicate fino a quando un evento successivo viene inviato. Se vengono forniti valori per END_SYNCHPOINT e END_OF_PERIOD, il programma Apply utilizza il valore END_SYNCHPOINT perché questo poi non deve eseguire alcun calcolo dalle tabelle di controllo per trovare il numero di sequenza di registrazione massimo da replicare.

Tabella 100. Colonne nella tabella IBMSNAP\_SUBS\_EVENT (Continua)

Nome colonna	Descrizione
END_OF_PERIOD	<p><b>Tipo di dati:</b> TIMESTAMP; <b>Impostabile su null:</b> Sì</p> <p>Una data/ora utilizzata dal programma Apply, che applica solo i dati che sono stati registrati fino a questo punto. Tutte le transazioni che sono sottoposte a commit dopo questo punto nella registrazione non sono replicate fino a quando un evento successivo viene inviato.</p>

## tabella ASN.IBMSNAP\_SUBS\_MEMBR

La tabella IBMSNAP\_SUBS\_MEMBR contiene informazioni riguardo le coppie di tabelle di origine e destinazione individuali definite per una serie di richieste. Una riga singola viene inserita automaticamente in questa tabella quando viene aggiunto un membro della serie di richieste. Utilizzare questa tabella per identificare una specifica coppia di tabelle di origine e destinazione all'interno di una serie di richieste.

**Server:** server di controllo Apply

**Indice:** APPLY\_QUAL, SET\_NAME, WHOS\_ON\_FIRST, SOURCE\_OWNER, SOURCE\_TARGET, SOURCE\_VIEW\_QUAL, TARGET\_OWNER, TARGET\_TABLE

**Importante:** Prestare attenzione quando si aggiorna questa tabella utilizzando SQL. La modifica non corretta di questa tabella può provocare risultati non previsti e perdita di dati.

Tabella 101 fornisce una breve descrizione delle colonne nella tabella IBMSNAP\_SUBS\_MEMBR.

Tabella 101. Colonne nella tabella IBMSNAP\_SUBS\_MEMBR

Nome colonna	Descrizione
APPLY_QUAL	<p><b>Tipo di dati:</b> CHAR(18); <b>Impostabile su null:</b> no</p> <p>Identifica in modo univoco quale programma Apply elabora questo membro della serie di richieste.</p>
SET_NAME	<p><b>Tipo di dati:</b> CHAR(18); <b>Impostabile su null:</b> no</p> <p>Il nome della serie di richieste a cui appartiene questo membro.</p>
WHOS_ON_FIRST	<p><b>Tipo di dati:</b> CHAR(1); <b>Impostabile su null:</b> no</p> <p>I valori seguenti sono utilizzati per controllare l'ordine di elaborazione negli scenari di replica update-anywhere.</p> <p><b>F</b> (primo) La tabella di origine è la replica e la tabella di destinazione è la principale. In caso di conflitti di aggiornamento tra la tabella di replica e la tabella principale, le transazioni in conflitto della replica saranno rifiutate. F non è utilizzato per richieste di sola lettura; è utilizzato per update-anywhere.</p> <p><b>S</b> (secondo) La tabella di origine è la tabella principale o un'altra origine e la tabella di destinazione è la replica o un'altra copia. In caso di conflitti di aggiornamento tra la tabella principale e la tabella di replica, le transazioni in conflitto della replica saranno rifiutate. S è utilizzato per tutte le richieste di sola lettura.</p>

Tabella 101. Colonne nella tabella IBMSNAP\_SUBS\_MEMBR (Continua)

Nome colonna	Descrizione
SOURCE_OWNER	<p><b>Tipo di dati:</b> VARCHAR(30); VARCHAR(128) per sottosistemi in modalità nuova funzione DB2 UDB per z/OS Versione 8; <b>Impostabile su null:</b> No</p> <p>Il qualificatore di alto livello per la tabella o vista di origine per questo membro.</p>
SOURCE_TABLE	<p><b>Tipo di dati:</b> VARCHAR(128); VARCHAR(18) per sottosistemi in modalità compatibilità DB2 UDB per z/OS Versione 8 o precedenti; <b>Impostabile su null:</b> No</p> <p>Il nome della tabella o vista di origine per questo membro.</p>
SOURCE_VIEW_QUAL	<p><b>Tipo di dati:</b> SMALLINT; <b>Impostabile su null:</b> no</p> <p>Supporta la visualizzazione di tabelle fisiche facendo corrispondere le colonne simili nella tabella IBMSNAP_REGISTER. Questo valore è impostato su 0 per tabelle fisiche definite come origini ed è maggiore di 0 per viste definite come origini. Questa colonna viene utilizzata per supportare più richieste per viste di origine differenti con valori delle colonne SOURCE_OWNER e SOURCE_TABLE identici.</p>
TARGET_OWNER	<p><b>Tipo di dati:</b> VARCHAR(30), VARCHAR(128) per sottosistemi in modalità nuova funzione DB2 UDB per z/OS Versione 8; <b>Impostabile su null:</b> No</p> <p>Il qualificatore di alto livello per la tabella o vista di destinazione per questo membro.</p>
TARGET_TABLE	<p><b>Tipo di dati:</b> VARCHAR(128), VARCHAR(18) per sottosistemi in modalità di compatibilità DB2 UDB z/OS Versione 8 o precedenti <b>Impostabile su null:</b> No</p> <p>Il nome della tabella o vista di destinazione per questo membro.</p>
TARGET_CONDENSED	<p><b>Tipo di dati:</b> CHAR(1); <b>Impostabile su null:</b> no</p> <p>Un indicatore che segnala:</p> <p><b>Y</b> Per ogni determinato valore chiave primario, la tabella di destinazione visualizza solo una riga.</p> <p><b>N</b> Tutte le modifiche devono rimanere per conservare una cronologia degli aggiornamenti completa.</p> <p><b>A</b> La tabella di destinazione è una tabella aggregati di base o aggregati di modifica.</p>
TARGET_COMPLETE	<p><b>Tipo di dati:</b> CHAR(1); <b>Impostabile su null:</b> no</p> <p>Un indicatore che segnala:</p> <p><b>Y</b> La tabella di destinazione contiene una riga per ogni valore chiave primario di interesse.</p> <p><b>N</b> La tabella di destinazione contiene alcune serie secondarie di righe di valori chiave primari.</p>

Tabella 101. Colonne nella tabella IBMSNAP\_SUBS\_MEMBER (Continua)

Nome colonna	Descrizione																
TARGET_STRUCTURE	<p><b>Tipo di dati:</b> SMALLINT; <b>Impostabile su null:</b> no</p> <p>La struttura della tabella di destinazione:</p> <table> <tr> <td>1</td> <td>Tabella utente</td> </tr> <tr> <td>3</td> <td>Tabella CCD</td> </tr> <tr> <td>4</td> <td>Tabella oraria</td> </tr> <tr> <td>5</td> <td>Tabella aggregati di base</td> </tr> <tr> <td>6</td> <td>Tabella aggregati di modifica</td> </tr> <tr> <td>7</td> <td>Replica</td> </tr> <tr> <td>8</td> <td>Copia dell'utente</td> </tr> <tr> <td>9</td> <td>Tabella CCD senza un'unione delle tabelle IBMSNAP_UOW e CD</td> </tr> </table>	1	Tabella utente	3	Tabella CCD	4	Tabella oraria	5	Tabella aggregati di base	6	Tabella aggregati di modifica	7	Replica	8	Copia dell'utente	9	Tabella CCD senza un'unione delle tabelle IBMSNAP_UOW e CD
1	Tabella utente																
3	Tabella CCD																
4	Tabella oraria																
5	Tabella aggregati di base																
6	Tabella aggregati di modifica																
7	Replica																
8	Copia dell'utente																
9	Tabella CCD senza un'unione delle tabelle IBMSNAP_UOW e CD																
PREDICATES	<p><b>Tipo di dati:</b> VARCHAR(1024); <b>Impostabile su null:</b> Sì.</p> <p>Elenca i predicati che devono essere posti in una clausola WHERE per la tabella nella colonna TARGET_TABLE. Questa clausola WHERE crea una serie secondaria di righe della tabella di origine. I predicati sono riconosciuti solo quando WHOS_ON_FIRST è impostato su S. Il predicato non può contenere una clausola ORDER BY perché il programma Apply non può generare una clausola ORDER BY. Le tabelle aggregati richiedono un predicato fittizio seguito da una clausola GROUP BY.</p> <p>Poiché il programma Apply utilizza questi predicati sia per la replica di aggiornamento completo che per la replica di acquisizione delle modifiche, questa colonna non può contenere predicati che coinvolgono colonne nella tabella CD o UOW. I predicati che contengono riferimenti della tabella CD o UOW sono memorizzati nella colonna in the UOW_CD_PREDICATES.</p>																
MEMBER_STATE	<p><b>Tipo di dati:</b> CHAR(1); <b>Impostabile su null:</b> Sì</p> <p>Un indicatore che segnala in quale stato è il membro:</p> <table> <tr> <td><b>N</b></td> <td>Nuovo (New) Il membro è nuovo per questa serie di richieste. Inoltre, tutti i membri che sono stati abilitati recentemente appariranno in questo stato.</td> </tr> <tr> <td><b>L</b></td> <td>Caricato (loaded) I membri di questa serie di richieste sono stati caricati, ma non sono ancora stati un ciclo di modifica.</td> </tr> <tr> <td><b>S</b></td> <td>Sincronizzato (synchronized) Il membro è stato fatto progredire dallo stato nuovo (N) allo stato caricato (L) e ora è sincronizzato con tutti gli altri membri della serie di richieste che sono in uno stato sincronizzato. Quando tutti i membri di una serie di richieste sono nello stato sincronizzato, possono verificarsi repliche di modifica a livello della serie di richieste.</td> </tr> <tr> <td><b>D</b></td> <td>Disabilitato (disabled) Il membro è disabilitato per questa serie di richieste.</td> </tr> </table>	<b>N</b>	Nuovo (New) Il membro è nuovo per questa serie di richieste. Inoltre, tutti i membri che sono stati abilitati recentemente appariranno in questo stato.	<b>L</b>	Caricato (loaded) I membri di questa serie di richieste sono stati caricati, ma non sono ancora stati un ciclo di modifica.	<b>S</b>	Sincronizzato (synchronized) Il membro è stato fatto progredire dallo stato nuovo (N) allo stato caricato (L) e ora è sincronizzato con tutti gli altri membri della serie di richieste che sono in uno stato sincronizzato. Quando tutti i membri di una serie di richieste sono nello stato sincronizzato, possono verificarsi repliche di modifica a livello della serie di richieste.	<b>D</b>	Disabilitato (disabled) Il membro è disabilitato per questa serie di richieste.								
<b>N</b>	Nuovo (New) Il membro è nuovo per questa serie di richieste. Inoltre, tutti i membri che sono stati abilitati recentemente appariranno in questo stato.																
<b>L</b>	Caricato (loaded) I membri di questa serie di richieste sono stati caricati, ma non sono ancora stati un ciclo di modifica.																
<b>S</b>	Sincronizzato (synchronized) Il membro è stato fatto progredire dallo stato nuovo (N) allo stato caricato (L) e ora è sincronizzato con tutti gli altri membri della serie di richieste che sono in uno stato sincronizzato. Quando tutti i membri di una serie di richieste sono nello stato sincronizzato, possono verificarsi repliche di modifica a livello della serie di richieste.																
<b>D</b>	Disabilitato (disabled) Il membro è disabilitato per questa serie di richieste.																



Tabella 101. Colonne nella tabella IBMSNAP\_SUBS\_MEMBR (Continua)

Nome colonna	Descrizione
TARGET_KEY_CHG	<p><b>Tipo di dati:</b> CHAR(1); <b>Impostabile su null:</b> no</p> <p>Un indicatore segnala come il programma Apply gestisce gli aggiornamenti quando, nella tabella di origine, vengono modificate le colonne di origine con le colonne chiave di destinazione della tabella di destinazione:</p> <p><b>Y</b> Il programma Apply aggiorna la tabella di destinazione basata sulle pre-immagini della colonna chiave di destinazione, ciò significa che il programma Apply modifica il predicato nei valori vecchi invece che nei nuovi. Verificare di aver registrato ogni colonna di pre-immagine della chiave di destinazione, in modo che sia presente nella tabella CD. Per la voce di registrazione corrispondente nella tabella di registro, verificare che il valore nella colonna CHG_UPD_TO_DEL_INS sia impostato su N.</p> <p><b>N</b> Il programma Apply, durante l'elaborazione degli aggiornamenti e delle eliminazioni, utilizza la logica che suppone che le colonne che contengono la chiave di destinazione non vengano mai aggiornate.</p>
UOW_CD_PREDICATES	<p><b>Tipo di dati:</b> VARCHAR(1024); <b>Impostabile su null:</b> Sì.</p> <p>Contiene i predicati che includono colonne dalla tabella CD o UOW necessari al programma Apply solo per la replica di acquisizione delle modifiche, e non per aggiornamenti completi. Durante la replica di acquisizione delle modifiche, il programma Apply elabora predicati in questa colonna e altri nella colonna PREDICATES. Durante un aggiornamento completo, il programma Apply elabora solo i predicati nella colonna PREDICATES.</p>
JOIN_UOW_CD	<p><b>Tipo di dati:</b> CHAR(1); <b>Impostabile su null:</b> Sì</p> <p>Un indicatore segnala se il programma Apply effettua un'unione delle tabelle CD e UOW quando elabora una tabella di destinazione copia utente. È necessario un indicatore quando si definisce un membro della serie di richieste con predicati che utilizzano colonne dalla tabella UOW che non sono nella tabella CD. Se la tabella di destinazione è di tipo copia utente, il programma Apply utilizza un'unione delle tabelle CD e UOW quando elabora il membro e ignora questa colonna quando elabora il membro.</p> <p><b>Y</b> Il programma Apply utilizza un'unione delle tabelle CD e UOW quando elabora il membro.</p> <p><b>N</b> Il programma Apply non utilizza un'unione delle tabelle CD e UOW quando elabora un membro; legge le modifiche solo dalla tabella CD.</p> <p><b>NULL</b> Il programma Apply ignora questa colonna quando elabora il membro. Se la tabella di destinazione è una copia utente e il valore in questa colonna è null, il programma Apply non effettua un'unione delle tabelle CD e UOW quando elabora il membro.</p>

Tabella 101. Colonne nella tabella IBMSNAP\_SUBS\_MEMBR (Continua)

Nome colonna	Descrizione
LOADX_TYPE	<p><b>Tipo di dati:</b> SMALLINT; <b>Impostabile su null:</b> Sì</p> <p>Il tipo di caricamento per questo membro. Il valore in questa colonna viene utilizzato per sovrascrivere i valori predefiniti.</p> <p>NULL</p> <p><b>z/OS</b> La funzione LOAD FROM CURSOR (disponibile con l'Utilities Suite DB2) viene utilizzata per questo membro.</p> <p><b>Linux UNIX Windows</b> La chiusura ASNLOAD determina l'utilità più appropriata per questo membro (opzione 3, 4, o 5).</p> <p>1 ASNLOAD non è utilizzata per questo membro. In questo caso viene disattivata realmente l'opzione ASNLOAD per un particolare membro della serie di richieste anche se è stato specificato LOADX all'avvio.</p> <p>2 Viene utilizzato un codice di chiusura ASNLOAD definito dall'utente o modificato dall'utente.</p> <p>3 La funzione LOAD FROM CURSOR viene utilizzata per questo membro.</p> <p><b>Linux UNIX Windows</b> 4 EXPORT e LOAD sono utilizzate per questo membro.</p> <p><b>Linux UNIX Windows</b> 5 EXPORT e IMPORT sono utilizzate per questo membro.</p> <p><b>Limitazione:</b> <b>Linux UNIX Windows</b> L'utilità LOAD non è supportata per tabelle raggruppate per intervalli. Per effettuare un aggiornamento completo di una tabella raggruppata per intervalli, è possibile utilizzare sia l'utilità IMPORT DB2 o il programma Apply per eseguire un aggiornamento completo della tabella attraverso SQL.</p>
LOADX_SRC_N_OWNER	<p><b>Tipo di dati:</b> VARCHAR(30); <b>Impostabile su null:</b> Sì</p> <p>Proprietario del nickname creato dall'utente. Questo valore è richiesto quando esistono tutte le seguenti condizioni:</p> <ul style="list-style-type: none"> <li>• La funzione LOAD FROM CURSOR viene utilizzata per questo membro (LOADX_TYPE è 3)</li> <li>• Il server di destinazione è Linux, UNIX o Windows</li> <li>• L'origine non è un nickname</li> </ul>
LOADX_SRC_N_TABLE	<p><b>Tipo di dati:</b> VARCHAR(128); <b>Impostabile su null:</b> Sì.</p> <p>La tabella de nickname creata dall'utente. Questo valore è richiesto quando esistono tutte le seguenti condizioni:</p> <ul style="list-style-type: none"> <li>• La funzione LOAD FROM CURSOR viene utilizzata per questo membro (LOADX_TYPE è 3)</li> <li>• Il server di destinazione è Linux, UNIX o Windows</li> <li>• L'origine non è un nickname</li> </ul>

## tabella ASN.IBMSNAP\_SUBS\_SET

La tabella IBMSNAP\_SUBS\_SET elenca tutte le serie di richieste definite sul server di controllo Apply e documenta l'avanzamento della replica per queste serie. Le righe vengono inserite in questa tabella quando viene creata la propria definizione della serie di richieste.

**Server:** server di controllo Apply

**Indice:** APPLY\_QUAL, SET\_NAME, WHOS\_ON\_FIRST

**Importante:** Prestare attenzione quando si aggiorna questa tabella utilizzando SQL. La modifica non corretta di questa tabella può provocare risultati non previsti e perdita di dati.

Tabella 102 fornisce una breve descrizione delle colonne nella tabella IBMSNAP\_SUBS\_SET.

Tabella 102. Colonne nella tabella IBMSNAP\_SUBS\_SET

Nome colonna	Descrizione
APPLY_QUAL	<b>Tipo di dati:</b> CHAR(18); <b>Impostabile su null:</b> no Identifica in modo univoco quale programma Apply elabora questa serie di richieste.
SET_NAME	<b>Tipo di dati:</b> CHAR(18); <b>Impostabile su null:</b> no Il nome della serie di richieste.
SET_TYPE	<b>Tipo di dati:</b> CHAR(1); <b>Impostabile su null:</b> no Un indicatore segnala se la serie è di sola lettura o lettura/scrittura: <b>R</b> La serie è di sola lettura. <b>U</b> La serie è una configurazione update-anywhere e, quindi, è lettura/scrittura.
WHOS_ON_FIRST	<b>Tipo di dati:</b> CHAR(1); <b>Impostabile su null:</b> no I valori seguenti sono utilizzati per controllare l'ordine di elaborazione negli scenari di replica update-anywhere. <b>F</b> (primo) La tabella di origine è la replica e la tabella di destinazione è la principale. In caso di conflitti di aggiornamento tra la tabella di replica e la tabella principale, le transazioni in conflitto della replica saranno rifiutate. F non è utilizzato per richieste di sola lettura; è utilizzato per update-anywhere. <b>S</b> (secondo) La tabella di origine è la tabella principale o un'altra origine e la tabella di destinazione è la replica o un'altra copia. In caso di conflitti di aggiornamento tra la tabella principale e la tabella di replica, le transazioni in conflitto della replica saranno rifiutate. S è utilizzato per tutte le richieste di sola lettura.
ACTIVATE	<b>Tipo di dati:</b> SMALLINT; <b>Impostabile su null:</b> no Un indicatore segnala se il programma Apply elaborerà la serie durante il prossimo ciclo: <b>0</b> La serie di richieste è disattivata. Il programma Apply non elaborerà la serie. <b>1</b> La serie di richieste è attiva all'infinito. Il programma Apply elaborerà la serie durante ogni ciclo Apply fino a quando non viene disattivata la serie o fino a quando il programma Apply non è in grado di elaborarla. <b>2</b> La serie di richieste è attiva solo per un ciclo Apply. Il programma Apply elaborerà la serie una volta e successivamente la disattiverà.

Tabella 102. Colonne nella tabella IBMSNAP\_SUBS\_SET (Continua)

Nome colonna	Descrizione
SOURCE_SERVER	<p><b>Tipo di dati:</b> CHAR(18); <b>Impostabile su null:</b> no</p> <p>Il nome database del server di controllo Capture in cui le tabelle e viste di origine sono definite.</p>
SOURCE_ALIAS	<p><b>Tipo di dati:</b> CHAR(8); <b>Impostabile su null:</b> Sì</p> <p>L'alias DB2 corrispondente al server di controllo Capture denominato nella colonna SOURCE_SERVER.</p>
TARGET_SERVER	<p><b>Tipo di dati:</b> CHAR(18); <b>Impostabile su null:</b> no</p> <p>Il nome database del server in cui le tabelle o viste di destinazione sono memorizzate.</p>
TARGET_ALIAS	<p><b>Tipo di dati:</b> CHAR(8); <b>Impostabile su null:</b> Sì</p> <p>L'alias DB2 corrispondente al server di destinazione denominato nella colonna TARGET_SERVER.</p>

Tabella 102. Colonne nella tabella IBMSNAP\_SUBS\_SET (Continua)

Nome colonna	Descrizione
STATUS	<p><b>Tipo di dati:</b> SMALLINT; <b>Impostabile su null:</b> no</p> <p>Un valore che rappresenta lo stato di lavoro per il programma Apply dopo un determinato ciclo:</p> <p><b>-1</b> La replica non è riuscita. Il programma Apply ha ritirato l'intera serie di righe applicate e nessun dato è stato sottoposto a commit. Se il parametro di avvio è SQLERRCONTINUE = Y, l'SQLSTATE restituito al programma Apply durante l'ultimo ciclo non è uno degli errori accettabili che sono stati indicati nel file di input per SQLERRCONTINUE (<i>apply_qualifier.SQS</i>).</p> <p><b>0</b> Il programma Apply ha elaborato la serie di richieste con esito positivo. Se il parametro di avvio è SQLERRCONTINUE = Y, il programma Apply non ha rilevato alcun errore SQL fra quelli indicati per il parametro di avvio SQLERRCONTINUE (in <i>apply_qualifier.SQS</i>) e non ha rifiutato alcuna riga.</p> <p><b>1</b> Il programma Apply sta elaborando la serie di richieste.</p> <p><b>2</b> Il programma Apply sta elaborando la serie di richieste in più cicli. Ha elaborato con esito positivo una sottoscrizione logica singola che è stata divisa in base alla colonna di controllo MAX_SYNC_MINUTES.</p> <p><b>16</b> Il programma Apply ha elaborato con esito la serie di richieste e ha restituito uno stato 0; tuttavia, ha rilevato alcuni errori SQL che erano stati segnalati per il parametro di avvio SQLERRCONTINUE (in <i>apply_qualifier.SQS</i>) e ha rifiutato alcune righe. Fare riferimento al file <i>apply_qualifier.ERR</i> per dettagli sulle righe che hanno avuto esito negativo.</p> <p><b>Esempio:</b> impostare SQLERRCONTINUE = Y e indicare che lo stato SQL disponibile è 23502 (SQL code -407). Si verifica un errore 23502, ma non si verificano altri errori. Il programma Apply termina l'elaborazione della serie di richieste e imposta lo stato su 16. Nella prossima esecuzione, si verifica un errore 23502, ma successivamente si verifica un errore 07006 (SQL code -301). Ora il programma Apply arresta l'elaborazione della serie di richieste, ritira l'intera serie di righe applicate e imposta lo stato su -1 (perché nessun dato è stato sottoposto a commit).</p> <p><b>18</b> Il programma Apply sta elaborando la serie di richieste in più cicli e restituisce uno stato 2, quindi ha elaborato con esito positivo una sottoscrizione logica singola che è stata divisa in base alla colonna di controllo MAX_SYNC_MINUTES. Tuttavia, rileva alcuni errori SQL che erano stati segnalati per il parametro di avvio SQLERRCONTINUE (in <i>apply_qualifier.SQS</i>) e ha rifiutato alcune righe. Fare riferimento al file <i>apply_qualifier.ERR</i> per dettagli sulle righe che hanno avuto esito negativo.</p>
LASTRUN	<p><b>Tipo di dati:</b> TIMESTAMP; <b>Impostabile su null:</b> no</p> <p>Il momento stimato in cui è iniziata l'ultima serie di richieste. Il programma Apply imposta il valore LASTRUN ogni volta che viene elaborata una serie di richieste. È il momento approssimativo sul server di controllo Apply in cui il programma Apply inizia ad elaborare la serie di richieste.</p>

