SYBASE®

User and Troubleshooting Guide

Replication Agent™ for DB2 UDB

15.0

[z/OS]

DOCUMENT ID: DC32005-01-1500-01

LAST REVISED: September 2007

Sybase, Inc., One Sybase Drive, Dublin, CA 94568.

Contents

About This Book

Audience

This book is intended for those who are responsible for:

- Administering a Replication Server system
- Administering an MVS system
- Administering a DB2 database
- Configuring Replication Agent™
- Troubleshooting Replication Agent

How to use this book

This book contains these chapters:

Chapter 1, “15.0 Replication Agent Overview,” explains the basic concepts of replication systems, the role of Replication Agent, the role that connectivity plays in replication systems, and how replication works in a data-sharing environment.

Chapter 2, “Replication Server Setup,” explains how to prepare a system for replication and what you need to know about datatype conversions before you begin.

Chapter 3, “Replication Agent Setup,” provides information about DB2 considerations that affect working with source tables. It also explains how to mark tables for replication.

Chapter 4, “Managing Replication Agent,” explains how to start up and shut down Replication Agent. It discusses Replication Agent startup problems, how to monitor LTM for MVS and Replication Server, and provides replication failure troubleshooting tips.

Chapter 5, “Known Issues and Error Recovery,” explains the known restrictions on Replication Agent and provides error recovery procedures.

Appendix A, “LTM for MVS Configuration Parameters,” discusses the LTM for MVS configuration file and the parameters that control the LTM for MVS component of Replication Agent.

Appendix B, “Replication Extract Messages,” lists and describes the messages issued by Replication Extract.

Appendix C, “LTM for MVS Messages,” lists the LTM for MVS error and infomational messages and suggests fixes for the errors they represent.

Appendix D, “MVS Console Messages,” lists Replication Agent error and informational messages and suggest fixes for the errors they represent.

Appendix E, “Running Multiple Replication Agents,” explains how to run multiple Replication Agents in a replication system.

Appendix F, “Creating User Exits,” provides information about the transaction programs written to take control at a determined point in a program (user exit).Replication Agent for DB2 UDB

Related documents

- Replication Agent for DB2 UDB *Release Bulletin* for OS/390
- Replication Agent for DB2 UDB *Installation Guide* for OS/390
- Replication Server documentation
- Adaptive Server Enterprise documentation

Other sources of information

Use the Sybase Getting Started CD, the SyBooks CD, and the Sybase Product Manuals Web site to learn more about your product:

- The Getting Started CD contains release bulletins and installation guides in PDF format, and may also contain other documents or updated information not included on the SyBooks CD. It is included with your software. To read or print documents on the Getting Started CD, you need Adobe Acrobat Reader, which you can download at no charge from the Adobe Web site using a link provided on the CD.
- The SyBooks CD contains product manuals and is included with your software. The Eclipse-based SyBooks browser allows you to access the manuals in an easy-to-use, HTML-based format.

  Some documentation may be provided in PDF format, which you can access through the PDF directory on the SyBooks CD. To read or print the PDF files, you need Adobe Acrobat Reader.

  Refer to the *SyBooks Installation Guide* on the Getting Started CD, or the *README.txt* file on the SyBooks CD for instructions on installing and starting SyBooks.
- The Sybase Product Manuals Web site is an online version of the SyBooks CD that you can access using a standard Web browser. In addition to product manuals, you will find links to EBFs/Maintenance, Technical Documents, Case Management, Solved Cases, newsgroups, and the Sybase Developer Network.

To access the Sybase Product Manuals Web site, go to Product Manuals at http://www.sybase.com/support/manuals/.

Sybase certifications on the Web

Technical documentation at the Sybase Web site is updated frequently.

❖ **Finding the latest information on product certifications**

1 Point your Web browser to Technical Documents at http://www.sybase.com/support/techdocs/.

2 Click Certification Report.

3 In the Certification Report filter select a product, platform, and timeframe and then click Go.

4 Click a Certification Report title to display the report.

❖ **Finding the latest information on component certifications**

1 Point your Web browser to Availability and Certification Reports at http://certification.sybase.com/.

2 Either select the product family and product under Search by Base Product; or select the platform and product under Search by Platform.

3 Select Search to display the availability and certification report for the selection.

❖ **Creating a personalized view of the Sybase Web site (including support pages)**

Set up a MySybase profile. MySybase is a free service that allows you to create a personalized view of Sybase Web pages.

1 Point your Web browser to Technical Documents at http://www.sybase.com/support/techdocs/.

2 Click MySybase and create a MySybase profile.

Sybase EBFs and software maintenance

❖ **Finding the latest information on EBFs and software maintenance**

1 Point your Web browser to the Sybase Support Page at http://www.sybase.com/support.

2 Select EBFs/Maintenance. If prompted, enter your MySybase user name and password.

3 Select a product.

4 Specify a time frame and click Go. A list of EBF/Maintenance releases is displayed.

   Padlock icons indicate that you do not have download authorization for certain EBF/Maintenance releases because you are not registered as a Technical Support Contact. If you have not registered, but have valid information provided by your Sybase representative or through your support contract, click Edit Roles to add the "Technical Support Contact" role to your MySybase profile.

5 Click the Info icon to display the EBF/Maintenance report, or click the product description to download the software.

Style conventions

Table 1 shows some of the style conventions used in the documentation for this product.

Table 1: Style conventions

Text	Explanation	Example
Bold	• Programs • Utilities • Procedures • Commands	create connection
Italics	• File names • Directory names • Properties	*confiuration file*
`Code`	• Code examples • Screen text	`Replication Agent for DB2/12.6`
Bold code	• User input • Command line input	*/F [job_name],C, trace=LTLebcdic*
Italics	Variables (replace these with the appropriate values for your site)	*host_name*
Code italics	Variables in code you type (replace these with the appropriate values for your site)	*maintenance_user_ID*

Syntax conventions

The following example illustrates some of the syntax conventions used in this guide:

```
COMMAND [object_name, [ {TRUE | FALSE} ] ]
```

Table 2 explains the syntax conventions used in this guide.

Table 2: Syntax conventions

Symbol	Explanation	Example
( )	Include parentheses as part of the command.	`START DATABASE (database_name)`
|	A vertical bar indicates that you can select only one of the options shown. Do not type the bar in your command.	`{red|yellow|blue}`
,	A comma indicates that you can choose one or more of the options shown. Separate each choice by using a comma as part of the command.	`{rice,potatoes,beans}`
{ }	Braces indicate that you must choose at least one of the enclosed options. Do not type the braces when you enter the option.	*Select only one:* `{red|yellow|blue}` *Select at least one:* `{rice,potatoes,beans}`
[ ]	Brackets indicate that you can choose one or more of the enclosed options, or none. Do not type the brackets when you enter the options.	`[anchovies]`
...	An ellipsis indicates that you can repeat the previous item as many times as necessary.	`{rice,potatoes}...`

If you need help

Each Sybase installation that has purchased a support contract has one or more designated people who are authorized to contact Sybase Technical Support. If you cannot resolve a problem using the manuals or online help, please have the designated person contact Sybase Technical Support or the Sybase subsidiary in your area.

CHAPTER 1

15.0 Replication Agent Overview

This chapter provides an overview of Replication Agent for DB2 UDB for OS/390 version 15.0 (hereafter referred to as Replication Agent).

If you are completely familiar with replication concepts, your MVS environment, and Replication Server, turn now to "Understanding the role of Replication Agent" on page 12.

Note The material in this chapter also appears in the Replication Agent for DB2 UDB *Installation Guide* for OS/390.

Replication Agent task and documentation topics overview

Table 