Tabella 102. Colonne nella tabella IBMSNAP\_SUBS\_SET (Continua)

Nome colonna	Descrizione
REFRESH_TYPE	<p><b>Tipo di dati:</b> CHAR(1); <b>Impostabile su null:</b> no</p> <p>Il tipo di pianificazione utilizzata per richiedere al programma Apply di elaborare questa serie di richieste:</p> <p><b>R</b> Il programma Apply utilizza la pianificazione in base all'ora. Utilizza il valore in SLEEP_MINUTES per determinare quando avviare l'elaborazione della serie di richieste.</p> <p><b>E</b> Il programma Apply utilizza la pianificazione basata sugli eventi. Verifica il valore di tempo nella tabella IBMSNAP_SUBS_EVENT per determinare quando avviare l'elaborazione della serie di richieste. Prima che ogni replica (acquisizione di modifiche o aggiornamento completo) possa iniziare, deve verificarsi un evento.</p> <p><b>B</b> Il programma Apply utilizza sia la pianificazione in base all'ora che quella basata sugli eventi. Quindi, elabora la serie di richieste basata sia sul tempo che sul criterio di eventi.</p>
SLEEP_MINUTES	<p><b>Tipo di dati:</b> INT; <b>Impostabile su null:</b> Sì</p> <p>Specifica il tempo (in minuti) di inattività tra le elaborazioni di serie di richieste. Il tempo di elaborazione viene utilizzato solo quando REFRESH_TYPE è R o B. Se il valore di SLEEP_MINUTES è NULL, il programma Apply elaborerà la serie a ciclo continuo. Il programma Apply elaborerà la serie più frequentemente possibile, ma elaborerà anche tutte le altre serie di richieste attive con lo stesso qualificatore Apply.</p>
EVENT_NAME	<p><b>Tipo di dati:</b> CHAR(18); <b>Impostabile su null:</b> Sì</p> <p>Una stringa a carattere univoco utilizzata per rappresentare il nome di un evento. Utilizzare questo identificatore per aggiornare la tabella eventi di sottoscrizione quando si desidera attivare la replica per una serie di richieste. Il nome evento viene utilizzato solo quando REFRESH_TYPE è E o B.</p>
LASTSUCCESS	<p><b>Tipo di dati:</b> TIMESTAMP; <b>Impostabile su null:</b> Sì</p> <p>La data/ora del server di controllo Apply per l'inizio dell'ultima elaborazione con esito positivo di una serie di richieste.</p>
SYNCHPOINT	<p><b>Tipo di dati:</b> CHAR(10) per dati bit; <b>Impostabile su null:</b> Sì</p> <p>Il programma Apply utilizza questa colonna per registrare il proprio avanzamento, segnalando che ha elaborato i dati fino a questo valore synchpoint per la serie di richieste.</p>
SYNCHTIME	<p><b>Tipo di dati:</b> TIMESTAMP; <b>Impostabile su null:</b> Sì</p> <p>Il programma Apply utilizza questa colonna per registrare il proprio avanzamento, segnalando che ha elaborato i dati fino a questa data/ora per la serie di richieste.</p>
CAPTURE_SCHEMA	<p><b>Tipo di dati:</b> VARCHAR(30), VARCHAR(128) per sottosistemi in modalità nuova funzione DB2 UDB per z/OS Versione 8; <b>Impostabile su null:</b> No</p> <p>Il nome dello schema delle tabelle di controllo Capture che elabora l'origine per questa serie di richieste.</p>

Tabella 102. Colonne nella tabella IBMSNAP\_SUBS\_SET (Continua)

Nome colonna	Descrizione
TGT_CAPTURE_SCHEMA	<p><b>Tipo di dati:</b> VARCHAR(30), VARCHAR(128) per sottosistemi in modalità nuova funzione DB2 UDB per z/OS Versione 8; <b>Impostabile su null:</b> Sì</p> <p>Se la tabella di destinazione è anche l'origine per altre serie di richieste (come ad esempio una tabella CCD esterna in una configurazione a più livelli o una tabella di replica in una configurazione update-anywhere), questa colonna contiene lo schema Capture utilizzato quando la tabella agisce come origine.</p>
FEDERATED_SRC_SRVR	<p><b>Tipo di dati:</b> VARCHAR(18); <b>Impostabile su null:</b> Sì</p> <p>Il nome del server remoto federato che è l'origine per la serie di richieste, che si applica solo a origini relazionali non-DB2.</p>
FEDERATED_TGT_SRVR	<p><b>Tipo di dati:</b> VARCHAR(18); <b>Impostabile su null:</b> Sì</p> <p>Il nome del server remoto federato che è la destinazione per la serie di richieste, che si applica solo a destinazioni relazionali non-DB2.</p>
JRN_LIB	<p><b>Tipo di dati:</b> CHAR(10); <b>Impostabile su null:</b> Sì</p> <p><b>System i</b> Questa colonna, che si applica solo sui server Capture System i, è il nome della libreria del giornale utilizzato dalla tabella di origine.</p>
JRN_NAME	<p><b>Tipo di dati:</b> CHAR(10); <b>Impostabile su null:</b> Sì</p> <p><b>System i</b> Questa colonna, che si applica solo sui server Capture System i, è il nome del giornale utilizzato da una tabella di origine. Un asterisco seguito da nove spazi bianchi in questa colonna indica che la colonna di origine attualmente non è in un giornale, in questo caso non è possibile acquisire dati per questa tabella di origine.</p>
OPTION_FLAGS	<p><b>Tipo di dati:</b> CHAR(4); <b>Impostabile su null:</b> No</p> <p>Riservato per future opzioni della replica SQL. Attualmente questa colonna contiene il valore predefinito di NNNN.</p>

Tabella 102. Colonne nella tabella IBMSNAP\_SUBS\_SET (Continua)

Nome colonna	Descrizione
COMMIT_COUNT	<p><b>Tipo di dati:</b> SMALLINT; <b>Impostabile su null:</b> Sì</p> <p>Un indicatore che segnala il tipo di elaborazione che il programma Apply esegue per una serie di richieste:</p> <p><b>NULL</b> Questo è il valore predefinito per una serie di richieste di sola lettura. Il programma Apply elaborerà serie di risposte estratte per i membri della serie di richieste <i>n</i>, un membro alla volta, fino a quando tutti i dati sono stati elaborati e quindi inoltrerà un commit singolo alla fine dell'elaborazione dei dati per l'intera serie. Il vantaggio di utilizzare questa impostazione COMMIT_COUNT è che l'elaborazione potrebbe essere completato più velocemente.</p> <p><i>Numero intero non NULL</i></p> <p>Il programma Apply elabora la serie di richieste in modalità transazionale. Dopo che tutte le serie di risposte vengono estratte, il contenuto dei file di trasferimento sarà applicato nell'ordine della sequenza di commit, ordinando ogni transazione con l'ordine del valore IBMSNAP_INTENTSEQ. Questo tipo di elaborazione permette di aprire ed elaborare tutti i file di trasferimento nello stesso momento. Un commit sarà emesso seguendo il numero di transazioni specificato in questa colonna. Ad esempio, 1 significa un commit dopo ogni transazione, 2 significa un commit dopo ogni due transazioni, e così via. Un numero intero 0 significa che un singolo commit sarà emesso dopo che tutti i dati estratti vengono applicati. Il vantaggio di utilizzare l'elaborazione con modalità transazionale è che l'elaborazione consente vincoli di integrità referenziale alla destinazione e possono essere emessi commit provvisori.</p>
MAX_SYNC_MINUTES	<p><b>Tipo di dati:</b> SMALLINT; <b>Impostabile su null:</b> Sì</p> <p>Un limite della soglia di tempo per regolare la quantità di dati di modifica per estrarli ed applicarli durante un ciclo di richieste. Il programma Apply suddivide la serie di richieste elaborandola in piccoli cicli basati sulla colonna IBMSNAP_LOGMARKER nella tabella UOW o CCD sul server Capture ed effettua un COMMIT sul server di destinazione dopo ogni piccolo ciclo con esito positivo. Il limite viene ricalcolato automaticamente se il programma Apply rileva un vincolo di risorse che rende il limite della serie inattuabile. I valori MAX_SYNC_MINUTES che sono meno di 1 verranno considerati come un valore MAX_SYNC_MINUTES uguale a null.</p>
AUX_STMTS	<p><b>Tipo di dati:</b> SMALLINT; <b>Impostabile su null:</b> no</p> <p>Il numero delle richieste SQL definite nella tabella IBMSNAP_SUBS_STMTS che possono essere in esecuzione prima o dopo che il programma Apply elabori una serie di richieste.</p>
ARCH_LEVEL	<p><b>Tipo di dati:</b> CHAR(4); <b>Impostabile su null:</b> No</p> <p>Il livello strutturale delle tabelle di controllo della replica. Questa colonna identifica le regole per cui viene creata una riga. Questo livello è definito da IBM.</p> <p><b>0801</b> Versione 8 o successiva Replica SQL</p> <p><b>0803</b> Versione 8 Replica SQL con supporto avanzato per origini Oracle</p> <p><b>0805</b> Versione 8 Replica SQL con supporto per modalità nuova funzione DB2 per z/OS</p>



## Tabella ASN.IBMSNAP\_SUBS\_STMTS

La tabella IBMSNAP\_SUBS\_STMTS contiene le istruzioni SQL definite dall'utente o richiami della procedura memorizzata che saranno eseguiti prima o dopo il ciclo di elaborazione della serie di istruzioni. Istruzioni di Esecuzione Immediata (EI) o procedure memorizzate possono essere eseguite solo sul server di origine o di destinazione. Questa tabella viene popolata quando viene definita una serie di istruzioni che utilizza istruzioni SQL o richiami della procedura memorizzata.

**Server:** server di controllo Apply

**Indice:** APPLY\_QUAL, SET\_NAME, WHOS\_ON\_FIRST, BEFORE\_OR\_AFTER, STMT\_NUMBER

**Importante:** Prestare attenzione quando si aggiorna questa tabella utilizzando SQL. La modifica non corretta di questa tabella può provocare risultati non previsti e perdita di dati. Il numero di voci per una sottoscrizione deve essere riflessa nella colonna AUX\_STMTS della tabella IBMSNAP\_SUBS\_SET. Se AUX\_STMTS è zero per una serie di richieste, le voci corrispondenti nella tabella IBMSNAP\_SUBS\_STMTS vengono ignorate dal programma Apply.

Tabella 103 fornisce una breve descrizione delle colonne nella tabella IBMSNAP\_SUBS\_STMTS.

Tabella 103. Colonne nella tabella IBMSNAP\_SUBS\_STMTS

Nome colonna	Descrizione
APPLY_QUAL	<b>Tipo di dati:</b> CHAR(18); <b>Impostabile su null:</b> no Identifica in modo univoco quale programma Apply elabora questa istruzione SQL o procedura memorizzata.
SET_NAME	<b>Tipo di dati:</b> CHAR(18); <b>Impostabile su null:</b> no IL nome della serie di richieste con cui è associata l'istruzione SQL o la procedura memorizzata.
WHOS_ON_FIRST	<b>Tipo di dati:</b> CHAR(1); <b>Impostabile su null:</b> no I valori seguenti sono utilizzati per controllare l'ordine di elaborazione negli scenari di replica update-anywhere. <b>F</b> (primo) La tabella di destinazione è la tabella utente o la replica principale. La tabella di origine è la replica dipendente e, in caso di conflitti di aggiornamento tra la tabella di origine e quella di destinazione, le transazioni in conflitto della tabella di origine saranno rifiutate F non viene utilizzato per richieste di sola lettura. <b>S</b> (secondo) La tabella di origine è la tabella utente, replica principale o un'altra origine. La tabella di destinazione è la replica dipendente o un'altra copia e, in caso di conflitti di aggiornamento tra la tabella di origine e la tabella di destinazione, le transazioni in conflitto della tabella di destinazione saranno rifiutate. S è utilizzato per tutte le richieste di sola lettura.

Tabella 103. Colonne nella tabella IBMSNAP\_SUBS\_STMTS (Continua)

Nome colonna	Descrizione
BEFORE_OR_AFTER	<p><b>Tipo di dati:</b> CHAR(1); <b>Impostabile su null:</b> no</p> <p>Un valore che indica quando e dove viene emessa l'istruzione:</p> <p><b>A</b> L'istruzione viene eseguita sul server di destinazione dopo che le righe della serie di risposte vengono applicate.</p> <p><b>B</b> L'istruzione viene eseguita sul server di destinazione prima che le righe della serie di risposte vengono applicate.</p> <p><b>S</b> L'istruzione viene eseguita sul server di controllo Capture prima di aprire i cursori della serie di risposte.</p> <p><b>G</b> Riservato per l'utilizzo tramite replica SQL.</p> <p><b>X</b> Riservato per l'utilizzo tramite replica SQL.</p>
STMT_NUMBER	<p><b>Tipo di dati:</b> SMALLINT; <b>Impostabile su null:</b> no</p> <p>Definisce l'ordine relativo di esecuzione all'interno dell'ambito del valore della colonna BEFORE_OR_AFTER.</p>
EI_OR_CALL	<p><b>Tipo di dati:</b> CHAR(1); <b>Impostabile su null:</b> no</p> <p>Un valore che segnala:</p> <p><b>E</b> L'istruzione SQL deve essere eseguita come EXEC SQL EXECUTE IMMEDIATE.</p> <p><b>C</b> L'istruzione SQL contiene un nome di una procedura memorizzata per essere eseguita come EXEC SQL CALL.</p>
SQL_STMT	<p><b>Tipo di dati:</b> VARCHAR(1024); <b>Impostabile su null:</b> Sì.</p> <p>Uno dei seguenti valori:</p> <p><b>Istruzione</b> L'istruzione SQL deve essere eseguita come un'istruzione EXEC SQL EXECUTE IMMEDIATE se EI_OR_CALL è E.</p> <p><b>Procedura</b> Il nome a 8 byte di una procedura memorizzata SQL, senza parametri, o la parola chiave CALL che viene eseguita come un'istruzione EXEC SQL CALL se EI_OR_CALL è C.</p>
ACCEPT_SQLSTATES	<p><b>Tipo di dati:</b> VARCHAR(50); <b>Impostabile su null:</b> Sì.</p> <p>Uno su dieci valori SQLSTATE a 5 byte, che è stato specificato quando è stata definita la serie di richieste. Questi valori non zero vengono accettati dal programma Apply come un'esecuzione con esito positivo. Tutti gli altri valori provocheranno l'esito negativo dell'esecuzione.</p>

## Control tables at the Monitor control server

The control tables at the Monitor control server contain information about when, how, and whom you want the Replication Alert Monitor to contact when an alert condition occurs. For Linux, UNIX, Windows, and z/OS, you build these control tables to your specifications by using the ASNCLP command-line program or Replication Center. Replication on System i does not have Monitor control tables.

Tabella 104 describes the control tables at the Monitor control server.

*Tabella 104. Control tables at the Monitor control server*

<b>Table name</b>	<b>Description</b>
"IBMSNAP_ALERTS table"	Contains a record of all the alerts issued by the Replication Alert Monitor.
"IBMSNAP_CONDITIONS table" a pagina 483	Contains the alert conditions for which the Replication Alert Monitor will contact someone, and contains the group or individual's name to contact if a particular condition occurs.
"IBMSNAP_CONTACTGRP table" a pagina 488	Contains the individual contacts that make up the contact groups.
"IBMSNAP_CONTACTS table" a pagina 489	Contains information on how the Replication Alert Monitor notifies each person or group when an alert condition that is associated with that contact name occurs.
"IBMSNAP_GROUPS table" a pagina 490	Contains the name and description of each contact group.
"IBMSNAP_MONENQ table" a pagina 490	Used to ensure that only one Replication Alert Monitor program is running per Monitor qualifier.
"IBMSNAP_MONPARMS table" a pagina 490	Contains parameters that you can modify to control the operations of the Monitor program.
"IBMSNAP_MONSERVERS table" a pagina 492	Contains the latest time that a server was monitored by a Replication Alert Monitor program (identified by a Monitor qualifier).
"IBMSNAP_MONTRACE table" a pagina 494	Contains messages from the Monitor program.
"IBMSNAP_MONTRAIL table" a pagina 494	Contains information about each monitor cycle.
"IBMSNAP_SUSPENDS table" a pagina 496	Contains information about temporary suspensions of the monitor program
"IBMSNAP_TEMPLATES table" a pagina 497	Contains information about how often and how long the monitor program is suspended.

## **IBMSNAP\_ALERTS table**

The IBMSNAP\_ALERTS table contains a record of all the alerts issued by the Replication Alert Monitor. The table records what alert conditions occur, at which servers they occur, and when they were detected.

**Server:** Monitor control server

**Non-unique index:** MONITOR\_QUAL, COMPONENT, SERVER\_NAME, SCHEMA\_OR\_QUAL, SET\_NAME, CONDITION\_NAME, ALERT\_CODE

Tabella 105 a pagina 482 provides a brief description of the columns in the IBMSNAP\_ALERTS table.

Tabella 105. Columns in the IBMSNAP\_ALERTS table

Column name	Description
MONITOR_QUAL	<p><b>Data type:</b> CHAR(18); <b>Nullable:</b> No.</p> <p>The Monitor qualifier that identifies which Replication Alert Monitor program issued the alert.</p>
COMPONENT	<p><b>Data type:</b> CHAR(1); <b>Nullable:</b> No.</p> <p>The replication component that is being monitored:</p> <p><b>C</b> Capture program</p> <p><b>A</b> Apply program</p> <p><b>S</b> Q Capture program</p> <p><b>R</b> Q Apply program</p>
SERVER_NAME	<p><b>Data type:</b> CHAR(18); <b>Nullable:</b> No.</p> <p>The name of the Capture control server, Apply control server, Q Capture server, or Q Apply server where the alert condition occurred.</p>
SERVER_ALIAS	<p><b>Data type:</b> CHAR(8); <b>Nullable:</b> Yes.</p> <p>The DB2 alias of the Capture control server, Apply control server, Q Capture server, or Q Apply server where the alert condition occurred.</p>
SCHEMA_OR_QUAL	<p><b>Data type:</b> VARCHAR(128); <b>Nullable:</b> No.</p> <p>The Capture schema, Apply qualifier, Q Capture schema, or Q Apply schema that is being monitored.</p>
SET_NAME	<p><b>Data type:</b> CHAR(18); <b>Nullable:</b> No, with default; <b>Default:</b> Current subscription set.</p> <p>If you set an alert condition for the Apply program, this column specifies the name of the subscription set that is being monitored. If you do not specify a set name, then monitoring is done at the Apply-qualifier level, meaning that every set within the given Apply qualifier is monitored.</p> <p>If you set an alert condition for the Q Apply receive queue depth or spill queue depth, this column specifies the name of the receive queue or spill queue that is being monitored.</p>
CONDITION_NAME	<p><b>Data type:</b> CHAR(18); <b>Nullable:</b> No.</p> <p>The condition code that was tested when the alert was triggered.</p>
OCCURRED_TIME	<p><b>Data type:</b> TIMESTAMP; <b>Nullable:</b> No.</p> <p>The time that the alert condition occurred at the Capture control server, Apply control server, Q Capture server, or Q Apply server.</p>
ALERT_COUNTER	<p><b>Data type:</b> SMALLINT; <b>Nullable:</b> No.</p> <p>The number of times that this alert has been previously detected in consecutive monitor cycles.</p>
ALERT_CODE	<p><b>Data type:</b> CHAR(10); <b>Nullable:</b> No.</p> <p>The message code that was issued when the alert occurred.</p>
RETURN_CODE	<p><b>Data type:</b> INT; <b>Nullable:</b> No.</p> <p>The integer value returned by a user condition.</p>

Tabella 105. Columns in the IBMSNAP\_ALERTS table (Continua)

Column name	Description
NOTIFICATION_SENT	<p><b>Data type:</b> CHAR(1); <b>Nullable:</b> No.</p> <p>A flag that indicates whether a notification message was sent:</p> <p><b>Y</b> A notification message was sent.</p> <p><b>E</b> A notification was not sent because the <b>email_server</b> parameter was not specified.</p> <p><b>N</b> A notification was not sent because the number of notifications already reached the limit set by the <b>max_notifications_per_alert</b> parameter.</p>
ALERT_MESSAGE	<p><b>Data type:</b> VARCHAR(1024); <b>Nullable:</b> No.</p> <p>The text of the message that was sent, including the message code.</p>

## IBMSNAP\_CONDITIONS table

The IBMSNAP\_CONDITIONS table contains the alert conditions for which the Replication Alert Monitor will contact someone, and it contains the group or individual's name to contact if a particular condition occurs. The Replication Alert Monitor can monitor a combination of conditions on Capture control servers, Apply control servers, Q Capture servers, and Q Apply servers.

**Server:** Monitor control server

**Non-unique index:** MONITOR\_QUAL, COMPONENT, SERVER\_NAME, SCHEMA\_OR\_QUAL, SET\_NAME, CONDITION\_NAME

Tabella 106 provides a brief description of the columns in the IBMSNAP\_CONDITIONS table.

Tabella 106. Columns in the IBMSNAP\_CONDITIONS table

Column name	Description
SERVER_NAME	<p><b>Data type:</b> CHAR(18); <b>Nullable:</b> No.</p> <p>The name of the Capture control server, Apply control server, Q Capture server, or Q Apply server where this condition is being monitored.</p>
COMPONENT	<p><b>Data type:</b> CHAR(1); <b>Nullable:</b> No.</p> <p>The replication component that is being monitored:</p> <p><b>C</b> Capture program</p> <p><b>A</b> Apply program</p> <p><b>S</b> Q Capture program</p> <p><b>R</b> Q Apply program</p>
SCHEMA_OR_QUAL	<p><b>Data type:</b> VARCHAR(128); <b>Nullable:</b> No.</p> <p>The Capture schema, Apply qualifier, Q Capture schema, or Q Apply schema that is being monitored.</p>
SET_NAME	<p><b>Data type:</b> CHAR(18); <b>Nullable:</b> No; <b>Default:</b> Current subscription set.</p> <p>If you set an alert condition for the Apply program, this column specifies the name of the subscription set that is being monitored. If you do not specify a set name, then monitoring is done at the Apply-qualifier level, meaning that every set within the given Apply qualifier is monitored.</p>

Tabella 106. Columns in the IBMSNAP\_CONDITIONS table (Continua)

Column name	Description
MONITOR_QUAL	<p><b>Data type:</b> CHAR(18); <b>Nullable:</b> No.</p> <p>The Monitor qualifier that identifies which Replication Alert Monitor program is monitoring the Capture control server, Apply control server, Q Capture server, or Q Apply server for this condition.</p>
SERVER_ALIAS	<p><b>Data type:</b> CHAR(8); <b>Nullable:</b> Yes.</p> <p>The DB2 alias of the Capture control server, Apply control server, Q Capture server, or Q Apply server where this condition is being monitored.</p>
ENABLED	<p><b>Data type:</b> CHAR(1); <b>Nullable:</b> No.</p> <p>A flag that indicates whether the Replication Alert Monitor will process this condition during the next monitoring cycle:</p> <p><b>Y</b>      The Replication Alert Monitor will process this definition during the next cycle.</p> <p><b>N</b>      The Replication Alert Monitor will ignore this definition during the next cycle.</p>
CONDITION_NAME	<p><b>Data type:</b> CHAR(18); <b>Nullable:</b> No.</p> <p>The name of the condition that the Replication Alert Monitor is monitoring at the given Capture control server, Apply control server, Q Capture server, or Q Apply server. Conditions for the Capture program begin with CAPTURE. Conditions for the Apply program begin with APPLY. Conditions for the Q Capture program begin with QCAPTURE. Conditions for the Q Apply program begin with QAPPLY.</p> <p><b>CAPTURE_STATUS</b> The status of the Capture program.</p> <p><b>CAPTURE_ERRORS</b> Whether the Capture program posted any error messages.</p> <p><b>CAPTURE_WARNINGS</b> Whether the Capture program posted any warning messages.</p> <p><b>CAPTURE_LASTCOMMIT</b> The last time the Capture program committed data during the last monitor cycle.</p> <p><b>CAPTURE_CLATENCY</b> The Capture program's current latency.</p> <p><b>CAPTURE_HLATENCY</b> Whether the Capture program's latency is greater than a certain number of seconds.</p> <p><b>CAPTURE_MEMORY</b> The amount of memory (in megabytes) that the Capture program is using.</p>

Tabella 106. Columns in the IBMSNAP\_CONDITIONS table (Continua)

Column name	Description
CONDITION_NAME (Continued)	<b>APPLY_STATUS</b> The status of the Apply program.
	<b>APPLY_SUBSFALING</b> Whether any subscription sets failed.
	<b>APPLY_SUBSINACT</b> Whether any subscription sets either failed or are inactive.
	<b>APPLY_ERRORS</b> Whether the Apply program posts any error messages.
	<b>APPLY_WARNINGS</b> Whether the Apply program posts any warning messages.
	<b>APPLY_FULLREFRESH</b> Whether a full refresh occurred.
	<b>APPLY_REJTRANS (update anywhere)</b> Whether the Apply program rejects transactions in any subscription set.
	<b>APPLY_SUBSDELAY</b> Whether the Apply program delays more than the time that you specified in the PARM_INT parameter.
	<b>APPLY_REWORKED</b> Whether the Apply program reworked any rows at the target table.
	<b>APPLY_LATENCY</b> Whether the end-to-end latency of the Apply program exceeds a threshold.
	CONDITION_NAME (Continued)
<b>QCAPTURE_ERRORS</b> Whether the Q Capture program posted any error messages.	
<b>QCAPTURE_WARNINGS</b> Whether the Q Capture program posted any warning messages.	
<b>QCAPTURE_LATENCY</b> Whether the Q Capture latency (the difference between the last insert into the IBMQREP_CAPMON table and the timestamp of the last transaction that the Q Capture program read in the DB2 log) exceeds a threshold.	
<b>QCAPTURE_MEMORY</b> Whether the memory amount that the Q Capture program used exceeds a threshold.	
<b>QCAPTURE_TRANSIZE</b> Whether a transaction exceeds the MAX_TRANS_SIZE (maximum transaction size) that is set in the IBMQREP_CAPMON table.	
<b>QCAPTURE_SUBSINACT</b> Whether a Q subscription changed to I (inactive) state.	

Tabella 106. Columns in the IBMSNAP\_CONDITIONS table (Continua)

Column name	Description
CONDITION_NAME (Continued)	<p><b>QAPPLY_STATUS</b> Whether the Q Apply program is down.</p> <p><b>QAPPLY_ERRORS</b> Whether the Q Apply program posted any error messages.</p> <p><b>QAPPLY_WARNINGS</b> Whether the Q Apply program posted any warning messages.</p> <p><b>QAPPLY_LATENCY</b> Whether the queue latency (the time it takes for a message to go from the send queue to the receive queue) exceeds a threshold.</p> <p><b>QAPPLY_EELATENCY</b> Whether the end-to-end latency (the time it takes for a transaction to replicate from the source to the target) exceeds a threshold.</p> <p><b>QAPPLY_EXCEPTIONS</b> Whether the Q Apply inserted a row in the IBMQREP_EXCEPTIONS table because of a SQL error or conflict.</p> <p><b>QAPPLY_MEMORY</b> Whether the amount of memory that the Q Apply program used to read messages from a particular receive queue exceeds a threshold.</p> <p><b>QAPPLY_RECVQINACT</b> Whether a receive queue changed to I (inactive) state.</p> <p><b>QAPPLY_SPILLQDEPTH</b> Whether the number of messages on a spill queue exceeds a threshold.</p> <p><b>QAPPLY_QDEPTH</b> Whether the number of messages on a receive queue exceeds a threshold.</p>



Tabella 106. Columns in the IBMSNAP\_CONDITIONS table (Continua)

Column name	Description
PARM_INT	<p><b>Data type:</b> INT; <b>Nullable:</b> Yes.</p> <p>The integer parameter for the condition. The value of this column depends on the value of the CONDITION_NAME column.</p> <p><b>CAPTURE_LASTCOMMIT</b> Threshold in seconds.</p> <p><b>CAPTURE_CLATENCY</b> Threshold in seconds.</p> <p><b>CAPTURE_HLATENCY</b> Threshold in seconds.</p> <p><b>CAPTURE_MEMORY</b> Threshold in megabytes.</p> <p><b>APPLY_SUBSDELAY</b> Threshold in seconds.</p> <p><b>APPLY_REWORKED</b> Threshold in rows reworked.</p> <p><b>APPLY_LATENCY</b> Threshold in seconds.</p> <p><b>QCAPTURE_LATENCY</b> Threshold in seconds</p> <p><b>QCAPTURE_MEMORY</b> Threshold in megabytes</p> <p><b>QCAPTURE_TRANSIZE</b> Threshold in megabytes</p> <p><b>QAPPLY_EELATENCY</b> Threshold in seconds</p> <p><b>QAPPLY_LATENCY</b> Threshold in seconds</p> <p><b>QAPPLY_MEMORY</b> Threshold in megabytes</p> <p><b>QAPPLY_SPILLQDEPTH</b> Threshold in number of messages.</p> <p><b>QAPPLY_QDEPTH</b> Threshold in number of messages.</p>

Tabella 106. Columns in the IBMSNAP\_CONDITIONS table (Continua)

Column name	Description
PARM_CHAR	<p><b>Data type:</b> VARCHAR(128); <b>Nullable:</b> Yes.</p> <p>The character parameter for the condition. This column holds additional strings used by the condition.</p> <p>The CAPTURE_STATUS and APPLY_STATUS conditions use the value of this column. The value of this column is a string concatenating three parameters separated by commas:</p> <ul style="list-style-type: none"> <li>Capture server or Apply control server. <ul style="list-style-type: none"> <li><b>z/OS</b> This is the DB2 subsystem name.</li> </ul> </li> <li>Remote DB2 instance name (only when the server is remote).</li> <li>Remote hostname.</li> </ul> <p>If the value is NULL or a zero length string, the Monitor program uses the following defaults:</p> <ul style="list-style-type: none"> <li>The CURRENT SERVER value from the Capture or Apply control server.</li> <li>The remote DB2 instance name value: <ul style="list-style-type: none"> <li><b>Linux UNIX</b> This value is the name of the user ID that was used when the UNIX server was connected.</li> <li><b>Windows</b> This value is DB.</li> </ul> </li> <li>The value of the hostname in the DB2 node directory.</li> </ul>
CONTACT_TYPE	<p><b>Data type:</b> CHAR(1); <b>Nullable:</b> No.</p> <p>A flag that indicates whether to contact an individual or a group if this condition occurs:</p> <p><b>C</b> Individual contact</p> <p><b>G</b> Group of contacts</p> <p><b>z/OS Z</b> Send alert to z/OS console.</p>
CONTACT	<p><b>Data type:</b> VARCHAR(127); <b>Nullable:</b> No.</p> <p>The name of the individual contact or group of contacts to be notified if this condition occurs. For alerts that are sent to the z/OS console, the value of this column is CONSOLE.</p>

## IBMSNAP\_CONTACTGRP table

The IBMSNAP\_CONTACTGRP table contains the individual contacts that make up contact groups. You can specify for the Replication Alert Monitor to contact these groups of individuals if an alert condition occurs. An individual can belong to multiple contact groups (the columns are not unique).

**Server:** Monitor control server

**Non-unique index:** GROUP\_NAME, CONTACT\_NAME

Tabella 107 a pagina 489 provides a brief description of the columns in the IBMSNAP\_CONTACTGRP table.

Tabella 107. Columns in the IBMSNAP\_CONTACTGRP table

Column name	Description
GROUP_NAME	<b>Data type:</b> VARCHAR(127); <b>Nullable:</b> No.  The name of the contact group.
CONTACT_NAME	<b>Data type:</b> VARCHAR(127); <b>Nullable:</b> No.  A contact name that is part of the group. These individuals are specified in the Monitor contacts (IBMSNAP_CONTACTS) table.

## IBMSNAP\_CONTACTS table

The IBMSNAP\_CONTACTS table contains the necessary information for the Replication Alert Monitor to use to notify individuals when an alert condition that is associated with the individuals (or their group) occurs. One individual per row is specified.

**Server:** Monitor control server

**Non-unique index:** CONTACT\_NAME

Tabella 108 provides a brief description of the columns in the IBMSNAP\_CONTACTS table.

Tabella 108. Columns in the IBMSNAP\_CONTACTS table

Column name	Description
CONTACT_NAME	<b>Data type:</b> VARCHAR(127); <b>Nullable:</b> No.  The name of the contact. Only an individual contact is allowed. Group names are not supported. For alerts that are sent to the z/OS console, the value of this column is CONSOLE.
EMAIL_ADDRESS	<b>Data type:</b> VARCHAR(128); <b>Nullable:</b> No.  The main e-mail or pager address for this contact. For alerts that are sent to the z/OS console, this column is ignored.
ADDRESS_TYPE	<b>Data type:</b> CHAR(1); <b>Nullable:</b> Yes.  A flag that indicates whether the e-mail address for this contact is an e-mail account or a pager address:  E      The e-mail address is for an e-mail account. P      The e-mail address is for a pager. Z      Alerts are sent to the z/OS console.
DELEGATE	<b>Data type:</b> VARCHAR(127); <b>Nullable:</b> Yes.  The contact name to receive the notifications in a delegation period. Only an individual contact name is allowed. Group names are not supported.
DELEGATE_START	<b>Data type:</b> DATE; <b>Nullable:</b> Yes.  The start date of a delegation period when notifications will be sent to the individual named in the DELEGATE column.
DELEGATE_END	<b>Data type:</b> DATE; <b>Nullable:</b> Yes.  The end date of a delegation period.

Tabella 108. Columns in the IBMSNAP\_CONTACTS table (Continua)

Column name	Description
DESCRIPTION	<b>Data type:</b> VARCHAR(1024); <b>Nullable:</b> Yes.  A description of the contact.

## IBMSNAP\_GROUPS table

The IBMSNAP\_GROUPS table contains the name and description of each contact group. One group per row is specified.

**Server:** Monitor control server

**Non-unique index:** GROUP\_NAME

Tabella 109 provides a brief description of the columns in the IBMSNAP\_GROUPS table.

Tabella 109. Columns in the IBMSNAP\_GROUPS table

Column name	Description
GROUP_NAME	<b>Data type:</b> VARCHAR(127); <b>Nullable:</b> Yes.  The name of the contact group.
DESCRIPTION	<b>Data type:</b> VARCHAR(1024); <b>Nullable:</b> Yes.  A description of the contact group.

## IBMSNAP\_MONENQ table

The IBMSNAP\_MONENQ table is reserved for future use by replication.

**Server:** Monitor control server

**Non-unique index:** MONITOR\_QUAL

Tabella 110 provides a brief description of the column in the IBMSNAP\_MONENQ table.

Tabella 110. Columns in the IBMSNAP\_MONENQ table

Column name	Description
MONITOR_QUAL	<b>Data type:</b> CHAR(18); <b>Nullable:</b> No.  Reserved for future use by replication.

## IBMSNAP\_MONPARMS table

The IBMSNAP\_MONPARMS table contains parameters that you can modify to control the operations of the Replication Alert Monitor.

You can define these parameters to set values such as the number of notification messages that the Monitor program will send when an alert condition is met. If you make changes to the parameters in this table, the Monitor program reads your modifications only during startup.

**Server:** Monitor control server

**Index:** MONITOR\_QUAL

**Default schema:** ASN

This table contains information that you can update by using SQL.

Tabella 111 provides a brief description of the columns in the IBMSNAP\_MONPARMS table.

*Tabella 111. Columns in the IBMSNAP\_MONPARMS table*

Column name	Description
MONITOR_QUAL	<b>Data type:</b> CHAR(18); <b>Nullable:</b> No.  The Monitor qualifier matches the parameters to the Replication Alert Monitor program to which these parameters apply.
ALERT_PRUNE_LIMIT	<b>Data type:</b> INT; <b>Nullable:</b> No, with default; <b>Default:</b> 10080 minutes (7 days).  A flag that indicates how old the data is before it will be pruned from the table.
AUTOPRUNE	<b>Data type:</b> CHAR(1); <b>Nullable:</b> No, with default; <b>Default:</b> Y.  A flag that indicates whether the Monitor program automatically prunes rows that are no longer needed from the IBMSNAP_ALERTS, IBMSNAP_MONTRACE, and IBMSNAP_MONTRAIL control tables:  Y      Autopruning is on. N      Autopruning is off.
EMAIL_SERVER	<b>Data type:</b> INT(128); <b>Nullable:</b> Yes.  The address of the e-mail server that is using the SMTP protocol.
LOGREUSE	<b>Data type:</b> CHAR(1); <b>Nullable:</b> No, with default; <b>Default:</b> N.  A flag that indicates whether the Monitor program overwrites the Monitor log file or appends to it.  Y      The Monitor program reuses the log file by first deleting it and then recreating it when the Monitor program is restarted. N      The Monitor program appends new information to the Monitor log file.
LOGSTDOUT	<b>Data type:</b> CHAR(1); <b>Nullable:</b> No, with default; <b>Default:</b> N.  A flag that indicates where the Monitor program directs the log file messages:  Y      The Monitor program directs log file messages to both the standard out (STDOUT) and the log file. N      The Monitor program directs most log file messages to the log file only. Initialization messages go to both the standard out (STDOUT) and the log file.
NOTIF_PER_ALERT	<b>Data type:</b> INT; <b>Nullable:</b> No, with default; <b>Default:</b> 3.  The number of notification messages that will be sent when an alert condition is met.
NOTIF_MINUTES	<b>Data type:</b> INT; <b>Nullable:</b> No, with default; <b>Default:</b> 60.  The number of minutes that you will receive notification messages when an alert condition is met.

Tabella 111. Columns in the IBMSNAP\_MONPARMS table (Continua)

Column name	Description
MONITOR_ERRORS	<p><b>Data type:</b> VARCHAR(128); <b>Nullable:</b> Yes.</p> <p>Specifies the e-mail address where notification messages are sent whenever an error occurs that is related to the operation of the Replication Alert Monitor.</p>
MONITOR_INTERVAL	<p><b>Data type:</b> INT; <b>Nullable:</b> No, with default; <b>Default:</b> 300 (5 minutes).</p> <p>How often, in seconds, the Replication Alert Monitor runs to monitor the alert conditions that were selected.</p>
MONITOR_PATH	<p><b>Data type:</b> VARCHAR(1040); <b>Nullable:</b> Yes.</p> <p>The path where the output from the Monitor program is sent.</p>
RUNONCE	<p><b>Data type:</b> CHAR(1); <b>Nullable:</b> No, with default; <b>Default:</b> N.</p> <p>A flag that indicates whether the Monitor program will check for the alert conditions that were selected:</p> <p>Y        The Monitor program checks for any alert conditions.</p> <p>N        The Monitor program does not check for any alert conditions.</p> <p>If RUNONCE is set to y, then the MONITOR_INTERVAL is ignored.</p>
TERM	<p><b>Data type:</b> CHAR(1); <b>Nullable:</b> No, with default; <b>Default:</b> N.</p> <p>A flag that indicates whether the Monitor program terminates when DB2 is placed in quiesce mode:</p> <p>N        The Monitor program stays active when DB2 is quiesced and waits for DB2 to be unquiesced.</p> <p>Y        The Monitor program terminates when DB2 is quiesced.</p> <p>Regardless of the value of TERM, a monitor program stops when DB2 shuts down. When DB2 starts again, you must restart the monitor program.</p>
TRACE_LIMIT	<p><b>Data type:</b> INT; <b>Nullable:</b> No, with default; <b>Default:</b> 10080.</p> <p>The number of minutes that rows remain in the IBMSNAP_MONTRACE table before they are eligible for pruning. During the pruning process, the rows in the Monitor trace table are pruned if the number of minutes (current timestamp - the time a row was inserted in the IBMSNAP_MONTRACE table) exceeds the value of TRACE_LIMIT.</p>
ARCH_LEVEL	<p><b>Data type:</b> CHAR(8); <b>Nullable:</b> No, with default; <b>Default:</b> 0905.</p> <p>The level of the monitor control tables. For Version 9.5 the level is 0905. Other values are 0802 and 0901.</p> <p><b>Important:</b> When updating the IBMSNAP_MONPARMS table, do not change the value in this column.</p>
DELAY	<p><b>Data type:</b> CHAR(1); <b>Nullable:</b> Yes <b>Default:</b> N</p> <p>A flag that indicates whether the monitor program should delay the start of monitoring by waiting until the specified monitor interval has elapsed, even for the first iteration.</p>

## IBMSNAP\_MONSERVICES table

The IBMSNAP\_MONSERVICES table contains information about the last time that the Replication Alert Monitor monitored a Capture control server, Apply control server, Q Capture server, or Q Apply server.

**Server:** Monitor control server

**Non-unique index:** MONITOR\_QUAL, SERVER\_NAME

Tabella 112 provides a brief description of the columns in the IBMSNAP\_MONSERVERS table.

Tabella 112. Columns in the IBMSNAP\_MONSERVERS table

Column name	Description
MONITOR_QUAL	<b>Data type:</b> CHAR(18); <b>Nullable:</b> No.  The Monitor qualifier that identifies which Replication Alert Monitor was monitoring the Capture control server, Apply control server, Q Capture server, or Q Apply server.
SERVER_NAME	<b>Data type:</b> CHAR(18); <b>Nullable:</b> No.  The name of the Capture control server, Apply control server, Q Capture server, or Q Apply server that the Replication Alert Monitor was monitoring.
SERVER_ALIAS	<b>Data type:</b> CHAR(8); <b>Nullable:</b> Yes.  The DB2 alias of the Capture control server, Apply control server, Q Capture server, or Q Apply server that the Replication Alert Monitor was monitoring.
LAST_MONITOR_TIME	<b>Data type:</b> TIMESTAMP; <b>Nullable:</b> Yes.  The time (at the Capture control server, Apply control server, Q Capture server, or Q Apply server) that the Replication Alert Monitor program last connected to this server. This value is used as a lower bound value to fetch messages from the control tables and is the same value that START_MONITOR_TIME from the last successful monitor cycle.
START_MONITOR_TIME	<b>Data type:</b> TIMESTAMP; <b>Nullable:</b> Yes.  The time (at the Capture control server, Apply control server, Q Capture server, or Q Apply server) that the Replication Alert Monitor connected to the Capture control server, Apply control server, Q Capture server, or Q Apply server. This value is used as an upper bound value to fetch alert messages from the control tables.
END_MONITOR_TIME	<b>Data type:</b> TIMESTAMP; <b>Nullable:</b> Yes.  The time (at the Capture control server, Apply control server, Q Capture server, or Q Apply server) that the Replication Alert Monitor ended monitoring this server.
LASTRUN	<b>Data type:</b> TIMESTAMP; <b>Nullable:</b> No.  The last time (at the Monitor control server) when the Replication Alert Monitor started to process the Capture control server, Apply control server, Q Capture server, or Q Apply server.
LASTSUCCESS	<b>Data type:</b> TIMESTAMP; <b>Nullable:</b> Yes.  The value from the LASTRUN column of the last time (at the Monitor control server) when the Replication Alert Monitor successfully completed processing the Capture control server, Apply control server, Q Capture server, or Q Apply server. If the monitoring of this server keeps failing, the value could be the same (the history of this column is stored in the IBMSNAP_MONTRAIL table).

Tabella 112. Columns in the IBMSNAP\_MONSERVERS table (Continua)

Column name	Description
STATUS	<p><b>Data type:</b> SMALLINT; <b>Nullable:</b> No.</p> <p>A flag that indicates the status of the monitoring cycle:</p> <p>-1 The Replication Alert Monitor failed to process this server successfully.</p> <p>0 The Replication Alert Monitor processed this server successfully.</p> <p>1 The Replication Alert Monitor is currently processing this server.</p>

## IBMSNAP\_MONTRACE table

The IBMSNAP\_MONTRACE table contains audit trail information for the Replication Alert Monitor. Everything that the Monitor program does is recorded in this table, which makes it one of the best places for you to look if a problem with the Monitor program occurs.

**Server:** Monitor control server

**Non-unique index:** MONITOR\_QUAL, TRACE\_TIME

Tabella 113 provides a brief description of the columns in the IBMSNAP\_MONTRACE table.

Tabella 113. Columns in the IBMSNAP\_MONTRACE table

Column name	Description
MONITOR_QUAL	<p><b>Data type:</b> CHAR(18); <b>Nullable:</b> No.</p> <p>The Monitor qualifier that identifies which Replication Alert Monitor issued the message.</p>
TRACE_TIME	<p><b>Data type:</b> TIMESTAMP; <b>Nullable:</b> No.</p> <p>The timestamp when the message was inserted into this table.</p>
OPERATION	<p><b>Data type:</b> CHAR(8); <b>Nullable:</b> No.</p> <p>A value used to classify messages:</p> <p><b>ERROR</b> An error message</p> <p><b>WARNING</b> A warning message</p> <p><b>INFO</b> An informational message</p>
DESCRIPTION	<p><b>Data type:</b> VARCHAR(1024); <b>Nullable:</b> No.</p> <p>The message code and text.</p>

## IBMSNAP\_MONTRAIL table

The IBMSNAP\_MONTRAIL table contains information about each monitor cycle. The Replication Alert Monitor inserts one row for each Capture control server, Apply control server, Q Capture server, and Q Apply server that it monitors.

**Server:** Monitor control server

**Non-unique index:** None



Tabella 114 provides a brief description of the columns in the IBMSNAP\_MONTRAIL table.

Tabella 114. Columns in the IBMSNAP\_MONTRAIL table

Column name	Description
MONITOR_QUAL	<b>Data type:</b> CHAR(18); <b>Nullable:</b> No  The Monitor qualifier that identifies which Replication Alert Monitor was monitoring the Capture control server, Apply control server, Q Capture server, or Q Apply server
SERVER_NAME	<b>Data type:</b> CHAR(18); <b>Nullable:</b> No  The name of the Capture control server, Apply control server, Q Capture server, or Q Apply server that the Replication Alert Monitor was monitoring.
SERVER_ALIAS	<b>Data type:</b> CHAR(8); <b>Nullable:</b> Yes  The DB2 alias of the Capture control server, Apply control server, Q Capture server, or Q Apply server that the Replication Alert Monitor was monitoring.
STATUS	<b>Data type:</b> SMALLINT; <b>Nullable:</b> No  A flag that indicates the status of the monitoring cycle: -1 The Replication Alert Monitor failed to process this server successfully. 0 The Replication Alert Monitor processed this server successfully. 1 The Replication Alert Monitor is currently processing this server.
LASTRUN	<b>Data type:</b> TIMESTAMP; <b>Nullable:</b> No  The time (at the Monitor control server) when the Replication Alert Monitor program last started to process the Capture control server, Apply control server, Q Capture server, or Q Apply server.
LASTSUCCESS	<b>Data type:</b> TIMESTAMP; <b>Nullable:</b> Yes  The last time (at the Monitor control server) when the Replication Alert Monitor successfully completed processing the Capture control server, Apply control server, Q Capture server, or Q Apply server.
ENDTIME	<b>Data type:</b> TIMESTAMP; <b>Nullable:</b> No, with default  The time when this row was inserted into this table. Default: Current timestamp
LAST_MONITOR_TIME	<b>Data type:</b> TIMESTAMP; <b>Nullable:</b> Yes  The time (at the Capture control server, Apply control server, Q Capture server, or Q Apply server) when the Replication Alert Monitor last connected to the Capture control server, Apply control server, Q Capture server, or Q Apply server. This value is used as a lower bound value to fetch messages from the control tables and is the same value as START_MONITOR_TIME from the previous successful monitor cycle.
START_MONITOR_TIME	<b>Data type:</b> TIMESTAMP; <b>Nullable:</b> Yes  The last time when the Replication Alert Monitor started to monitor the Capture control server, Apply control server, Q Capture server, or Q Apply server.
END_MONITOR_TIME	<b>Data type:</b> TIMESTAMP; <b>Nullable:</b> Yes  The last time when the Replication Alert Monitor finished monitoring the Capture control server, Apply control server, Q Capture server, or Q Apply server.

Tabella 114. Columns in the IBMSNAP\_MONTRAIL table (Continua)

Column name	Description
SQLCODE	<b>Data type:</b> INT; <b>Nullable:</b> Yes  The SQLCODE of any errors that occurred during this monitor cycle.
SQLSTATE	<b>Data type:</b> CHAR(5); <b>Nullable:</b> Yes  The SQLSTATE of any errors that occurred during this monitor cycle.
NUM_ALERTS	<b>Data type:</b> INT; <b>Nullable:</b> No  The number of alert conditions that occurred during this monitor cycle.
NUM_NOTIFICATIONS	<b>Data type:</b> INT; <b>Nullable:</b> No  The number of notifications that were sent during this monitor cycle.
SUSPENSION_NAME	<b>Data type:</b> VARCHAR(128); <b>Nullable:</b> Yes  The name of the suspension that is used to stop the operation of the monitor for defined periods.

## IBMSNAP\_SUSPENDS table

The IBMSNAP\_SUSPENDS table stores information about temporary suspensions of the monitor program.

**Server:** Monitor control server

**Default schema:** ASN

**Primary key:** SUSPENSION\_NAME

**Unique index:** SERVER\_NAME, TEMPLATE\_NAME, START

Tabella 115 provides a brief description of the columns in the IBMSNAP\_SUSPENDS table.

Tabella 115. Columns in the IBMSNAP\_SUSPENDS table

Column name	Description
SUSPENSION_NAME	<b>Data type:</b> VARCHAR(128); <b>Nullable:</b> No  The name of the monitor suspension.
SERVER_NAME	<b>Data type:</b> CHAR(18); <b>Nullable:</b> No  The name of the Q Capture server, Q Apply server, Capture control server, or Apply control server where monitoring is suspended.
SERVER_ALIAS	<b>Data type:</b> CHAR(18); <b>Nullable:</b> Yes  The alias of the server where monitoring is suspended.
TEMPLATE_NAME	<b>Data type:</b> VARCHAR(128); <b>Nullable:</b> Yes  The name of the monitor suspension template. If the value of this column does not exist in the IBMSNAP_TEMPLATES control table, the monitor suspends one time from the START timestamp until the STOP timestamp.

Tabella 115. Columns in the IBMSNAP\_SUSPENDS table (Continua)

Column name	Description
START	<b>Data type:</b> TIMESTAMP; <b>Nullable:</b> No  The time to start using the template. If no template is specified, this is the start time of the suspension.
STOP	<b>Data type:</b> TIMESTAMP; <b>Nullable:</b> No  The time to stop using the template. If no template is specified, this is the end time of the suspension.

## IBMSNAP\_TEMPLATES table

The IBMSNAP\_TEMPLATES table stores information about how often and how long the monitor program is suspended. This information is called a monitor suspension template.

**Server:** Monitor control server

**Default schema:** ASN

**Unique index:** TEMPLATE\_NAME

Tabella 116 provides a brief description of the columns in the IBMSNAP\_TEMPLATES table.

Tabella 116. Columns in the IBMSNAP\_TEMPLATES table

Column name	Description
TEMPLATE_NAME	<b>Data type:</b> VARCHAR(128); <b>Nullable:</b> No  The name of the monitor suspension template.
START_TIME	<b>Data type:</b> TIME; <b>Nullable:</b> No  The time of the day to start the suspension. Default: 00:00:00
WDAY	<b>Data type:</b> SMALLINT; <b>Nullable:</b> Yes  The day of the week in which the suspension begins, starting with 0 for Sunday and continuing to 6 for Saturday. A null values means the suspension can begin on any day.
DURATION	<b>Data type:</b> INTEGER; <b>Nullable:</b> No  The duration of the suspension in minutes.

## Tabelle sul server di destinazione

Vari tipi di tabelle di destinazione sono memorizzati nel server di destinazione. Se non viene utilizzata una tabella esistente come propria tabella di destinazione, il programma della riga comandi ASNCLP o il Centro di replica creano la tabella di destinazione in base alla specifiche dell'utente basate su come l'utente definisce il membro della serie di richieste.

Tabella 117 a pagina 498 descrive le tabelle nel server di destinazione.

Tabella 117. Consultazione rapida per le tabelle di destinazione

Nome tabella	Descrizione
"Tabella aggregati di base"	Contiene i dati che sono stati aggregati dalla tabella di origine.
"Tabella aggregati di modifica"	Contiene dati che sono stati aggregati da una tabella CD.
"Destinazioni CCD" a pagina 80	Contiene informazioni sulle modifiche che si verificano all'origine e contiene ulteriori colonne per identificare l'ordinamento sequenziale per queste modifiche.
"Tabella oraria" a pagina 501	Una copia dei dati di origine, con un'ulteriore colonna che registra l'ora specifica nella registrazione di origine in cui i dati sono stati sottoposti a commit.
"Tabella di replica" a pagina 502	Un tipo di tabella di destinazione utilizzata per la replica update-anywhere.
"Tabella copia utente" a pagina 502	Una copia della tabella di origine.

## Tabella aggregati di base

Una tabella aggregati di base è una tabella di destinazione che contiene i risultati di funzioni aggregate eseguite sui dati ubicati nella tabella di origine.

*schema.aggregati\_base*

**Server:** server di destinazione

**Importante:** Se si utilizza SQL per aggiornare questa tabella, si corre il rischio di perdere i propri aggiornamenti il programma Apply esegue un aggiornamento completo.

Tabella 118 fornisce una breve descrizione delle colonne nella tabella aggregati di base.

Tabella 118. Colonne nella tabella aggregati di base

Nome colonna	Descrizione
<i>colonne utente</i>	I dati aggregati che sono stati calcolati dalla tabella di origine.
IBMSNAP_LLOGMARKER	La data/ora corrente sul server di origine quando è iniziata l'aggregazione dei dati nella tabella di origine.
IBMSNAP_HLOGMARKER	La data/ora corrente sul server di origine quando è completata l'aggregazione dei dati nella tabella di origine.

## Tabella aggregati di modifica

Una tabella aggregati di modifica è una tabella di destinazione che contiene i risultati di funzioni aggregate eseguite sui dati nella tabella CD (change-data). Questa tabella è simile alla tabella aggregati di base, eccetto per il fatto che le funzioni che sono eseguite nella tabella CD vengono effettuate solo per modifiche che si verificano durante uno specifico intervallo di tempo.

*schema.aggregati\_modifica*

**Server:** server di destinazione

**Importante:** Se si utilizza SQL per aggiornare questa tabella, si corre il rischio di perdere i propri aggiornamenti il programma Apply esegue un aggiornamento completo.

Tabella 119 fornisce breve descrizione delle colonne nella tabella aggregati di modifica.

Tabella 119. Colonne nella tabella aggregati di modifica

Nome colonna	Descrizione
<i>colonne chiave utente</i>	Le colonne che compongono la chiave di destinazione.
<i>colonna non chiave utente</i>	La colonna di dati non chiave dalla tabella di origine. I nomi delle colonne in questa tabella di destinazione non devono corrispondere ai nomi delle colonne nella tabella di origine, ma i tipi di dati devono corrispondere.
<i>colonne calcolate utente</i>	Colonne definite dall'utente derivate da espressioni SQL. È possibile utilizzare colonne calcolate con funzioni SQL per convertire i tipi di dati di origine in tipi di dati di destinazione differenti.
IBMSNAP_LLOGMARKER	Il valore IBMSNAP_LOGMARKER o IBMSNAP_LLOGMARKER meno recente presente nelle righe della tabella (CD+UOW) o CCD che vengono aggregate.
IBMSNAP_HLOGMARKER	Il valore IBMSNAP_LOGMARKER o IBMSNAP_HLOGMARKER più recente presente nelle righe della tabella (CD+UOW) o CCD che vengono aggregate.

## Destinazioni CCD

Le tabelle CCD (Consistent-change-data) forniscono dati transazionali con commit che possono essere letti e utilizzati da altre applicazioni, come ad esempio WebSphere DataStage. È inoltre possibile utilizzare una tabella CCD per il controllo dei dati di origine o per mantenere una cronologia delle modalità di utilizzo dei dati.

Ad esempio, è possibile tenere traccia dei confronti dei dati precedenti e successivi, di quando si sono verificate le modifiche, e dell'ID utente che ha aggiornato la tabella di origine.

Per definire una tabella di destinazione di sola lettura che conservi la cronologia della tabella di origine, definire la tabella CCD di destinazione in modo che includa i seguenti attributi:

### Non concentrata

Per conservare un record di tutte le modifiche di origine, definire la tabella CCD in modo che sia non concentrata. In tal modo, essa memorizzerà una riga per ciascuna modifica che viene eseguita. Poiché le tabelle non concentrate contengono diverse righe con lo stesso valore chiave, non definire un indice univoco. Una tabella CCD non concentrata contiene una riga per ciascuna operazione UPDATE, INSERT o DELETE, in tal modo, conserva una cronologia delle operazioni eseguite nella tabella di origine. Se si catturano le operazioni UPDATE come operazioni INSERT e DELETE (per le colonne della chiave di partizione), la tabella CCD conterrà due righe per ciascun aggiornamento, una riga per DELETE e una riga per INSERT.

### Completa o incompleta

È possibile scegliere se si desidera che la tabella CCD sia completa o incompleta. Poiché le tabelle CCD incomplete inizialmente non contengono una serie completa di righe di origine, creare una tabella CCD incompleta

per conservare la cronologia degli aggiornamenti in una tabella di origine (gli aggiornamenti da quando il programma Apply ha iniziato ad inserire i dati nella tabella CCD).

#### Includi colonne UOW

Per migliorare la funzione di controllo, includere le colonne aggiuntive dalla tabella UOW. Per una maggiore identificazione orientata verso l'utente, le colonne per l'ID correlazione di DB2 per z/OS e l'ID autorizzazione principale o il nome del lavoro di System i e il profilo dell'utente sono disponibili nella tabella UOW.

Per definizione, una tabella CCD include sempre le seguenti colonne in aggiunta alle colonne replicate dalla tabella di origine:

Colonna	Descrizione
IBMSNAP_INTENTSEQ	<p><b>Tipo di dati:</b> CHAR(10) PER DATI BIT; <b>Impostabile su null:</b> No</p> <p>Un numero di sequenza che identifica una modifica in modo univoco. Tale valore è ascendente in una transazione.</p> <p><b>z/OS</b> Il numero della sequenza della registrazione (LRSN o RBA) di ciascun aggiornamento, eliminazione e inserimento.</p>
IBMSNAP_OPERATION	<p><b>Tipo di dati:</b> CHAR(1); <b>Impostabile su null:</b> no</p> <p>Un flag che indica il tipo di operazione: I (INSERT), U (UPDATE) o D (DELETE).</p>
IBMSNAP_COMMITSEQ	<p><b>Tipo di dati:</b> CHAR(10) PER DATI BIT; <b>Impostabile su null:</b> No</p> <p>Un numero di sequenza per ciascuna riga all'interno di una transazione.</p> <p><b>z/OS</b> Il numero della sequenza della registrazione (LRSN or RBA) del record di commit dell'origine.</p>
IBMSNAP_LOGMARKER	<p><b>Tipo di dati:</b> TIMESTAMP; <b>Impostabile su null:</b> no</p> <p>L'ora in cui è stato eseguito il commit dei dati.</p>

Quando si crea una tabella CCD incompleta (COMPLETE=N) con il programma della riga comandi ASNCLP o con il Centro di replica, è possibile specificare ulteriori colonne di controllo. Nella seguente tabella vengono descritte queste colonne:

Colonna	Descrizione
IBMSNAP_AUTHID	<p><b>Tipo di dati:</b> VARCHAR(30); <b>Impostabile su null:</b> sì</p> <p>L'ID utente che ha aggiornato la tabella di origine.</p> <p><b>z/OS</b> Questa colonna è l'ID di autorizzazione primario.</p>
IBMSNAP_AUTHTKN	<p><b>Tipo di dati:</b> VARCHAR(30); <b>Impostabile su null:</b> sì</p> <p>Il token di autorizzazione associato alla transazione.</p> <p><b>z/OS</b> L'ID di correlazione (normalmente un nome lavoro) che ha eseguito l'aggiornamento dell'origine.</p>
IBMSNAP_PLANID	<p><b>Tipo di dati:</b> VARCHAR(8); <b>Impostabile su null:</b> Sì</p> <p><b>z/OS</b> Il nome del piano associato alla transazione. Questa colonna sarà nulla per DB2 per Linux, UNIX e Windows.</p>
IBMSNAP_UOWID	<p><b>Tipo di dati:</b> CHAR(10) FOR BIT DATA; <b>Impostabile su null:</b> Sì</p> <p>L'identificatore UOW (unit-of-work) dal record di registrazione per una riga.</p> <p><b>z/OS</b> L'identificatore unit-of-work, a volte chiamato URID (unit-of-recovery ID) della transazione.</p>

## Tabella oraria

La tabella oraria contiene una copia dei dati di origine, con un'ulteriore colonna di sistema (IBMSNAP\_LOGMARKER) contenente la data/ora del momento in cui, approssimativamente, la particolare riga è stata inserita o aggiornata sul server di origine.

*schema.punto\_nel\_tempo*

**Server:** server di destinazione

**Importante:** Se si utilizza SQL per aggiornare questa tabella, si corre il rischio di perdere i propri aggiornamenti il programma Apply esegue un aggiornamento completo.

Tabella 120 fornisce una breve descrizione delle colonne nella tabella oraria.

Tabella 120. Colonne nella tabella oraria

Nome colonna	Descrizione
<i>colonne chiave utente</i>	Le colonne che compongono la chiave di destinazione.

Tabella 120. Colonne nella tabella oraria (Continua)

Nome colonna	Descrizione
<i>colonna non chiave utente</i>	La colonna di dati non chiave dalla tabella o vista di origine. I nomi delle colonne in questa tabella di destinazione non devono corrispondere ai nomi delle colonne nella tabella di origine, ma i tipi di dati devono corrispondere.
<i>colonne calcolate utente</i>	Colonne definite dall'utente derivate da espressioni SQL. È possibile utilizzare colonne calcolate con funzioni SQL per convertire i tipi di dati di origine in tipi di dati di destinazione differenti.
IBMSNAP_LOGMARKER	L'ora approssimativa del commit sul server di controllo Capture. Questa colonna è nulla a seguito di un aggiornamento completo.

## Tabella di replica

La tabella di replica deve avere le stesse colonne chiave della tabella di origine. A causa di queste analogie, la tabella di replica può essere utilizzata come tabella di origine per altre serie di richieste. La conversione di una tabella di destinazione in una tabella di origine viene effettuata automaticamente quando viene definito un tipo di destinazione della replica e specificato l'attributo CHANGE DATA CAPTURE.

*schema.replica*

**Server:** server di destinazione

Questa tabella contiene le informazioni che si possono aggiornare utilizzando SQL.

Tabella 121 fornisce una breve descrizione delle colonne nella tabella di replica.

Tabella 121. Colonne nella tabella di

Nome colonna	Descrizione
<i>colonne chiave utente</i>	Le colonne che compongono la chiave di destinazione, che deve essere la stessa chiave primaria della tabella principale.
<i>colonna non chiave utente</i>	La colonna di dati non chiave dalla tabella di origine. I nomi delle colonne in questa tabella di destinazione non devono corrispondere ai nomi delle colonne nella tabella di origine, ma i tipi di dati devono corrispondere.

## Tabella copia utente

La tabella copia utente è una tabella di destinazione che contiene una copia delle colonne della tabella di origine. Questa tabella di destinazione può essere una serie secondaria di righe o colonne della tabella di origine, ma non può contenere ulteriori colonne.

*schema.copia\_utente*

**Server:** server di destinazione

**Importante:** Se si utilizza SQL per aggiornare questa tabella, si corre il rischio di perdere i propri aggiornamenti il programma Apply esegue un aggiornamento completo.

Eccetto per l'impostazione secondaria e l'incremento di dati, una tabella copia utente riflette uno stato valido della tabella di origine, ma non necessariamente lo stato più corrente. I riferimenti alle tabelle copia utente (o qualsiasi altro tipo di



tabella di destinazione) riducono il rischio di problemi di conflitto che derivano da un volume elevato di accesso diretto alle tabelle di origine. L'accesso alle tabelle copia utente locali è molto più veloce dell'utilizzo della rete per accedere alle tabelle di origine remota per ogni interrogazione.

Tabella 122 fornisce una breve descrizione delle colonne nella tabella copia utente.

*Tabella 122. Colonne nella tabella copia utente*

<b>Nome colonna</b>	<b>Descrizione</b>
<i>colonne chiave utente</i>	Le colonne che compongono la chiave di destinazione.
<i>colonna non chiave utente</i>	La colonna di dati non chiave dalla tabella o vista di origine. I nomi delle colonne in questa tabella di destinazione non devono corrispondere ai nomi delle colonne nella tabella di origine, ma i tipi di dati devono corrispondere.
<i>colonne calcolate utente</i>	Colonne definite dall'utente derivate da espressioni SQL. È possibile utilizzare colonne calcolate con funzioni SQL per convertire i tipi di dati di origine in tipi di dati di destinazione differenti.



---

## Appendice A. Gli schemi di codifica UNICODE e ASCII per la replica SQL (z/OS)

z/OS

La replica SQL per OS/390 e z/OS Versione 7 o successiva supporta gli schemi di codifica UNICODE e ASCII.

Per sfruttare lo schema di codifica UNICODE, è necessario disporre almeno di DB2 per OS/390 e di z/OS Versione 7. Inoltre, è necessario creare manualmente o convertire l'origine, la destinazione e le tabelle di controllo della replica SQL, così come descritto nelle sezioni seguenti. Ciononostante, l'ambiente di replica esistente funzionerà con la replica SQL per OS/390 e z/OS Versione 7 o successiva, anche se l'utente non modificherà alcuno schema di codifica. Se il proprio sistema è UNICODE, è necessario aggiungere i comandi ENCODING(EBCDIC) su BIND PLAN e PACKAGE per i programmi Capture, Apply e Replication Alert Monitor.

---

### Regole per la scelta di uno schema di codifica

z/OS

Se l'origine, il CD e le tabelle di destinazione di cui si dispone utilizzano lo stesso schema di codifica, è possibile ridurre al minimo la necessità di conversioni di dati nell'ambiente di replica.

Quando vengono scelti gli schemi di codifica per le tabelle, attenersi alla regola del CCSID singolo:

I dati dello spazio tabella sono stati codificati mediante CCSID ASCII, EBCDIC o UNICODE. Lo schema di codifica di tutte le tabelle alle quali si fa riferimento mediante un'istruzione SQL dev'essere lo stesso. Inoltre, tutte le tabelle utilizzate in viste e unioni devono utilizzare lo stesso schema di codifica.

Se non viene seguita la regola del CCSID singolo, DB2 rileverà tale violazione e restituirà SQLCODE -873 durante il bind o l'esecuzione.

La scelta di ASCII o UNICODE per le tabelle dipenderà dalla configurazione client/server. Attenersi a queste regole in special modo quando vengono scelti schemi di codifica per le tabelle:

- Le tabelle di origine o destinazione in DB2 per OS/390 possono essere EBCDIC, ASCII o UNICODE. Queste ultime possono essere copiate da o in tabelle con la stesso schema di codifica, o con uno schema differente, in qualsiasi DBMS supportato (famiglia DB2 o non DB2, con DataJoiner).
- Su un server di origine DB2 per OS/390, le tabelle CD e UOW nello stesso server non devono utilizzare lo stesso schema di codifica se, al momento della creazione del membro del set di sottoscrizioni, il tipo di destinazione è USERCOPY e se JOIN\_UOW\_CD non equivale a Y. Altrimenti, le tabelle CD e UOW devono utilizzare lo stesso schema di codifica.
- La tabella IBMSNAP\_SIGNAL deve disporre della codifica EBCDIC, affinché il programma Capture non debba tradurre i segnali in EBCDIC quando provvede a selezionarli dalla tabella dei segnali.

- Tutte le tabelle di controllo (ASN.IBMSNAP\_SUBS\_XXXX) sullo stesso server di controllo devono utilizzare lo stesso schema di codifica.
- Le altre tabelle di controllo possono utilizzare uno schema di codifica qualsiasi.

---

## Impostazione degli schemi di codifica

z/OS

Per specificare lo schema di codifica appropriato per le tabelle, modificare l'SQL utilizzato per la generazione delle tabelle.

Si consiglia di arrestare i programmi Capture e Apply prima di modificare lo schema di codifica delle tabelle esistenti.

**Nota:** I Riferimenti SQL DB2 per z/OS V8 contengono ulteriori informazioni su CCSID.

Per impostare gli schemi di codifica:

1. Creare nuove tabelle di origine e destinazione con lo schema di codifica adeguato. Dopodiché, si consiglia di reinizializzare Capture mediante avvio a freddo e riavviare il programma Apply.
2. Se le tabelle di origine e destinazione sono già state create, modificare gli schemi di codifica delle tabelle di origine e destinazione esistenti. Le tabelle esistenti devono disporre dello stesso schema di codifica all'interno di uno spazio tabella
  - a. Utilizzare l'utilità Reorg Tablespace per annullare il caricamento dello spazio tabella esistente.
  - b. Eliminare lo spazio tabella esistente.
  - c. Ricreare lo spazio tabella specificando il nuovo schema di codifica.
  - d. Utilizzare l'utilità Load per il caricamento dei dati precedenti nel nuovo spazio tabella. Consultare il manuale *DB2 for z/OS V8 Utility Guide and Reference* per ulteriori informazioni sulle utilità Load e Reorg.
3. Utilizzare il Replication Center per la creazione di nuove tabelle di controllo, con lo schema di codifica adeguato.
4. Utilizzare le utilità Reorg e Load per modificare lo schema di codifica per le tabelle di controllo esistenti e le tabelle CD.
5. Quando si creano nuove origini per la replica o nuovi set di sottoscrizioni mediante ASNCLP o il Replication Center, specificare lo schema di codifica adeguato.

---

## Appendice B. Avvio dei programmi di replica SQL dall'interno di un'applicazione (Linux, UNIX, Windows)

### Linux UNIX Windows

È possibile avviare qualsiasi programma di replica (programma Capture, programma Apply, Replication Alert Monitor) per un ciclo di replica dall'interno dell'applicazione richiamando routine.

Per utilizzare queste routine è necessario specificare l'opzione AUTOSTOP per il programma Capture e l'opzione COPYONCE per il programma Apply in quanto l'API supporta solo l'esecuzione sincrona.

Esempi di API e dei rispettivi makefile sono presenti nelle seguenti directory:

### Windows

sqllib\samples\repl

### Linux UNIX

sqllib/samples/repl

Tali directory contengono i seguenti file per l'avvio del programma Capture:

#### **capture\_api.c**

Codice di esempio per l'avvio del programma Capture su Windows, Linux o UNIX.

#### **capture\_api\_nt.mak**

Makefile per il codice di esempio su Windows.

#### **capture\_api\_unix.mak**

Makefile per il codice di esempio su UNIX.

Tali directory contengono i seguenti file per l'avvio del programma Apply:

#### **apply\_api.c**

Codice di esempio per l'avvio del programma Apply su Windows, Linux o UNIX.

#### **apply\_api\_nt.mak**

Makefile per il codice di esempio su Windows.

#### **apply\_api\_unix.mak**

Makefile per il codice di esempio su UNIX.

Tali directory contengono i seguenti file per l'avvio di Replication Alert Monitor:

#### **monitor\_api.c**

Codice di esempio per l'avvio di Replication Alert Monitor su Windows, Linux o UNIX.

#### **monitor\_api\_nt.mak**

Makefile per il codice di esempio su Windows.

#### **monitor\_api\_unix.mak**

Makefile per il codice di esempio su UNIX.



## Appendice C. How the Capture program processes journal entry types for SQL replication (System i)

### System i

The following table describes how the Capture program processes different journal entry types.

Tabella 123. Capture program processing by journal entry

Journal code <sup>1</sup>	Entry type	Description	Processing
C	CM	Set of record changes committed	Insert a record in the UOW stable.
C	RB	Rollback	No UOW row inserted.
F	AY	Journalized changes applied to physical file member	Issue an ASN2004 message and full refresh of file.
F	CE	Change end of data for physical file	Issue an ASN2004 message and full refresh of file.
F	CR	Physical file member cleared	Issue an ASN2004 message and full refresh of file.
F	EJ	Journaling for physical file member ended	Issue an ASN200A message and full refresh of the file. A full-refresh occurs whenever the Capture program reads an EJ journal entry, regardless of whether the user or the system caused journaling to end. Refer to the appropriate System i documentation for information about implicit end-journal events for a file.
F	IZ	Physical file member initialized	Issue an ASN2004 message and full refresh of file.
F	MD	Member removed from physical file (DLTLIB, DLTF, or RMVM)	Issue an ASN200A message and attempt a full refresh.
F	MF	Storage for physical file member freed	Issue an ASN200A message and full refresh of file.
F	MM	Physical file containing member moved (Rename Object (RNMOBJ) of library, Move Object (MOVOBJ) of file)	Issue an ASN200A message and attempt a full refresh.
F	MN	Physical file containing member renamed (RNMOBJ of file, Rename Member (RNMM))	Issue an ASN200A message and attempt a full refresh.
F	MR	Physical file member restored	Issue an ASN2004 message and full refresh of file.
F	RC	Journalized changes removed from physical file member	Issue an ASN2004 message and full refresh of file.

Tabella 123. Capture program processing by journal entry (Continua)

Journal code <sup>1</sup>	Entry type	Description	Processing
F	RG	Physical file member reorganized	If the RRN of the source table is being used as the replication key, issue an ASN2004 message and full refresh of file.
J	NR	Identifier for next journal receivers	Reset the Capture program.
J	PR	Identifier for previous journal receivers	Increment the unique sequence number counter.
R	DL	Record deleted from physical file member	Insert a DLT record in the CD table.
R	DR	Record deleted for rollback	Insert a DLT record in the CD table.
R	PT	Record added to physical file member	Insert an ADD record in the CD table.
R	PX	Record added directly to physical file member	Insert an ADD record in the CD table.
R	UB	Before-image of record updated in physical file member	See note 2.
R	UP	After-image of record updated in physical file member	See note 2.
R	BR	Before-image of record updated for rollback	See note 3.
R	UR	After-image of record updated for rollback	See note 3.

**Nota:**

- The following values are used for the journal codes:  
**C** Commitment control operation  
**F** Database file operation  
**J** Journal or journal receiver operation  
**R** Operation on specific record
- The R-UP image and the R-UB image form a single UPD record in the CD table if the PARTITION\_KEYS\_CHG column in the register table is N. Otherwise, the R-UB image inserts a DLT record in the CD table and the R-UP image inserts an ADD record in the CD table.
- The R-UR image and the R-BR image form a single UPD record in the CD table if the PARTITION\_KEYS\_CHG column in the register table is N. Otherwise, the R-BR image inserts a DLT record in the CD table and the R-UR image inserts an ADD record in the CD table.

All other journal entry types are ignored by the Capture program.



---

## Documentazione del prodotto

La documentazione viene fornita in varie ubicazioni e in diversi formati, incluso nella guida che è possibile aprire direttamente dall'interfaccia del prodotto, in un centro informazioni a livello della suite e in pubblicazioni in file PDF.

È possibile inoltre ordinare le pubblicazioni IBM in formato cartaceo in linea o mediante il proprio rappresentante IBM.

Per ordinare le pubblicazioni in linea, consultare l'IBM Publications Center all'indirizzo [www.ibm.com/shop/publications/order](http://www.ibm.com/shop/publications/order).

È possibile inviare i propri commenti relativi alla documentazione utilizzando una delle seguenti modalità:

- Modulo in linea per i commenti dei lettori: [www.ibm.com/software/data/rcf/](http://www.ibm.com/software/data/rcf/)
- E-mail: [comments@us.ibm.com](mailto:comments@us.ibm.com)

---

## Come contattare IBM

È possibile contattare l'IBM per ottenere assistenza clienti, servizi software, informazioni sui prodotti e informazioni generali. È possibile anche fornire commenti sui prodotti e sulla documentazione.

### Assistenza clienti

Per accedere all'assistenza clienti per i prodotti IBM e per ottenere informazioni sul download dei prodotti, visitare il sito di supporto e download all'indirizzo [www.ibm.com/support/us/](http://www.ibm.com/support/us/).

È possibile richiedere assistenza visitando il sito di richiesta di servizi del supporto software all'indirizzo [www.ibm.com/software/support/probsub.html](http://www.ibm.com/software/support/probsub.html).

### La mia IBM

È possibile gestire link a siti Web e informazioni IBM che soddisfano specifiche necessità di supporto tecnico degli utenti creando un account al sito "La mia IBM" all'indirizzo [www.ibm.com/account/us/](http://www.ibm.com/account/us/).

### Servizi software

Per informazioni relative a software, IT e servizi di consulenza aziendale, visitare il sito di soluzioni all'indirizzo [www.ibm.com/businesssolutions/us/en](http://www.ibm.com/businesssolutions/us/en).

### Supporto al prodotto Information Management

Per ottenere supporto, notizie e altre informazioni su Information Management, visitare il sito all'indirizzo [www.ibm.com/software/data/support/](http://www.ibm.com/software/data/support/).

### Supporto per prodotti di replica, pubblicazioni eventi e Federation

Per ottenere supporto, visitare:

- IBM InfoSphere Federation Server  
[www.ibm.com/software/data/integration/support/federation\\_server/](http://www.ibm.com/software/data/integration/support/federation_server/)
- IBM InfoSphere Replication Server  
[www.ibm.com/software/data/integration/support/replication\\_server/](http://www.ibm.com/software/data/integration/support/replication_server/)
- IBM InfoSphere Data Event Publisher  
[www.ibm.com/software/data/integration/support/data\\_event\\_publisher/](http://www.ibm.com/software/data/integration/support/data_event_publisher/)

## Supporto per i prodotti Classic

Per ottenere supporto, visitare:

- IBM InfoSphere Classic Federation Server for z/OS  
[www.ibm.com/software/data/integration/support/classic\\_federation\\_server\\_z/](http://www.ibm.com/software/data/integration/support/classic_federation_server_z/)
- IBM InfoSphere Classic Replication Server for z/OS  
[www.ibm.com/software/data/infosphere/support/replication-server-z/](http://www.ibm.com/software/data/infosphere/support/replication-server-z/)
- IBM InfoSphere Classic Data Event Publisher for z/OS  
[www.ibm.com/software/data/integration/support/data\\_event\\_publisher\\_z/](http://www.ibm.com/software/data/integration/support/data_event_publisher_z/)
- IBM InfoSphere Data Integration Classic Connector for z/OS  
[www.ibm.com/software/data/integration/support/data\\_integration\\_classic\\_connector\\_z/](http://www.ibm.com/software/data/integration/support/data_integration_classic_connector_z/)

## Informazioni generali

Per informazioni generali su IBM, visitare il sito [www.ibm.com](http://www.ibm.com).

## Commenti sul prodotto

È possibile fornire commenti generali sui prodotti tramite il Consumability Survey all'indirizzo [www.ibm.com/software/data/info/consumability-survey](http://www.ibm.com/software/data/info/consumability-survey).

## Commenti sulla documentazione

Per lasciare un commento, è possibile selezionare il link dei commenti in qualsiasi argomento del centro informazioni.

È possibile anche inviare commenti relativi alle pubblicazioni in file PDF, al centro informazioni o a qualsiasi altra documentazione utilizzando una delle seguenti modalità:

- Modulo in linea per i commenti dei lettori: [www.ibm.com/software/data/rcf/](http://www.ibm.com/software/data/rcf/)
- E-mail: [comments@us.ibm.com](mailto:comments@us.ibm.com)

---

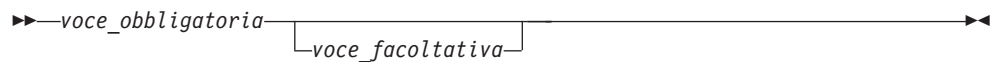
## Come leggere i diagrammi di sintassi

Le seguenti regole vengono applicate ai diagrammi di sintassi utilizzati in queste informazioni:

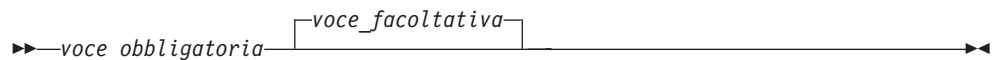
- Leggere i diagrammi di sintassi da sinistra a destra, dall'alto in basso, seguendo il percorso della riga. Vengono utilizzate le seguenti convenzioni:
  - Il simbolo >>--- indica l'inizio di un diagramma di sintassi.
  - Il simbolo ---> indica che il diagramma di sintassi continua sulla riga successiva.
  - Il simbolo >--- indica che un diagramma di sintassi continua dalla riga precedente.
  - Il simbolo --->< indica la fine di un diagramma di sintassi.
- Le voci obbligatorie vengono visualizzate sulla riga orizzontale (il percorso principale).



- Le voci facoltative vengono visualizzate sotto il percorso principale.



Se una voce facoltativa viene visualizzata sopra il percorso principale, tale voce non ha alcun effetto sull'esecuzione dell'elemento della sintassi e viene utilizzata solo per consentirne la leggibilità.

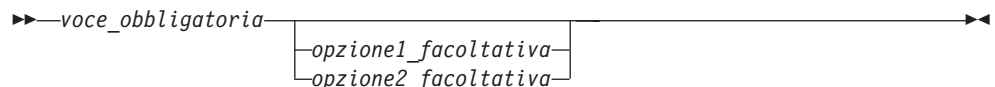


- Se è possibile scegliere fra due o più voci, queste vengono visualizzate verticalmente, in uno stack.

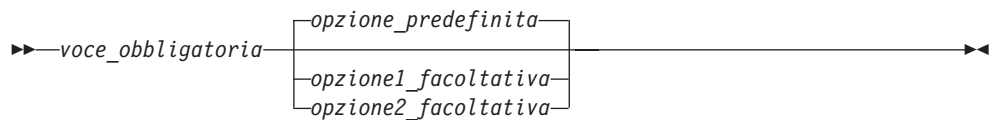
Se è necessario scegliere una determinata voce dello stack, tale voce viene visualizzata sul percorso principale.



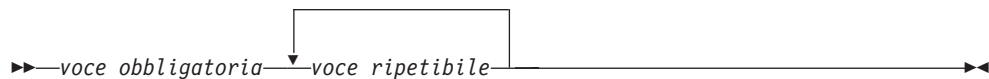
Se la scelta di una voce è facoltativa, l'intero stack viene visualizzato sotto il percorso principale.



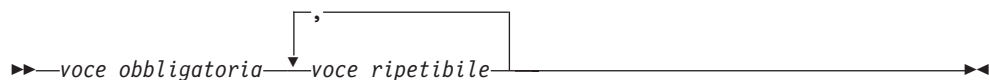
Se una delle voci rappresenta quella predefinita, questa viene visualizzata sopra il percorso principale e le restanti opzioni vengono mostrate in basso.



- Una freccia rivolta verso sinistra, sopra la riga principale, indica una voce che può essere ripetuta.

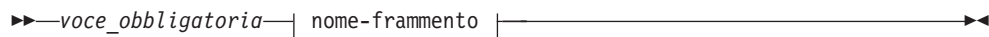


Se la freccia contiene una virgola, è necessario separare le voci ripetute con la virgola.

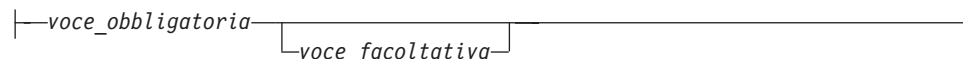


Una freccia di ripetizione posta al di sopra di uno stack indica che è possibile ripetere le voci nello stack.

- A volte è necessario dividere un diagramma in frammenti. Il frammento di sintassi viene mostrato separatamente dal diagramma principale, ma i contenuti di tale frammento devono essere letti come se fossero scritti sul percorso principale del diagramma.



**Nome-frammento:**



- Le parole chiave e le relative abbreviazioni minime, se applicabili, vengono visualizzate in maiuscolo. Devono essere scritte esattamente come mostrato.
- Le variabili vengono visualizzate in corsivo minuscolo (ad esempio, nome-colonna). Rappresentano i nomi o i valori forniti dall'utente.
- Se nel diagramma non viene mostrata alcuna punteggiatura intermedia, separare le parole chiave e i parametri con almeno uno spazio.
- Immettere i segni di punteggiatura, parentesi, operatori matematici o altri simboli simili esattamente come mostrato nel diagramma.
- Le note in calce vengono mostrate con un numero tra parentesi, ad esempio (1).

---

## Accessibilità dei prodotti

È possibile ottenere informazioni relative allo stato di accessibilità dei prodotti IBM.

I moduli dei prodotti e le interfacce utente IBM Information Server non sono totalmente accessibili. Il programma di installazione consente di installare i seguenti moduli e componenti dei prodotti:

- WebSphere Business Glossary
- Information Server Business Glossary Anywhere
- IBM WebSphere DataStage and QualityStage
- IBM Information Server FastTrack
- WebSphere Information Analyzer
- IBM WebSphere Information Services Director
- IBM Metadata Workbench

Per ulteriori informazioni relative allo stato di accessibilità dei prodotti IBM, visitare il sito [http://www.ibm.com/able/product\\_accessibility/index.html](http://www.ibm.com/able/product_accessibility/index.html).

### Documentazione accessibile

La documentazione accessibile per i prodotti di IBM Information Server viene fornita in un centro informazioni. Il centro informazioni presenta la documentazione in formato XHTML 1.0, visualizzabile nella maggior parte dei browser Web. XHTML consente di impostare le preferenze di visualizzazione nel proprio browser, nonché di utilizzare utilità per la lettura dello schermo e altre tecnologie di assistenza per accedere alla documentazione.



---

## Informazioni particolari

Queste informazioni sono state sviluppate per prodotti e servizi offerti negli Stati Uniti.

IBM può non offrire i prodotti, i servizi o le funzioni presentati in questo documento in altri paesi. Consultare il proprio rappresentante locale IBM per informazioni sui prodotti ed i servizi attualmente disponibili nella propria zona. Qualsiasi riferimento ad un prodotto, programma o servizio IBM non implica o intende dichiarare che solo quel prodotto, programma o servizio IBM può essere utilizzato. Qualsiasi prodotto funzionalmente equivalente al prodotto, programma o servizio che non violi alcun diritto di proprietà intellettuale IBM può essere utilizzato. Tuttavia, è responsabilità dell'utente valutare e verificare il funzionamento di qualsiasi prodotto, programma o servizio non IBM.

IBM può avere applicazioni di brevetti o brevetti in corso relativi all'argomento descritto in questo documento. La fornitura del presente documento non concede alcuna licenza a tali brevetti. È possibile inviare per iscritto richieste di licenze a:

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785 U.S.A.

Per richieste di licenze relative ad informazioni double-byte (DBCS), contattare il Dipartimento di Proprietà Intellettuale IBM nel proprio paese o inviare richieste per iscritto a:

Intellectual Property Licensing  
Legal and Intellectual Property Law  
IBM Japan, Ltd.  
3-2-12, Roppongi, Minato-ku, Tokyo 106-8711 Japan

**Il seguente paragrafo non si applica al Regno Unito o a qualunque altro paese in cui tali dichiarazioni sono incompatibili con le norme locali:** IBM (INTERNATIONAL BUSINESS MACHINES CORPORATION) FORNISCE LA PRESENTE PUBBLICAZIONE "NELLO STATO IN CUI SI TROVA" SENZA GARANZIE DI ALCUN TIPO, ESPRESSE O IMPLICITE, IVI INCLUSE, A TITOLO DI ESEMPIO, GARANZIE IMPLICITE DI NON VIOLAZIONE, DI COMMERCIALIZZABILITÀ E DI IDONEITÀ PER UNO SCOPO PARTICOLARE. Alcuni stati non consentono la rinuncia ad alcune garanzie espresse o implicite in determinate transazioni, pertanto, la presente dichiarazione può non essere applicabile.

Queste informazioni potrebbero includere inesattezze tecniche o errori tipografici. Le modifiche alle presenti informazioni vengono effettuate periodicamente; tali modifiche saranno incorporate nelle nuove pubblicazioni della pubblicazione. IBM può effettuare miglioramenti e/o modifiche ai prodotti e/o ai programmi descritti nella presente pubblicazione in qualsiasi momento senza preavviso.

Qualsiasi riferimento in queste informazioni a siti Web non IBM sono fornite solo per convenienza e non servono in alcun modo da approvazione di tali siti Web. I

materiali presenti in tali siti Web non sono parte dei materiali per questo prodotto IBM e l'utilizzo di tali siti Web è a proprio rischio.

IBM può utilizzare o distribuire qualsiasi informazione fornita in qualsiasi modo ritenga appropriato senza incorrere in alcun obbligo verso l'utente.

I licenziatari di questo programma che desiderano avere informazioni allo scopo di abilitare: (i) lo scambio di informazioni tra i programmi creati indipendentemente e gli altri programmi (incluso il presente) e (ii) il reciproco utilizzo di informazioni che sono state scambiate, dovrebbero contattare:

IBM Corporation  
J46A/G4  
555 Bailey Avenue  
San Jose, CA 95141-1003 U.S.A.

Tali informazioni possono essere disponibili, in base ad appropriate clausole e condizioni, includendo in alcuni casi, il pagamento di una tassa.

Il programma concesso in licenza descritto nel presente documento e tutto il materiale concesso in licenza disponibile sono forniti da IBM in base alle clausole dell'Accordo per Clienti IBM (IBM Customer Agreement), dell'IBM IPLA (IBM International Program License Agreement) o qualsiasi altro accordo equivalente tra le parti.

Qualsiasi dato sulle prestazioni qui contenuto è stato determinato in un ambiente controllato. Pertanto, i risultati ottenuti in altri ambienti operativi possono notevolmente variare. Alcune misurazioni possono essere state effettuate sui sistemi del livello di sviluppo e non vi è alcuna garanzia che tali misurazioni resteranno invariate sui sistemi generalmente disponibili. Inoltre, alcune misurazioni possono essere state stimate tramite estrapolazione. I risultati reali possono variare. Gli utenti del presente documento dovranno verificare i dati applicabili per i propri ambienti specifici.

Le informazioni relative a prodotti non IBM sono ottenute dai fornitori di quei prodotti, dagli annunci pubblicati e da altre fonti disponibili al pubblico. IBM non ha testato quei prodotti e non può confermarne l'accuratezza della prestazione, la compatibilità o qualsiasi altro reclamo relativo ai prodotti non IBM. Le domande sulle capacità dei prodotti non IBM dovranno essere indirizzate ai fornitori di tali prodotti.

Tutte le dichiarazioni relative all'orientamento o alle intenzioni future di IBM sono soggette a modifica o a ritiro senza preavviso e rappresentano solo mete e obiettivi.

Queste informazioni sono solo per scopi di pianificazione. Le presenti informazioni sono soggette a modifiche prima che i prodotti descritti siano resi disponibili.

Queste informazioni contengono esempi di dati e report utilizzati in quotidiane operazioni aziendali. Per illustrarle nel modo più completo possibile, gli esempi includono i nomi di individui, società, marchi e prodotti. Tutti questi nomi sono fittizi e qualsiasi somiglianza con nomi ed indirizzi utilizzati da gruppi aziendali realmente esistenti è puramente casuale.

LICENZA SUL DIRITTO D'AUTORE:



Queste informazioni contengono programmi applicativi di esempio in linguaggio sorgente, che illustrano tecniche di programmazione su varie piattaforme operative. È possibile copiare, modificare e distribuire questi programmi di esempio sotto qualsiasi forma senza alcun pagamento alla IBM, allo scopo di sviluppare, utilizzare, commercializzare o distribuire i programmi applicativi in conformità alle API (application programming interface) a seconda della piattaforma operativa per cui i programmi di esempio sono stati scritti. Questi esempi non sono stati testati approfonditamente tenendo conto di tutte le condizioni possibili. La IBM, quindi, non può garantire o sottintendere l'affidabilità, l'utilità o il funzionamento di questi programmi. I programmi di esempio sono forniti "NELLO STATO IN CUI SI TROVANO", senza alcun tipo di garanzia. IBM non intende essere responsabile per alcun danno derivante dal vostro uso dei programmi di esempio.

Ogni copia o qualsiasi parte di questi programmi di esempio o qualsiasi lavoro derivato, devono contenere le seguenti informazioni relative alle leggi sul diritto d'autore:

© (nome della propria azienda) (anno). Parti di questo codice derivano dai Programmi di Esempio della IBM Corp. © Tutelato dalle leggi sul diritto d'autore IBM Corp. \_immettere l'anno o gli anni\_. Tutti i diritti riservati.

Se si visualizzano tali informazioni come softcopy, non potranno apparire le fotografie e le illustrazioni a colori.

---

## Marchi

I marchi IBM e alcuni marchi non IBM sono contrassegnati alla prima occorrenza in questo documento con il simbolo appropriato.

IBM, IBM logo e ibm.com sono marchi di International Business Machines Corp., registrati in molte giurisdizioni in tutto il mondo. Nomi di altri prodotti o servizi possono essere marchi di IBM o altre società. Un elenco aggiornato di marchi IBM è disponibile alla pagina Web "Informazioni su copyright e marchi registrati" all'indirizzo [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml).

I seguenti termini sono marchi di altre società:

Adobe, Adobe logo, PostScript e PostScript logo sono marchi di Adobe Systems Incorporated negli Stati Uniti e/o in altri paesi.

IT Infrastructure Library è un marchio della Central Computer and Telecommunications Agency, che è ora parte dell'Office of Government Commerce.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, e Pentium sono marchi di Intel Corporation o sue affiliate negli Stati Uniti e in altri paesi.

Linux è un marchio registrato di Linus Torvalds negli Stati Uniti e/o in altri paesi.

Microsoft, Windows, Windows NT e Windows logo sono marchi di Microsoft Corporation negli Stati Uniti e/o in altri paesi.

ITIL è un marchio e un marchio comunitario registrato dell'Office of Government Commerce ed è registrato presso l'Ufficio brevetti e marchi degli Stati Uniti.

UNIX è un marchio registrato di The Open Group negli Stati Uniti e/o in altri paesi.

Cell Broadband Engine è un marchio di Sony Computer Entertainment, Inc. negli Stati Uniti e/o in altri paesi ed è utilizzato in base a licenza.

Java e tutti i marchi basati su Java, sono marchi di Sun Microsystems, Inc., negli Stati Uniti e/o in altri paesi.

United States Postal Service è proprietario dei seguenti marchi: CASS, CASS Certified, DPV, LACSLink, ZIP, ZIP + 4, ZIP Code, Post Office, Postal Service, USPS e United States Postal Service. IBM Corporation è una concessionaria di licenze DPV e LACSLink non esclusiva di United States Postal Service.

Nomi di altre società, prodotti o servizi possono essere marchi di altre società.

# Indice analitico

## A

accessibilità 511  
accessibilità dei prodotti  
  accessibilità 515  
ADDDPRREG command 321  
ADDDPRSUB command 329  
ADDDPRSUBM command 344  
ADDJOBSCDE command 244  
aggiornamenti  
  come eliminazioni e inserimenti 49  
  conflitti 54  
alert conditions  
  ASNMAIL exit routine 222  
  e-mail notification 220  
  for Apply program 217  
  for Capture program 217  
  for Q Apply program 217  
  for Q Capture program 217  
  list 217  
  overview 217  
alert\_prune\_limit parameter, Replication  
  Alert Monitor 233  
alerts  
  sending to z/OS console 222  
ALWINACT parameter 391  
ambienti di replica  
  copia 195  
Analyzer  
  for System i  
    creating SQL packages 30  
    invocation parameters 355  
Analyzer report  
  ANZDPR command 354  
ANZDPR command 354  
ANZDPRJRN command 37  
applicazioni  
  avvio programmi di replica da 507  
Apply control tables  
  IBMSNAP\_APPLY\_JOB 454  
Apply program  
  alert conditions 217  
  for System i  
    ALWINACT parameter 391  
    APYQUAL parameter 390  
    checking status 257  
    COPYONCE parameter 392  
    creating SQL packages 30  
    CTLSVR parameter 390  
    DELAY parameter 392  
    FULLREFPGM parameter 391  
    INACTMSG parameter 391  
    JOB parameter 389  
    OPTSNGSET parameter 393  
    RTYWAIT parameter 392  
    scheduling 244  
    setting up 33  
    starting 133, 388  
    stopping 363  
    SUBNFYPM parameter 391  
    TRACE parameter 390  
    TRLREUSE parameter 392  
Apply program (*Continua*)  
  for System i (*Continua*)  
    USER parameter 389  
Apply qualifiers  
  use when starting the Apply  
    program 133  
APYQUAL parameter 390  
ARM (Automatic Restart Manager) 162  
arresto  
  Programma Apply  
    per System i 145  
    per UNIX 145  
    per Windows 145  
    per z/OS 145  
  Programma Capture  
    per System i 127  
    per UNIX 127  
    per Windows 127  
    per z/OS 127  
arresto cattura modifiche 168  
ASNDONE exit routine  
  using 147  
ASNLOAD exit routine  
  for System i 154  
ASNMAIL exit routine 222  
asnmcmd command 290  
asnpwd 293  
asnpwd command 17  
asnsrct 297  
asnsdrop 300  
asnslist command 301  
asntdiff command 302, 307  
asntdiff utility  
  overview 209  
asntrc 310  
asntrep command 317  
asntrep utility  
  usage guide 214  
asntrepair utility  
  usage guide 209  
asntrepair utility  
  overview 209  
assistenza, clienti 511  
assistenza clienti 511  
associazione  
  colonne di origine alle colonne di  
    destinazione 91  
  tipi di dati tra le tabelle 91  
  tra origini e destinazioni 74  
attivazione delle serie di richieste 68  
attività avviate dal sistema 157  
attributi  
  modifica degli oggetti registrati 166  
  modifica delle serie di richieste 175  
authorization  
  for Replication Alert Monitor 224  
autoprunes parameter, Replication Alert  
  Monitor 233  
autorizzazione  
  per i trigger Capture 17  
  per il programma Apply 15

autorizzazione (*Continua*)  
  per il programma Capture 14  
  per la gestione 13  
avvio  
  Controllo degli avvisi di replica  
    per UNIX 507  
    per Windows 507  
  Programma Apply  
    per UNIX 131, 507  
    per Windows 131, 507  
    per z/OS 131, 159  
  Programma Capture  
    per UNIX 109, 507  
    per Windows 109, 507  
    per z/OS 109  
avvio a caldo, programma Capture  
  per UNIX 114  
  per Windows 114  
  per z/OS 114  
avvio a freddo, programma Capture  
  impedire 206  
  per UNIX 114  
  per Windows 114  
  per z/OS 114

## B

bind  
  Programma Apply  
    per Linux 29  
    per UNIX 29  
    per Windows 29  
  Programma Capture  
    per Linux 28  
    per UNIX 28  
    per Windows 28  
binding  
  Replication Alert Monitor  
    for Linux 224  
    for UNIX 224  
    for Windows 224  
BLOB (binary large object)  
  considerazioni sulle repliche 96  
blocchi  
  sulle tabelle CCD 10  
blocco dei dati 68

## C

CAPCTLLIB parameter 398  
Capture  
  più partizioni del database 31  
  utilizzo di più partizioni del  
    database 26  
Capture control tables  
  IBMSNAP\_AUTHTKN 416  
  IBMSNAP\_REG\_EXT 434  
Capture program  
  alert conditions 217

- Capture program (*Continua*)
  - for System i
    - CAPCTLLIB parameter 398
    - changing attributes 357
    - checking status 257
    - CLNUPITV parameter 397
    - cold start parameters 396
    - creating SQL packages 29
    - FRCFRQ parameter 400
    - JOB parameter 396
    - journal entry types 509
    - journals and journal receivers, managing 35
    - JRN parameter 398
    - LAG parameter 400
    - MEMLMT parameter 399
    - MONITV parameter 399
    - MONLMT parameter 398
    - overriding attributes of 377
    - progress of 257
    - reinitializing 376
    - RESTART parameter 396
    - RETAIN parameter 399
    - scheduling 244
    - setting up 33
    - starting 112, 395
    - stopping 366
    - TRCLMT parameter 398
    - WAIT parameter 397
    - warm start parameters 396
- caratteri finali, in script SQL generati 259
- Centro di replica
  - comunicazioni con
    - Programma Apply 245
    - programma Capture 245
    - Replication Alert Monitor 249
    - trigger Capture 245
  - connettività 21
  - funzioni di promozione 195
- changing Capture parameters for System i 357
- CHGDPRCAPA command 357
- CHGJRN command 36
- chiavi di destinazione 92
- chiavi di partizionamento logico
  - descrizione 49
- chiavi primarie
  - partizionamento logico 49
  - utilizzato come chiave di destinazione 92
- ciclo di richieste 68
- clausola WHERE
  - restrizioni della colonna PREDICATES 103
  - serie secondarie di righe 90
- CLNUPITV parameter 397
- CLOB (character large object)
  - considerazioni sulle repliche 96
- coerenza dei dati 87
- cold start, Capture program for System i 396
- cold startmode 114
- colonna JOIN\_UOW\_CD 103
- colonna PREDICATES 103
- colonna UOW\_CD\_PREDICATES 103
- colonne
  - aggiunta a tabelle di origine registrata 166
  - associazione dalle origini alle destinazioni 91
  - calcolate 90, 107
  - creazione di serie secondarie nella destinazione 90
  - sull'origine 43
  - definizione nella tabella di destinazione 90
  - disponibile per la replica 43
  - post-immagine 45
  - pre-immagine 45
  - registrazione nella tabella di origine 43
  - ridenominazione 91, 107
- colonne calcolate 90
- creazione 107
- Tabella CD 79
- tabella di origine 79
- colonne chiave di destinazione
  - aggiornamento 94
- colonne di chiavi primarie aggiornate 49
- colonne GENERATED ALWAYS 99
- colonne IDENTITY 99
- colonne post-immagine 45
- colonne pre-immagine
  - limitazioni 45
  - registrazione 45
  - tabelle di modifica/agggregazione 90
- columns
  - relative record numbers on System i 57
- comandi
  - asncmd 281
  - asnapply 275
  - asncap 263
  - asnccmd 271
- comandi di replica
  - \$TA JES2
    - Apply per z/OS 244
    - Capture per z/OS 244
  - AT 243, 244
  - AT NetView
    - Apply per z/OS 244
    - Capture per z/OS 244
  - backup database 27
  - configurazione aggiornamento database 27
  - per System i
    - ENDDPRCAP 127
    - STRDPRAPY 134
  - per UNIX
    - asnanalyze 282
  - per Windows
    - asnanalyze 282
  - per z/OS
    - MODIFY 157
- comandi di sistema
  - asncmd 281
  - asnapply 275
  - asncap 263
  - asnccmd 271
- comando \$TA JES2 244
- comando asncmd 281
- comando asnanalyze 282
- comando asnapply 275
- comando asncap 263
- comando AT
  - Controllo degli avvisi di replica 243, 244
  - Programma Apply 243, 244
  - Programma Capture 243, 244
  - Replication Alert Monitor 243
- comando AT NetView
  - Apply per z/OS 244
  - Capture per z/OS 244
- comando backup database 27
- comando configurazione aggiornamento database 27
- comando ENDDPRCAP 127
- comando MODIFY 157, 159
- comando STRDPRAPY 134
- componente replica
  - SQLcomunicazione 245
- comunicazione tra componenti replica SQL 245
- configurazione
  - connettività 21
    - Programma Apply per Linux 29
    - per UNIX 29
    - per Windows 29
  - Programma Capture per UNIX 27
  - per Windows 27
- configurazione della replica con tre livelli 84
- configuring
  - Replication Alert Monitor
    - for Linux 224
    - for UNIX 224
    - for Windows 224
- conflitti
  - impedire 8
- connecting
  - to System i server 21
- connectivity
  - between DB2 operating systems 21
- connettività
  - fra i sistemi operativi DB2 21
  - recupero malfunzionamento per le tabelle di controllo 206
- connettività rete 21
- console MVS 157
- Console MVS 159
- contact groups 215
- contacts
  - description 215
- contacts and contact groups, defining 226
- contacts and contact groups for monitoring, defining 226
- control tables
  - authorization requirements for System i 33
  - creating
    - in IASP groups 24
    - on System i 24, 362
  - granting authority for System i 368
  - IBMSNAP\_APPLY\_JOB 454
  - IBMSNAP\_AUTHTKN 416

control tables (*Continua*)  
 IBMSNAP\_REG\_EXT 434  
 Monitor control server  
 IBMSNAP\_ALERTS 481  
 IBMSNAP\_CONDITIONS 483  
 IBMSNAP\_CONTACTGRP 488  
 IBMSNAP\_CONTACTS 489  
 IBMSNAP\_GROUPS 490  
 IBMSNAP\_MONENQ 490  
 IBMSNAP\_MONPARMS 490  
 IBMSNAP\_MONSERVERS 493  
 IBMSNAP\_MONTRACE 494  
 IBMSNAP\_MONTRAIL 494  
 IBMSNAP\_SUSPENDS 496  
 IBMSNAP\_TEMPLATES 497  
 list 481  
 quick reference  
 at a glance 407  
 Replication Alert Monitor,  
 creating 225  
 revoking authority for System i 386  
 controllo  
 avvio a freddo 80, 499  
 dati di origine 45  
 spazi nei dati 80, 499  
 tendenze cronologiche 252  
 Controllo degli avvisi di replica  
 comunicazioni con  
 Capture 249  
 programma Apply 249  
 per UNIX  
 avvio 507  
 verifica dello stato 251  
 per Windows  
 avvio 507  
 per z/OS  
 verifica dello stato 251  
 pianificazione 243, 244  
 coordinamento degli eventi della  
 replica 188  
 copia delle configurazioni della  
 replica 195  
 copia di aggiornamento completo  
 opzione di registrazione 44  
 COPYONCE parameter 392  
 creating monitors 227  
 creazione di serie secondarie  
 colonne nella destinazione 90  
 colonne registrate 43  
 dati di origine  
 uso di viste 102  
 righe di modifiche nella  
 destinazione 90  
 righe registrate delle modifiche 44  
 tecniche avanzate  
 durante la registrazione 102  
 utilizzo dei predicati 103  
 creazione di una serie secondaria di righe  
 (orizzontale)  
 sull'origine 44  
 creazione di una serie secondaria  
 orizzontale (riga)  
 nella destinazione 90  
 CRTDPRTBL command 362  
 CRTJRN command 33  
 CRTJRNRCV command 33  
 CTLSVR parameter 390

current receiver size 35

## D

dati  
 creazione di serie secondarie  
 durante la registrazione 102  
 uso di viste 102  
 utilizzo dei predicati 103  
 utilizzo delle viste per specificare i  
 predicati 103  
 impedimento di eliminazioni  
 doppie 58  
 manipolazione 105  
 richiamo da tabelle di origine 207  
 tecniche avanzate per la creazione di  
 serie secondarie 101  
 trasformazione  
 creazione di colonne calcolate 107  
 durante la registrazione 105  
 durante la sottoscrizione 105  
 ridenominazione delle  
 colonne 91, 107  
 visualizzazione cronologia 252  
 dati cronologici  
 dati di origine 45  
 tabelle CCD 80, 499  
 Dati DATE, replica 97  
 dati di staging 84  
 Dati NUMBER, replica 97  
 Dati TIMESTAMP, replica 97  
 DB2 Extenders  
 limitazioni 96  
 DB2 per z/OS  
 pianificazione 12  
 DB2 Query Patroller and replication 32  
 DBADM 13  
 DBCLOB (double-byte character large  
 object)  
 considerazioni sulle repliche 96  
 DELAY parameter 392  
 delete journal receiver exit program 199  
 delete journal receiver exit routine  
 about 37  
 delimitatore ; 259  
 delimitatore # 259  
 delimitatori, in script SQL generati 259  
 difference table 209  
 dimensione ricevitore, corrente 4  
 dimensione ricevitore corrente 4  
 disattivazione  
 oggetti registrati 168  
 serie di richieste 68, 185  
 dizionari di compressione (z/OS) 200  
 documentazione  
 accessibile 511  
 DPR registrations (System i)  
 adding 321  
 removing 382  
 DSPJRN command 257  
 durante dell'intervallo 72  
 durata relativa 72

## E

e-mail notification, replication 220  
 elaborazione della modalità di  
 transazione 4  
 elaborazione in modalità tabella 4, 71  
 elaborazione in modalità transazione 71  
 elaborazione runtime 71, 106  
 eliminazione  
 tabella dei segnali (SIGNAL) 208  
 Tabella  
 IBMSNAP\_APPLYTRACE 205  
 tabella IBMSNAP\_APPLYTRAIL 205  
 Tabella IBMSNAP\_CAPMON 207  
 Tabella IBMSNAP\_CAPTRACE 207  
 Tabella IBMSNAP\_UOW 450  
 tabella UOW (unit-of-work) 204  
 Tabelle CD (Change-Data) 204  
 tabelle di controllo 205  
 eliminazione automatica 203  
 eliminazioni doppie 58  
 email\_server parameter, Replication Alert  
 Monitor 233  
 ENDDPRAPY command 363  
 ENDDPRCAP command 366  
 errors  
 monitor\_errors parameter 233  
 monitoring with alert conditions 215  
 replication  
 alert conditions,  
 APPLY\_ERRORS 217  
 alert conditions,  
 CAPTURE\_ERRORS 217  
 alert conditions,  
 QAPPLY\_ERRORS 217  
 alert conditions,  
 QCAPTURE\_ERRORS 217  
 SQL 217  
 esecuzione, script SQL 259  
 esecuzione di prospetti a richiesta 251  
 event publishing  
 authorization requirements  
 Replication Alert Monitor 224  
 storing user IDs and passwords 17  
 event publishing commands  
 asnpwd 293  
 ansrct 297  
 asnsdrop 300  
 asnslist 301  
 asntdiff 302, 307  
 asntrc 310  
 asntrep 317  
 eventi, coordinamento 188  
 exit routines  
 ASNDONE  
 using 147  
 ASNLOAD  
 for System i 154  
 delete journal receiver (System i) 37

## F

fattore di blocco 68  
 file  
 \*.APP.log 136  
 \*.CAP.log 114  
 \*.err 136

- file (*Continua*)
  - \*.sqs 136
  - asndone.smp 146
  - asnload.ini 153
- file \*.APP.log 136
- file \*.CAP.log 114
- file \*.err 136
- file \*.sqs 136
- file asndone.smp 146
- file asnload.ini 153
- file di log Capture 114
- file diagnostici
  - memorizzazione 6, 7, 8
- file SQL, modifica 259
- frammentazione
  - orizzontale
    - nella destinazione 90
    - sull'origine 44
  - replica di aggiornamento ovunque 8
  - replica peer-to-peer 8
  - verticale
    - nella destinazione 90
    - sull'origine 43
- FRCFRQ parameter 400
- full-refresh copying
  - Apply for System i 57, 391
- FULLREPPGM parameter 391
- funzione load from cursor 152

## G

- gestione
  - requisiti di autorizzazione 13
- giornali
  - registrazione come origini 39
- global record 437
- GRTDPRAUT command
  - granting privileges to SQL packages 31
  - syntax 368
- GRTOBJAUT command 31

## I

- IASP groups 24
- IBMSNAP\_ALERTS control table 481
- IBMSNAP\_APPLY\_JOB table 454
- IBMSNAP\_AUTHTKN table 416
- IBMSNAP\_CONDITIONS control table 483
- IBMSNAP\_CONTACTGRP control table 488
- IBMSNAP\_CONTACTS control table 489
- IBMSNAP\_GROUPS control table 490
- IBMSNAP\_MONENQ control table 490
- IBMSNAP\_MONPARMS control table 490
- IBMSNAP\_MONSERVERS control table 493
- IBMSNAP\_MONTRACE control table 494
- IBMSNAP\_MONTRAIL control table 494
- IBMSNAP\_REG\_EXT table 434
- IBMSNAP\_SUSPENDS control table 496

- IBMSNAP\_TEMPLATES control table 497
- ID di correlazione 58
- ID utente
  - autorizzazione 14
  - per i trigger Capture 17
  - per il programma Apply 15
  - per il programma Capture 14
- impostazione
  - programmi Apply
    - per Linux 27
    - per UNIX 27
    - per Windows 27
  - programmi Capture
    - per Linux 27
    - per UNIX 27
    - per Windows 27
- impostazione secondaria della colonna (verticale)
  - nella destinazione 90
  - sull'origine 43
- impostazione secondaria della riga (orizzontale)
  - nella destinazione 90
- impostazione secondaria orizzontale (riga)
  - sull'origine 44
- impostazione secondaria verticale (colonna)
  - nella destinazione 90
  - sull'origine 43
- impostazione variabili d'ambiente
  - Programma Capture 27
- IMS DataPropagator 39
- INACTMSG parameter 391
- Independent Auxiliary Storage Pool (IASP) groups 24
- indici
  - tabelle di destinazione 92
- indici di destinazione 92
- informazioni legali 517
- integrazione
  - serie di richieste 181
  - trigger 9
- integrità referenziale 87
- invocation parameters
  - Analyzer
    - for System i 355
  - Apply program
    - for System i 133, 389
  - Capture program
    - for System i 112, 358, 396
  - Replication Alert Monitor
    - for UNIX 285
    - for Windows 285
    - for z/OS 285
  - replication commands
    - for System i 322, 331, 345, 363, 364, 366, 368, 376, 378, 382, 383, 385, 387, 389, 396, 403
- INZDPRCAP command 376
- istruzioni SQL
  - definizione per la serie di richieste 71
  - elaborazione runtime 106

## J

- JCL
  - avvio del programma Apply 157
  - avvio del programma Capture 157
  - avvio di Replication Alert Monitor 157
- JOB parameter 389, 396
- journal jobs
  - checking status 257
- journal message queues 36
- journal receivers
  - access 199
  - creating for source tables 33
  - delete journal receiver exit routine 37
  - managing 35
  - system management 35
  - threshold 35
  - user management 36
- journals
  - creating 33
  - creating for source tables 33
  - default message queue 36
  - entry types 509
  - managing 35
  - QSQRN journal 33
  - setup 33
  - starting 33
  - using 33
  - using remote journal function 56
  - using with different system times 35
- JRN parameter 398

## L

- LAG parameter 400
- latenza
  - Programma Apply 256
  - Programma Capture 254
- lavori di replica grandi 68
- lettore dello schermo 511
- lettura di dipendenze 54
- LIBPATH 27
- limitazioni
  - clausola WHERE 90
  - codifica dei dati 95
  - colonne LONG nelle tabelle di Oracle 95
  - DB2 Extenders LOB (large objects) 96
  - Microsoft SQL Server 45
  - nomi colonna, limiti 45
  - origini di dati relazionali non DB2 50, 54
  - Origini Oracle 95
  - procedure memorizzate 106
  - replica eterogenea 84, 87
  - repliche eterogenee 45
  - Sybase 45
  - Tabelle ASCII 505
  - tabelle CCD 87
  - tabelle di destinazione esistenti 89
  - Tabelle Unicode 505
  - tipi di dati astratti 95
  - tipi di dati definiti dall'utente 95
  - tipi di dati esterni 95
  - tipi di dati LOB 87

limitazioni (*Continua*)  
 tipi di dati LONG VARCHAR 95  
 tipi di dati LONG VARGRAPHIC 95  
 tipi di dati spaziali 95  
 viste 59  
 LOB (large object)  
 considerazioni sulle repliche 96  
 restrizioni dell'aggiornamento  
 ovunque 87

## M

manipolazione dei dati  
 creazione di colonne calcolate 107  
 durante la registrazione 105  
 durante la sottoscrizione 105  
 ridenominazione delle colonne 91,  
 107  
 marchi 519  
 max\_notification\_minutes parameter,  
 Replication Alert Monitor 233  
 max\_notifications\_per\_alert parameter,  
 Replication Alert Monitor 233  
 MAX\_SYNCH\_MINUTES, blocco dei  
 dati 68  
 membri impostazione richieste  
 abilitazione 175  
 aggiunta 74, 173  
 applicazione della serie secondaria di  
 righe 90  
 applicazione di una serie secondaria  
 di colonne 90  
 associazione dei tipi di dati 91  
 associazione tra le colonne 91  
 definizione chiave di destinazione 92  
 disabilitazione 174  
 numero per serie di richieste 64  
 replica con livelli multipli 84  
 replica di aggiornamento  
 ovunque 87  
 selezione dei tipi di destinazione 77  
 MEMLMT parameter 399  
 memoria  
 lettura record di registrazione 1  
 pianificazione 1  
 processi batch 1  
 Programma Apply 3  
 Programma Capture 1  
 registrazioni 1  
 serie di richieste 3  
 transazioni 1  
 utilizzo della tabella  
 IBMSNAP\_CAPMON per  
 ottimizzare 1  
 memorizzazione  
 file diagnostici 8  
 file diagnostici Apply 7  
 file diagnostici Capture 6  
 file trasferimento Apply 7  
 file trasferimento Capture 6  
 registrazione database e dati  
 giornale 4  
 requisiti 3  
 Tabella CD 5  
 tabella UOW 5  
 tabelle di controllo 5  
 tabelle di destinazione 5

memory  
 alert conditions  
 APPLY\_MEMORY 217  
 CAPTURE\_MEMORY 217  
 QAPPLY\_MEMORY 217  
 QCAPTURE\_MEMORY 217  
 Replication Alert Monitor 223  
 message queues, for journals 36  
 messages 233  
 messaggi 254, 255  
 Microsoft SQL Server  
 restrizioni replica 45  
 migrazione  
 pianificazione 1  
 mini-cicli 68  
 modalità batch JCL 157  
 modalità di condivisione dei dati 163  
 modifica, script SQL 259  
 monitor control server  
 IBMSNAP\_SUSPENDS control  
 table 496  
 IBMSNAP\_TEMPLATES control  
 table 497  
 Monitor control server  
 IBMSNAP\_ALERTS control table 481  
 IBMSNAP\_CONDITIONS control  
 table 483  
 IBMSNAP\_CONTACTGRP control  
 table 488  
 IBMSNAP\_CONTACTS control  
 table 489, 490  
 IBMSNAP\_MONENQ control  
 table 490  
 IBMSNAP\_MONPARMS control  
 table 490  
 IBMSNAP\_MONSERVICES control  
 table 493  
 IBMSNAP\_MONTRACE control  
 table 494  
 IBMSNAP\_MONTRAIL control  
 table 494  
 list of control tables 481  
 monitor\_errors parameter, Replication  
 Alert Monitor 233  
 monitor\_limit parameter  
 Replication Alert Monitor 233  
 monitor\_path parameter, Replication  
 Alert Monitor 233  
 Monitor program  
 messages 233  
 printing 233  
 monitor qualifier  
 replication 215  
 monitoring  
 for System i 257  
 replication 215  
 status of programs 257  
 monitoring publishing 227  
 monitoring replication 227  
 MONITV parameter 399  
 MONLMT parameter 398

## N

nickname  
 limitazioni  
 aggiornamento ovunque 50, 87

nickname (*Continua*)  
 limitazioni (*Continua*)  
 con tabelle CCD 45  
 replica con livelli multipli 84  
 tabelle di aggregati 79  
 per la funzione load from cursor 152  
 registrazione 41  
 NLS (national language support) 11  
 nomi  
 di trigger Capture 9  
 per servizi Windows 261  
 regole qualificatore Apply 261  
 regole qualificatore Monitor 261  
 regole schema Capture 261  
 serie di richieste 176  
 nomi servizi Windows 261  
 numeri di record relativo  
 utilizzato come chiave di  
 destinazione 92

## O

oggetti  
 arresto cattura modifiche 168  
 disattivazione 168  
 modifica degli attributi 166  
 registrazione 165  
 riattivazione 169  
 operating  
 Replication Alert Monitor 290  
 OPTSNGSET parameter 393  
 opzione arrestare Capture a seguito di  
 errore 48  
 origini  
 associazione con le destinazioni 74  
 gestione tabelle CCD 60  
 opzioni di registrazione  
 aggiornamenti come eliminazioni e  
 inserimenti 49  
 arrestare Capture a seguito di  
 errore 48  
 colonne post-immagine 45  
 colonne pre-immagine 45  
 copia di aggiornamento  
 completo 44  
 impostazione secondaria della  
 colonna (verticale) 43  
 impostazione secondaria della riga  
 (orizzontale) 44  
 prefisso pre-immagine 48  
 replica di modifica e  
 acquisizione 44  
 ricattura delle modifiche  
 (aggiornamenti) 50  
 rilevazione di conflitti 54  
 registrazione  
 origini dati IMS 39  
 relazionale non-DB2 41  
 tabelle DB2 39  
 viste 57, 59  
 registrazione colonne 43  
 richieste a 65  
 righe di registrazione 44  
 tabelle CCD (consistent-change-  
 data) 84  
 origini dati IMS  
 gestione tabelle CCD 60

- origini dati IMS (*Continua*)
  - registrazione 39
  - uso di tabelle CCD 39
- origini dati non relazionali
  - gestione tabelle CCD 60
  - uso di tabelle CCD 39
- origini dati relazionali non DB2
  - limitazioni
    - replica con livelli multipli 84
    - registrazione 41
    - uso di tabelle CCD 41
- Origini dati relazionali non-Db2
  - registrazione 41
  - server di origine 9
- origini della replica
  - gestione tabelle CCD 60
  - registrazione
    - colonne 43
    - origini dati IMS 39
    - Origini dati relazionali non-Db2 41
    - tabelle DB2 39
    - viste 59
  - richieste a 65
  - tabelle CCD (consistent-change-data) 84
  - unioni 58
- origini di dati relazionali non DB2
  - blocchi 10
  - limitazioni
    - aggiornamento ovunque 50, 54, 87
    - tabelle di aggregati 79
- origini replica
  - associazione con le destinazioni 74
  - registrazione
    - righe 44
- ottimizzazione
  - parametro commit\_interval 1
  - parametro memory\_limit 1
  - prestazioni 12
- overriding attributes (System i)
  - Capture program 377
- OVRDPRCAPA command 377

## P

- pacchetti, rebind 202
- parameters
  - Replication Alert Monitor
    - alert\_prune\_limit 233
    - autoprune 233
    - default values 233
    - description 233
    - email\_server 233
    - max\_notification\_minutes 233
    - max\_notifications\_per\_alert 233
    - monitor\_errors 233
    - monitor\_limit 233
    - monitor\_path 233
    - runonce 233
    - trace\_limit 233
- parameters, invocation
  - Analyzer
    - for System i 355
  - Apply program
    - for System i 133, 389

- parameters, invocation (*Continua*)
  - Capture program
    - for System i 358, 396
  - Replication Alert Monitor
    - for UNIX 285
    - for Windows 285
    - for z/OS 285
  - replication commands
    - for System i 322, 331, 345, 363, 364, 366, 368, 376, 378, 382, 383, 385, 387, 389, 396, 403
- parametri, richiamo
  - Programma Apply
    - per UNIX 136
    - per Windows 136
    - per z/OS 136
  - programma Capture
    - per Windows 114
  - Programma Capture
    - per UNIX 114
    - per z/OS 114
- parametri configurazione per DB2
  - APPLHEAPSZ 27
  - DBHEAP 27
  - LOGBUFSZ 27
  - LOGFILSIZ 27
  - LOGPRIMARY 27
  - LOGSECOND 27
  - MAXAPPLS 27
- parametri di richiamo
  - programma Apply
    - per UNIX 136
  - Programma Apply
    - per Windows 136
    - per z/OS 136
  - programma Capture
    - per Windows 114
    - per z/OS 114
  - Programma Capture
    - per System i 109
    - per UNIX 114
- parametro add\_partition
  - panoramica 115
- parametro apply\_path 136
- parametro apply\_qual 136
- parametro arm 162
- parametro asynchlogrd
  - panoramica 114
- parametro autoprune
  - panoramica 114
- parametro autostop 114
- parametro caf 114, 136
- parametro capture\_path 114
- parametro capture\_schema 114
- parametro capture\_server 114
- parametro commit\_interval
  - ottimizzazione 1
- parametro configurazione
  - APPLHEAPSZ 27
- parametro configurazione DBHEAP 27
- parametro configurazione
  - LOGBUFSZ 27
- parametro configurazione
  - LOGFILSIZ 27
- parametro configurazione
  - LOGPRIMARY 27

- parametro configurazione
  - LOGSECOND 27
- parametro configurazione
  - MAXAPPLS 27
- parametro control\_server 136
- parametro copyonce 136
- parametro db2\_subsystem 136
- parametro delay 136
- parametro errwait 136
- parametro inamsg 136
- parametro intervallo commit
  - panoramica 114
- parametro lag\_limit 114
- parametro limite memoria
  - panoramica 114
- parametro loadxit 136
- parametro logreuse (per Apply) 136
- parametro logreuse (per Capture) 114
- parametro logstdout (per Apply) 136
- parametro logstdout (per Capture) 114
- parametro memory\_limit
  - ottimizzazione 1
- parametro monitor\_interval (per Capture) 114
- parametro monitor\_limit 114
- parametro notify 136
- parametro opt4one 136
- Parametro prune\_interval 114
- parametro pwdfile 136
- parametro retention\_limit 114
- parametro sleep 136
- Parametro sleep\_interval 114
- parametro spillfile 136
- parametro sqlerrcontinue 136
- parametro startmode 114
- parametro term (per Apply) 136
- parametro term (per Capture) 114
- parametro trace\_limit
  - panoramica 114
- parametro trlreuse 136
- partizione di più database
  - record di registrazione 198
- password file 17
  - creating 293, 310
- passwords, storing 17
- personalizzazione, script SQL 259
- piani, rebind 202
- pianificazione
  - blocchi sulle tabelle CCD 10
  - coesistenza di trigger 9
  - influenza registrazione 9
  - memoria 1
  - migrazione 1
  - programmi di replica 243
  - requisiti di memorizzazione 3
  - rilevazione di conflitti 8, 54
  - serie di richieste 72
  - velocità di trasmissione della transazione 9
- pianificazione basata sugli eventi 72
- pianificazione in base all'ora 72
- più partizioni del database
  - Capture 31
- predicati
  - creazione di serie secondarie 103
  - definizione per le tabelle di destinazione 90



- prefisso, pre-immagine 48
- prefisso pre-immagine 48
- prestazioni
  - ottimizzazione 12
- primary keys
- relative record numbers for System i 57
- printing
  - Monitor program
  - messages 233
- procedure CALL
  - definizione per la serie di richieste 71
  - elaborazione runtime prima e dopo 106
- procedure memorizzate
  - definizione per la serie di richieste 71
  - manipolazione dei dati 106
- processi batch
  - esecuzione 157
  - memoria utilizzata da 1
- programma Apply
  - comando MODIFY 159
  - comunicazioni con
    - Centro di replica 245
    - dati di prestazioni 252
  - per UNIX
    - parametro apply\_path 136
    - parametro notify 136
    - parametro spillfile 136
  - per z/OS
    - parametro db2\_subsystem 136
- Programma Apply
  - analisi latenza 256
  - analisi velocità di trasmissione 256
  - blocco dei dati 68
  - comandi 263
    - asncmd 281
    - asnapply 275
  - comunicazioni con
    - Controllo degli avvisi di replica 249
    - Programma Capture 245, 246
    - trigger Capture 245, 248
  - connettività 21
  - elaborazione in modalità tabella 71
  - elaborazione in modalità transazione 71
  - funzionamento 131
  - ID utente 15
  - impostazione dei valori predefiniti per i parametri 144
  - istruzioni di elaborazione
    - runtime 106
  - messaggi 255
    - stampa 255
  - mini-cicli 68
  - modifica dei valori dei parametri 144
  - per Linux
    - bind 29
    - configurazione 29
    - impostazione 27
  - per System i
    - arresto 145
  - per UNIX
    - arresto 145

- Programma Apply (*Continua*)
  - per UNIX (*Continua*)
    - avvio 131, 507
    - bind 29
    - configurazione 29
    - impostazione 27
    - parametri predefiniti 134
    - parametro apply\_qual 136
    - parametro caf 136
    - parametro control\_server 136
    - parametro copyonce 136
    - parametro delay 136
    - parametro errwait 136
    - parametro inamsg 136
    - parametro loadxit 136
    - parametro logreuse 136
    - parametro logstdout 136
    - parametro opt4one 136
    - parametro pwdfile 136
    - parametro sleep 136
    - parametro sqlerrcontinue 136
    - parametro term 136
    - parametro trlreuse 136
    - verifica dello stato 251
  - per Windows
    - arresto 145
    - avvio 131, 507
    - bind 29
    - configurazione 29
    - impostazione 27
    - parametri predefiniti 134
    - parametro apply\_path 136
    - parametro apply\_qual 136
    - parametro caf 136
    - parametro control\_server 136
    - parametro copyonce 136
    - parametro delay 136
    - parametro errwait 136
    - parametro inamsg 136
    - parametro loadxit 136
    - parametro logreuse 136
    - parametro logstdout 136
    - parametro notify 136
    - parametro opt4one 136
    - parametro pwdfile 136
    - parametro sleep 136
    - parametro spillfile 136
    - parametro sqlerrcontinue 136
    - parametro term 136
    - parametro trlreuse 136
    - verifica dello stato 251
  - per z/OS
    - arresto 145
    - avvio 131, 159
    - impostazione 31
    - parametri 134
    - parametro apply\_path 136
    - parametro apply\_qual 136
    - parametro caf 136
    - parametro control\_server 136
    - parametro copyonce 136
    - parametro delay 136
    - parametro errwait 136
    - parametro inamsg 136
    - parametro loadxit 136
    - parametro logreuse 136
    - parametro logstdout 136
    - parametro notify 136
    - parametro opt4one 136
    - parametro pwdfile 136
    - parametro sleep 136
    - parametro spillfile 136
    - parametro sqlerrcontinue 136
    - parametro term 136
    - parametro trlreuse 136
    - verifica dello stato 251

- Programma Apply (*Continua*)
  - per z/OS (*Continua*)
    - parametro notify 136
    - parametro opt4one 136
    - parametro pwdfile 136
    - parametro sleep 136
    - parametro spillfile 136
    - parametro term 136
    - parametro trlreuse 136
    - valore predefinito 134
    - verifica dello stato 251
  - pianificazione 243
  - requisiti di autorizzazione 15
- programma Capture
  - dati di prestazioni 252
  - dove avviarlo 114
  - per System i
    - requisiti di autorizzazione 13
  - per UNIX
    - parametro capture\_server 114
    - parametro intervallo commit 114
    - parametro retention\_limit 114
    - Parametro sleep\_interval 114
    - parametro term 114
    - parametro trace\_limit 114
  - per Windows
    - parametro add\_partition 115
  - per z/OS
    - parametro add\_partition 115
  - pianificazione 243
- Programma Capture
  - analisi latenza 254
  - analisi velocità di trasmissione 254
  - comandi 263
    - asncap 263
    - asncmd 271
  - comando MODIFY 159
  - comunicazioni con
    - Centro di replica 245
    - programma Apply 246
    - Programma Apply 245
    - Replication Alert Monitor 249
  - connettività 21
  - esecuzione di più di uno 25
  - ID utente 14
  - impostazione dei valori predefiniti per i parametri 123
  - impostazione variabili d'ambiente 27
  - memoria utilizzata da 1
  - messaggi 254
    - stampa 254
  - modifica degli schemi 171
  - modifica dei valori dei parametri 123
  - modifica del funzionamento durante l'esecuzione 125
  - per Linux
    - bind 28
    - impostazione 27
  - per System i
    - arresto 127
    - funzionamento 109
    - parametri predefiniti 112
  - per UNIX
    - arresto 127
    - avvio 109, 507
    - bind 28
    - configurazione 27

- Programma Capture (*Continua*)
  - per UNIX (*Continua*)
    - funzionamento 109
    - impostazione 27
    - inizializzazione in corso 128
    - parametri avvio a caldo 114
    - parametri avvio a freddo 114
    - parametri predefiniti 112
    - parametro add\_partition 115
    - parametro asynchlogrd 114
    - parametro autoprune 114
    - parametro autostop 114
    - parametro caf 114
    - parametro capture\_path 114
    - parametro capture\_schema 114
    - parametro lag\_limit 114
    - parametro limite memoria 114
    - parametro logreuse 114
    - parametro logstdout 114
    - parametro monitor\_interval 114
    - parametro monitor\_limit 114
    - Parametro prune\_interval 114
    - parametro startmode 114
    - ripristino 130
    - sospensione 129
    - verifica dello stato 251
  - per Windows
    - arresto 127
    - avvio 109, 507
    - bind 28
    - configurazione 27
    - funzionamento 109
    - impostazione 27
    - inizializzazione in corso 128
    - parametri avvio a caldo 114
    - parametri avvio a freddo 114
    - parametri predefiniti 112
    - parametro asynchlogrd 114
    - parametro autoprune 114
    - parametro autostop 114
    - parametro caf 114
    - parametro capture\_path 114
    - parametro capture\_schema 114
    - parametro capture\_server 114
    - parametro commit\_interval 114
    - parametro lag\_limit 114
    - parametro logreuse 114
    - parametro logstdout 114
    - parametro memory\_limit 114
    - Parametro monitor\_interval 114
    - parametro monitor\_limit 114
    - parametro prune\_interval 114
    - parametro retention\_limit 114
    - parametro sleep\_interval 114
    - parametro startmode 114
    - parametro term 114
    - parametro trace\_limit 114
    - ripristino 130
    - sospensione 129
    - verifica dello stato 251
  - per z/OS
    - arresto 127
    - avvio 109
    - funzionamento 109
    - impostazione 31
    - inizializzazione in corso 128
    - parametri avvio a caldo 114

- Programma Capture (*Continua*)
  - per z/OS (*Continua*)
    - parametri avvio a freddo 114
    - parametri predefiniti 112
    - parametro asynchlogrd 114
    - parametro autoprune 114
    - parametro autostop 114
    - parametro caf 114
    - parametro capture\_path 114
    - parametro capture\_schema 114
    - parametro capture\_server 114
    - parametro intervallo commit 114
    - parametro lag\_limit 114
    - parametro limite memoria 114
    - parametro logreuse 114
    - parametro logstdout 114
    - parametro monitor\_interval 114
    - parametro monitor\_limit 114
    - Parametro prune\_interval 114
    - parametro retention\_limit 114
    - Parametro sleep\_interval 114
    - parametro startmode 114
    - parametro term 114
    - parametro trace\_limit 114
    - ripristino 130
    - sospensione 129
    - verifica dello stato 251
  - prevenzione avvio a freddo 206
  - requisiti di autorizzazione 14
  - segnali 188
  - programma di utilità RUNSTATS 201
  - promozione
    - configurazioni della replica 195
  - prospetto Analyzer
    - comando asnanalyze 282
  - punti di recupero, distribuiti 190
  - punti di recupero distribuiti 190

## Q

- Q Apply program
  - alert conditions 217
- Q Capture program
  - alert conditions 217
- Q replication
  - authorization requirements
    - for Replication Alert Monitor 224
  - control tables
    - list, Replication Alert Monitor 481
  - storing user IDs and passwords 17
- Q replication commands
  - asnsrct 297
  - asnsdrop 300
  - asnslist 301
  - asnspwd 293
  - asnstrc 310
  - asntdiff 302, 307
  - asntrep 317
- QTIME differences 35
- qualificatori Apply
  - controllo stato 256
  - modifica nelle serie di richieste 183
  - numero di serie di richieste
    - associate 64
  - regole di denominazione 261
  - utilizzo all'avvio del programma
    - Apply 131

- qualificatori Monitor, regole di
  - denominazione 261

## R

- RCVJRNE command 35
- rebind, pacchetti e piani 202
- record di registrazione
  - archiviato prima di essere acquisito 4
  - conservazione 198
  - dizionari di compressione
    - (z/OS) 200
  - manutenzione 197
  - partizioni di più database 198
- recupero da errore di I/O, tabelle di
  - controllo 206
- recupero roll-forward 27
- registering
  - options for sources
    - relative record numbers 57
    - using remote journals 56
- registrations
  - adding 321
  - removing 382
- registrazione
  - oggetti 165
  - opzioni per origini
    - aggiornamenti come eliminazioni e inserimenti 49
  - arrestare Capture a seguito di errore 48
  - colonne post-immagine 45
  - colonne pre-immagine 45
  - copia di aggiornamento
    - completo 44
  - creazione di una serie secondaria
    - di righe (orizzontale) 44
  - impostazione secondaria della colonna (verticale) 43
  - prefisso pre-immagine 48
  - replica di modifica e acquisizione 44
  - ricattura delle modifiche
    - (aggiornamenti) 50
  - rilevazione di conflitti 54
  - origini dati IMS 39
  - Origini dati relazionali non-Db2 41
  - pianificazione influenza su 9
  - tabelle 165
  - tabelle DB2 39
  - viste
    - panoramica 57, 59
    - procedura 165
- registrazioni
  - aggiunta di colonne 166
  - arresto cattura modifiche 168
  - attributi, modifica 166
  - disattivazione 168
  - riattivazione 169
  - rimozione 170
- registrazioni origine, gestione 197
- regole cattura righe 44
- reinitializing 231
- reinizializzazione del programma
  - Capture
    - per UNIX 128
    - per Windows 128

- reinizializzazione del programma
  - Capture *(Continua)*
  - per z/OS 128
- relative record numbers
  - as primary key for System i 57
  - support for System i 57
- remote journaling with different system times 35
- remote journals as sources 56
- remote source tables 56
- replica a fasi 84
- replica aggiornamento differenziale 44
- replica con livelli multipli
  - definizione delle serie di richieste 84
- replica DB2
  - requisiti di autorizzazione 13
- replica di aggiornamento ovunque
  - definizione delle serie di richieste 87
  - frammentazione per 8
  - ricattura delle modifiche 50
  - rilevazione di conflitti
    - panoramica 54
    - pianificazione per 8
    - requisiti 45, 54
- replica di modifica e acquisizione
  - descrizione 44
  - opzione di registrazione 44
- replica eterogenea
  - limitazioni
    - tabelle CCD 45
- replica peer-to-peer
  - rilevazione di conflitti 8
- Replica SQL 251
  - panoramica replica 1
- Replication Alert Monitor 226, 230, 231
  - alert conditions
    - e-mail notifications 220
    - events 215
    - list 217
    - overview 217
    - sending alerts to z/OS
      - console 222
      - status 215
      - thresholds 215
    - alerts 215
    - authorization requirements 224
    - changing alert conditions 228
    - changing runtime parameters 236
    - Classic replication 215
    - comando MODIFY 159
    - comunicazioni con
      - Centro di replica 249
    - contact groups 215
    - contacts 215
    - control tables
      - IBMSNAP\_ALERTS 481
      - IBMSNAP\_CONDITIONS 483
      - IBMSNAP\_CONTACTGRP 488
      - IBMSNAP\_CONTACTS 489
      - IBMSNAP\_GROUPS 490
      - IBMSNAP\_MONENQ 490
      - IBMSNAP\_MONPARMS 490
      - IBMSNAP\_MONSERVERS 493
      - IBMSNAP\_MONTRACE 494
      - IBMSNAP\_MONTRAIL 494
    - creating control tables 225
    - description 215
  - Replication Alert Monitor *(Continua)*
    - for Linux
      - binding 224
    - for UNIX
      - binding 224
      - operating 290
    - for Windows
      - binding 224
      - operating 290
    - for z/OS
      - operating 290
    - memory 223
    - monitoring replication, overview 215
    - operational errors 237
    - parameters
      - alert\_prune\_limit 233
      - autoprun 233
      - default values 233
      - descriptions 233
      - email\_server 233
      - max\_notification\_minutes 233
      - max\_notifications\_per\_alert 233
      - monitor\_errors 233
      - monitor\_interval 233
      - monitor\_limit 233
      - monitor\_path 233
      - runonce 233
      - trace\_limit 233
    - per Windows
      - verifica dello stato 251
    - pruning control tables 238
    - selecting alert conditions 227
    - setting up 223
    - specifying notification criteria 237
    - specifying run frequency 237
    - stopping 232
  - Replication Analyzer
    - for System i
      - creating SQL packages 30
      - invocation parameters 355
  - replication commands
    - ADDJOBSCDE 244
    - asnslist 301
    - asntdiff 302, 307
    - asntrep 317
    - CRTJRNRCV 33
    - DSPJRN 257
    - for System i
      - ADDDPRREG 321
      - ADDDPRSUB 329
      - ADDDPRSUBM 344
      - ANZDPR 354
      - ANZDPRJRN 37
      - CHGDPRCAPA 357
      - CHGJRN 36
      - CRTDPRTBL 362
      - CRTJRN 33
      - ENDDPRAPY 363
      - ENDDPRCAP 366
      - GRTDPRAUT 31, 368
      - GRTOBJAUT 31
      - INZDPRCAP 376
      - OVRDPRCAPA 377
      - RCVJRNE 35
      - RMVDPRREG 382
      - RMVDPRSUB 383
      - RMVDPRSUBM 385
  - replication commands *(Continua)*
    - for System i *(Continua)*
      - RVKDPRAUT 386
      - SBMJOB 244
      - STRDPRAPY 388
      - STRDPRCAP 395
      - STRJRNPf 33
      - WRKDPTRC 402
      - WRKJOB 257
      - WRKSBJOB 257
      - WRKSBSJOB 257
    - for UNIX
      - asnmcmd 290
    - for Windows
      - asnmcmd 290
    - for z/OS
      - asnmcmd 290
  - replication services
    - creating 240
    - description 239
    - display name 239
    - dropping 242
    - listing 301
    - name 239
    - starting 241
    - stopping 241
    - viewing 241
  - repliche eterogenee
    - limitazioni
      - aggiornamento ovunque 50, 87
      - replica con livelli multipli 84
      - tabelle di aggregati 79
      - registrazione origini 41
    - requisiti di registrazione
      - server d'origine relazionale non-DB2 9
      - server di destinazione 4
      - server di origine DB2 4
    - RESTART parameter 396
    - restrizioni per la codifica dei dati 95
    - resuming 231
    - RETAIN parameter 399
    - riattivazione
      - oggetti 169
      - registrazioni 169
      - tabelle 169
    - ricattura delle modifiche (aggiornamenti) 50
    - ricevitori giornale
      - corrente, dimensione 4
      - manutenzione 197
    - richiesta delle origini 65
    - ridenominazione delle colonne 91, 107
    - righe
      - creazione di serie secondarie
        - nella destinazione 90
        - sull'origine 44
      - definizione nella tabella di destinazione 90
      - disponibile per la replica 44
      - registrazione nella tabella di origine 44
    - rilevamento di spazi 80, 499
    - rilevazione di conflitti
      - livelli di 54
      - panoramica 54
      - pianificazione 8

- rilevazione di conflitti (*Continua*)
  - replica di aggiornamento ovunque 8
  - replica peer-to-peer 8
  - requisiti 45
- riorganizzazione
  - tabelle di controllo 202
- ripristino
  - Programma Capture
    - per UNIX 130
    - per Windows 130
    - per z/OS 130
- RMVDPREG command 382
- RMVDPRSUB command 383
- RMVDPRSUBM command 385
- routine di chiusura
  - ASNDONE
    - utilizzo 146
  - ASNLOAD
    - per UNIX 149
    - per Windows 149
    - per z/OS 151
    - personalizzazione 152
    - utilizzo 148
- routine di chiusura ASNDONE
  - transazioni rifiutate 54
  - utilizzo 146
- routine di chiusura ASNLOAD
  - descrizione 148
  - gestione errori 148
  - per UNIX 149
  - per Windows 149
  - per z/OS 151
  - personalizzazione
    - funzionamento 152
  - uso del file asnload.ini 153
  - uso della funzione load from cursor 152
- ROWID 96
- RRN 57
- RTYWAIT parameter 392
- runonce parameter, Replication Alert Monitor 233
- RVKDPRAUT command 386

## S

- SBMJOB command 244
- schemi
  - modifica 171
  - regole di denominazione 261
- schemi Capture
  - modifica 171
  - regole di denominazione 261
  - utilizzo di più 25
- SCM (Service Control Manager)
  - creating replication services 240
  - description 239
  - dropping replication services 242
  - starting replication services 241
  - stopping replication services 241
  - viewing replication services 241
- script SQL 259
- script SQL generati 259
- segnali
  - CAPSTART 192
  - CAPSTOP 193

- segnali (*Continua*)
  - impostazione punti di recupero
    - distribuiti 190
    - STOP 189, 190
    - USER 188
  - segnali CAPSTART 192
  - segnali CAPSTOP 193
  - segnali di Capture 188
  - segnali STOP 189, 190
  - segnali USER 188
- serie di richieste
  - abilitazione dei membri 175
  - aggiunta di membri 74, 173
  - coerenza dei dati 87
  - colonne 90
  - creazione 65
  - creazione di nuove 173
  - disabilitazione dei membri 174
  - disattivazione 185
  - integrazione 181
  - integrità referenziale 87
  - istruzioni di elaborazione
    - runtime 106
  - istruzioni SQL 71
  - livello di attivazione 68
  - mini-cicli 68
  - modalità di elaborazione 71
  - modifica
    - attributi 175
    - nomi 176
    - qualificatori Apply 183
  - numero di qualificatori Apply 64
  - pianificazione
    - basata sugli eventi 72
    - in base all'ora 72
  - procedure memorizzate 71
  - replica con livelli multipli 84
  - replica di aggiornamento
    - ovunque 87
  - righe 90
  - rimozione 187
  - suddivisione 178
- serie di richieste non attive 68
- server di controllo Capture
  - più schemi Capture 25
- server di destinazione
  - influenza registrazione 4
- server di origine
  - DB2
    - influenza registrazione 4
- server origine
  - relazionale non-DB2
    - influenza registrazione 9
- server Q Capture
  - tabella di controllo
    - IBMQREP\_COLVERSION 424
  - tabella di controllo
    - IBMQREP\_IGNTRAN 428
  - tabella di controllo
    - IBMQREP\_TABVERSION 449
- Server Q Capture
  - tabella di controllo
    - IBMQREP\_IGNTRANTRC 429
- server relazionali non-DB2
  - connessione 22
- Service Control Manager (SCM)
  - creating replication services 240

- Service Control Manager (SCM) (*Continua*)
  - description 239
  - dropping replication services 242
  - starting a replication service 241
  - stopping a replication service 241
  - viewing replication services 241
- servizi software 511
- setting up
  - journals 33
    - Replication Alert Monitor 224
- sistemi origine, gestione 197
- sospensione
  - Programma Capture
    - per UNIX 129
    - per Windows 129
    - per z/OS 129
- source tables
  - creating journals for 33
- sources
  - registration options
    - relative record numbers 57
    - using remote journals 56
- spazio su disco
  - requisiti 3
- SQL packages
  - creating for Apply program 30
  - creating for Capture program 29
  - creating for Replication Analyzer 30
- SQL replication commands
  - asnpwd 293
  - asnsrct 297
  - asnsdrop 300
  - asnrct 310
- stampa
  - Programma Apply
    - messaggi 255
  - Programma Capture
    - stampa 254
- starting 230
  - Apply program
    - for System i 133, 388
  - Capture program
    - for System i 112, 395
- stato
  - programma Apply 251
  - programma Capture 251
  - Replication Alert Monitor 251
- status
  - Apply program 257
  - Capture program 257
  - journal jobs 257
- stopping
  - Apply program
    - for System i 363
  - Capture program
    - for System i 366
  - Replication Alert Monitor
    - for UNIX 290
    - for Windows 290
    - for z/OS 290
- STRDPRAPY command 388
- STRDPRCAP command 395
- STRJRNP command 33
- SUBNFYPM parameter 391
- subscription-set members
  - adding 344

- subscription-set members (*Continua*)
  - removing 385
- subscription sets
  - adding 329
  - removing 383
- suddivisione
  - serie di richieste 178
- suggerimenti
  - eliminazione di righe dalla tabella di traccia Apply 136
  - stima dello spazio di utilizzo 3
  - uso dei parametri sleep e copyonce 136
  - uso delle procedure memorizzate per ulteriori elaborazioni di serie 146
  - verifica inizio cattura modifiche 109
  - verifica se Apply ha elaborato una serie con esito positivo 136
- supporto IBM 511
- supporto nomi lunghi
  - pianificazione 12
- suspending 231
- Sybase
  - restrizioni replica 45
- synchronization
  - asntdiff and asntrepair utilities 209
- SYSADM 13
- system change journal management 35
- system commands
  - asnpwd 293
  - asnscri 297
  - asnsdrop 300
  - asnslist 301
  - asntdiff 302, 307
  - asntrc 310
  - asntrep 317
- System i data sources
  - with remote journaling 56
- System i server
  - connecting to 21

## T

- tabella IBMSNAP\_SUBS\_COLS 465
- tabella APPPARMS (parametri Apply)
  - modifica 145
- tabella CAPPARMS (parametri Capture)
  - modifica 127
  - utilizzo 123
- tabella copia utente
  - definizione 77
  - struttura 502
  - utilizzo 79
- tabella degli eventi di richieste (SUBS\_EVENT)
  - inserimento degli eventi 72
- tabella dei segnali (SIGNAL)
  - eliminazione 208
- tabella di controllo
  - IBMQREP\_COLVERSION 424
- tabella di controllo
  - IBMQREP\_IGNTRAN 428
- tabella di controllo
  - IBMQREP\_IGNTRANTRC 429
- tabella di controllo
  - IBMQREP\_TABVERSION 449
- tabella IBMSNAP\_APPENQ 453

- Tabella IBMSNAP\_APPLYTRACE
  - eliminazione 205
  - struttura 458
- tabella IBMSNAP\_APPLYTRAIL
  - eliminazione 205
  - struttura 459
- tabella IBMSNAP\_APPPARMS 455
  - utilizzo 144
- tabella IBMSNAP\_CAPENQ 417
- Tabella IBMSNAP\_CAPMON
  - eliminazione 207
  - struttura 418
- Tabella IBMSNAP\_CAPPARMS
  - struttura 420
- Tabella IBMSNAP\_CAPSCHEMAS 423
- Tabella IBMSNAP\_CAPTRACE
  - eliminazione 207
  - struttura 425
- tabella IBMSNAP\_PARTITIONINFO 430
- Tabella
  - IBMSNAP\_PARTITIONINFO 430
- tabella IBMSNAP\_PRUNCNTL 431
- Tabella IBMSNAP\_PRUNCNTL 431
- Tabella IBMSNAP\_PRUNE\_LOCK 433
- Tabella IBMSNAP\_PRUNE\_SET 434
- tabella IBMSNAP\_REG\_SYNCH 443
- tabella IBMSNAP\_REGISTER 436
- Tabella IBMSNAP\_REGISTER 436
- Tabella IBMSNAP\_RESTART 444
- tabella IBMSNAP\_SEQTABLE 446
- Tabella IBMSNAP\_SIGNAL
  - struttura 446
- Tabella IBMSNAP\_SUBS\_COLS 465
- tabella IBMSNAP\_SUBS\_EVENT
  - struttura 467
- Tabella IBMSNAP\_SUBS\_SET 473
- tabella IBMSNAP\_SUBS\_STMTS 479
- Tabella IBMSNAP\_SUBS\_STMTS 479
- Tabella IBMSNAP\_UOW
  - eliminazione 450
  - struttura 450
- tabella membri richiesta (SUBS\_MEMBR) 152
- tabella membri sottoscrizione (SUBS\_MEMBR) 468
- tabella parametri Apply (APPPARMS)
  - modifica 145
- tabella parametri Capture (CAPPARMS)
  - modifica 127
  - utilizzo 123
- tabella SIGNAL (dei segnali)
  - eliminazione 208
- tabella SUBS\_EVENT (eventi di richieste)
  - inserimento degli eventi 72
- tabella SUBS\_MEMBR (membri sottoscrizione) 152
- Tabella SUBS\_MEMBR (membri sottoscrizione) 468
- tabella UOW (unit-of-work)
  - colonne nelle tabelle CCD 80, 499
  - eliminazione 204
  - requisiti di memorizzazione 5
- tabelle
  - aggiunta di colonne 166
  - aggregati di modifica 498
  - aggregazione di base 498
  - arresto cattura modifiche 168

- tabelle (*Continua*)
  - CCD (consistent-change-data)
    - server di controllo Capture 426
  - CD (change-data) 427
  - copia utente 502
  - disattivazione 168
  - gestione tabelle CCD 60
  - IBMQREP\_COLVERSION 424
  - IBMQREP\_IGNTRAN 428
  - IBMQREP\_IGNTRANTRC 429
  - IBMQREP\_TABVERSION 449
  - IBMSNAP\_APPENQ 453
  - IBMSNAP\_APPLYTRACE 458
  - IBMSNAP\_APPLYTRAIL 459
  - IBMSNAP\_APPPARMS 455
  - IBMSNAP\_CAPENQ 417
  - IBMSNAP\_CAPMON 207, 418
  - IBMSNAP\_CAPPARMS 420
  - IBMSNAP\_CAPSCHEMAS 423
  - IBMSNAP\_CAPTRACE 207, 425
  - IBMSNAP\_PARTITIONINFO 430
  - IBMSNAP\_PRUNCNTL 431
  - IBMSNAP\_PRUNE\_LOCK 433
  - IBMSNAP\_PRUNE\_SET 434
  - IBMSNAP\_REG\_SYNCH 443
  - IBMSNAP\_REGISTER 436
  - IBMSNAP\_RESTART 444
  - IBMSNAP\_SEQTABLE 446
  - IBMSNAP\_SIGNAL 446
  - IBMSNAP\_SUBS\_COLS 465
  - IBMSNAP\_SUBS\_EVENT 467
  - IBMSNAP\_SUBS\_SET 473
  - IBMSNAP\_SUBS\_STMTS 479
  - IBMSNAP\_UOW 450
  - istantanea 501
  - modifica degli attributi 166
  - registrazione
    - DB2 39
    - procedura 165
    - relazionale non-DB2 41
  - replica 8, 502
  - riattivazione 169
  - rilevazione di conflitto per 8
  - rimozione registrazioni 170
  - SUBS\_MEMBR (membri richiesta) 152
  - SUBS\_MEMBR (membri sottoscrizione) 468
  - tabelle di controllo
    - dinamiche 201
    - eliminazione 205
    - manutenzione 201
    - programma di utilità RUNSTATS 201
    - recupero da errore di I/O 206
    - recupero da malfunzionamento
      - connettività 206
      - riorganizzazione 202
      - statiche 202
  - tabelle di destinazione 208
    - manutenzione 208
  - tabelle aggregati
    - aggregati di modifica 498
    - aggregazione di base 498
  - tabelle aggregati di base
    - utilizzo 79

- Tabelle aggregati di modifica
  - definizione 77
  - struttura 498
- Tabelle ASCII 505
- tabelle CCD (consistent-change-data)
  - aggiunta di colonne UOW 80, 499
  - blocchi su 10
  - esterna
    - replica con livelli multipli 84
  - interna
    - destinazioni multiple 83
  - origini dati non relazionali
    - gestione tabelle CCD 60
    - uso di tabelle CCD 39
  - origini della replica 84
  - origini di dati relazionali non DB2
    - uso di tabelle CCD 41
  - struttura
    - server di controllo Capture 426
  - utilizzo
    - cronologia o controllo 80, 499
    - replica con livelli multipli 84
- tabelle CCD esterne
  - replica con livelli multipli 84
- tabelle CCD interne
  - destinazioni multiple 83
- tabelle CD (change data)
  - eliminazione 204
  - requisiti di memorizzazione 5
  - struttura 427
- tabelle CD (Change-Data)
  - riepilogo contenuti 79
- Tabelle CD (Change-Data)
  - eliminazione 204
  - per unioni 58
  - per viste 57
  - requisiti di memorizzazione 5
  - struttura 427
- tabelle change-data (CD)
  - riepilogo contenuti 79
- tabelle codici
  - compatibile 10
  - traduzione 10
  - variabile di ambiente
    - DB2CODEPAGE 11
- tabelle DB2
  - registrazione 39
- tabelle definite dall'utente 77, 89
- tabelle della replica
  - ricattura delle modifiche 50
- tabelle di aggregati
  - aggregati di modifica 79
  - aggregazione di base 79
- tabelle di aggregazione di base
  - definizione 77
  - struttura 498
- tabelle di aggregazione di modifiche
  - utilizzo 79
- tabelle di catalogo, registrazione 39
- tabelle di controllo
  - CCD (consistent-change-data)
    - server di controllo Capture 426
  - CD (change-data) 427
  - concessione dell'autorizzazione per System i 13
  - consultazione rapida
    - server Capture 414
- tabelle di controllo (*Continua*)
  - consultazione rapida (*Continua*)
    - server di controllo Apply 452
    - server di destinazione 497
  - creazione 23
    - origini relazionali per non-DB2 25
    - più partizioni del database 26
    - più serie 25
    - più sistemi operativi database 23
  - dinamiche 201
  - eliminazione 205
  - IBMSNAP\_APPENQ 453
  - IBMSNAP\_APPLYTRACE 458
  - IBMSNAP\_APPLYTRAIL 459
  - IBMSNAP\_APPPARMS 455
  - IBMSNAP\_CAPENQ 417
  - IBMSNAP\_CAPMON
    - eliminazione 207
    - struttura 418
  - IBMSNAP\_CAPPARMS
    - struttura 420
  - IBMSNAP\_CAPSCHEMAS 423
  - IBMSNAP\_CAPTRACE
    - eliminazione 207
    - struttura 425
  - IBMSNAP\_PARTITIONINFO 430
  - IBMSNAP\_PRUNCNTL 431
  - IBMSNAP\_PRUNE\_LOCK 433
  - IBMSNAP\_PRUNE\_SET 434
  - IBMSNAP\_REG\_SYNCH 443
  - IBMSNAP\_REGISTER 436
  - IBMSNAP\_RESTART 444
  - IBMSNAP\_SEQTABLE 446
  - IBMSNAP\_SIGNAL 446
  - IBMSNAP\_SUBS\_COLS 465
  - IBMSNAP\_SUBS\_EVENT 467
  - IBMSNAP\_SUBS\_SET 473
  - IBMSNAP\_SUBS\_STMTS 479
  - IBMSNAP\_UOW 450
  - manutenzione 201
  - programma di utilità RUNSTATS 201
  - rebind, pacchetti e piani 202
  - recupero da errore di I/O 206
  - recupero da malfunzionamento
    - connettività 206
  - requisiti di memorizzazione 5
  - riorganizzazione 202
  - server Capture 414
  - server di controllo Apply 452
  - server di destinazione 497
  - server Q Capture
    - IBMQREP\_COLVERSION 424
    - IBMQREP\_IGNTRAN 428
    - IBMQREP\_IGNTRANTRC 429
    - IBMQREP\_TABVERSION 449
  - statiche 202
  - SUBS\_MEMBR (membri
    - sottoscrizione) 468
- Tabelle di controllo Apply
  - APPPARMS (parametri Apply)
    - modifica 145
  - IBMSNAP\_APPENQ 453
  - IBMSNAP\_APPLYTRACE 458
  - IBMSNAP\_APPLYTRAIL 459
  - IBMSNAP\_APPPARMS 455
    - utilizzo 144
  - IBMSNAP\_SUBS\_COLS 465
- Tabelle di controllo Apply (*Continua*)
  - IBMSNAP\_SUBS\_EVENT 467
  - IBMSNAP\_SUBS\_SET 473
  - IBMSNAP\_SUBS\_STMTS 479
  - SUBS\_MEMBR (membri
    - sottoscrizione) 468
- tabelle di controllo Capture
  - CAPPARMS (parametri Capture)
    - modifica 127
    - utilizzo 123
  - CCD (consistent-change-data) 426
  - CD (change-data) 427
  - IBMSNAP\_CAPENQ 417
  - IBMSNAP\_CAPMON 418
  - IBMSNAP\_CAPPARMS
    - struttura 420
  - IBMSNAP\_CAPSCHEMAS 423
  - IBMSNAP\_CAPTRACE 425
  - IBMSNAP\_PARTITIONINFO 430
  - IBMSNAP\_PRUNCNTL 431
  - IBMSNAP\_PRUNE\_LOCK 433
  - IBMSNAP\_PRUNE\_SET 434
  - IBMSNAP\_REG\_SYNCH 443
  - IBMSNAP\_REGISTER 436
  - IBMSNAP\_RESTART 444
  - IBMSNAP\_SEQTABLE 446
  - IBMSNAP\_SIGNAL 446
  - IBMSNAP\_UOW 450
- tabelle di controllo dinamiche 201
- tabelle di controllo statiche 202
- tabelle di destinazione
  - aggregati di modifica
    - definizione 77
    - struttura 498
  - aggregazione di base
    - definizione 77
    - struttura 498
    - utilizzo 79
  - aggregazione di modifiche
    - utilizzo 79
  - applicazione della serie secondaria di righe 90
  - applicazione di una serie secondaria di colonne 90
  - associazione alle origini 74
  - CCD (consistent-change-data)
    - panoramica 77
  - copia utente
    - definizione 77
    - struttura 502
    - utilizzo 79
  - definita dall'utente 77, 89
  - definizione chiave di destinazione 92
  - definizione delle colonne 90
  - definizione delle righe 90
  - frammentazione 90
  - istantanea
    - definizione 77
    - struttura 501
    - utilizzo 79
  - manutenzione 208
  - nuove colonne per 107
  - replica
    - definizione 77
    - rilevazione di conflitto per 8
    - struttura 502
    - utilizzo 87

- tabelle di destinazione (*Continua*)
  - requisiti di memorizzazione 5
  - strutture della tabella, consultazione rapida 497
- tabelle di destinazione multiple 83
- tabelle di origine
  - aggiunta di colonne 166
  - manutenzione 197
  - richiamo di dati persi 207
- Tablelle di replica
  - definizione 77
  - definizione delle destinazioni di lettura/scrittura 87
  - struttura 502
- tabelle esistenti come destinazioni 89
- tabelle orarie
  - struttura 501
  - utilizzo 79
- tabelle partizionate, replica 32
- tabelle partizionate in base all'intervallo, replica 32
- tabelle principali (aggiornamenti)
  - ricattura delle modifiche 50
- tabelle principali (aggiornamento ovunque)
  - panoramica 87
- tabelle segnali giornale
  - arresto 190
  - CAPSTOP 193
- Tablelle Unicode 505
- table differencing utility 209, 302, 307
- table repair utility 214, 317
- tables
  - AUTHTKN (Apply-qualifier cross-reference) 416
  - IBMSNAP\_ALERTS 481
  - IBMSNAP\_APPLY\_JOB 454
  - IBMSNAP\_CONDITIONS 483
  - IBMSNAP\_CONTACTGRP 488
  - IBMSNAP\_CONTACTS 489
  - IBMSNAP\_GROUPS 490
  - IBMSNAP\_MONENQ 490
  - IBMSNAP\_MONPARMS 490
  - IBMSNAP\_MONSERVERS 493
  - IBMSNAP\_MONTRACE 494
  - IBMSNAP\_MONTRAIL 494
  - IBMSNAP\_REG\_EXT 434
  - IBMSNAP\_SUSPENDS 496
  - IBMSNAP\_TEMPLATES 497
- target tables
  - repairing 214
- tipi di dati
  - associazione tra le colonne 91
  - replica
    - LOB (large objects) 96
- tipi di dati astratti 95
- tipi di dati definiti dall'utente 95
- tipi di dati esterni 95
- tipi di dati LONG VARCHAR 95
- tipi di dati LONG VARGRAPHIC 95
- tipi di dati spaziali 95
- tipi di dati speciali
  - replica
    - LOB (large objects) 96
- tips
  - using stored procedures with ASNDONE 147

- trace facility
  - for System i 402
- trace\_limit parameter
  - Replication Alert Monitor 233
  - use with asnmon command 285
- TRACE parameter 390
- traduzione dati 11
- transazioni
  - memoria utilizzata da 1
- trasferimento file
  - memorizzazione per Apply 7
  - memorizzazione per Capture 6
- trasformazione dei dati
  - creazione di colonne calcolate 107
  - durante la registrazione 105
  - durante la sottoscrizione 105
  - ridenominazione delle colonne 91, 107
- TRCLMT parameter 398
- trigger
  - acquisizione dati 9
  - integrazione 9
- trigger Capture
  - comunicazioni con
    - Centro di replica 245
    - programma Apply 248
    - Programma Apply 245
  - conflitti con trigger preesistenti 9
  - nomi di 9
  - pianificazione 9
  - requisiti di autorizzazione 17
- TRLREUSE parameter 392
- troubleshooting commands
  - WRKDPTRTC 402
- TSO 157, 159

## U

- unioni come origini 58
- unioni interne come origini 58
- user IDs, storing 17
- USER parameter 389
- Utilità ASNPLXFY 163
- utilities
  - table differencing 302, 307
  - table repair 317

## V

- valori predefiniti
  - per parametri Apply (Linux, UNIX, Windows, z/OS) 134, 136
  - per parametri Apply (System i) 134
  - per parametri Capture (Linux, UNIX, Windows, z/OS) 112
  - per parametri Capture (System i) 112
  - per parametri Capture (UNIX, Windows, z/OS) 114
- variabile di ambiente
  - DB2CODEPAGE 11, 27
- variabile di ambiente DB2INSTANCE 27
- Variabile LANG
  - impostazione 11
- variabili d'ambiente DB2DBDFT 27
- variabili di ambiente
  - DB2CODEPAGE 11, 27

- variabili di ambiente (*Continua*)
  - DB2DBDFT 27
  - DB2INSTANCE 27
  - LIBPATH 27
  - Programma Capture 27
- variabili giornaliere
  - DB2CODEPAGE 11, 27
  - DB2DBDFT 27
  - DB2INSTANCE 27
- velocità di trasferimento
  - trigger Capture 9
- velocità di trasmissione
  - Programma Apply 256
  - Programma Capture 254
- velocità di trasmissione della transazione
  - trigger Capture 9
- viste
  - limitazioni 57, 59
  - modifica degli attributi 166
  - registrazione
    - come origini 59
    - panoramica 57
    - procedura 165
    - uso di ID di correlazione 58
- viste CD (Change-Data) 57
- viste DB2
  - registrazione 59

## W

- WAIT parameter 397
- warm start, Capture program for System i 396
- warmns startmode 114
- warmsi startmode 114
- Windows service
  - creating 297, 300
- Windows Service Control Manager (SCM)
  - asnslst command 301
  - description 239
  - listing replication services 301
- WRKDPTRTC command 402
- WRKJOB command 257
- WRKSBJOB command 257
- WRKSBSJOB command 257

## Z

- z/OS console
  - sending monitor alerts 222









Stampato in Italia

SC13-4147-02



Spine information:

IBM InfoSphere Replication Server

**Version 9.7**

**SQL Replication Guide and Reference**

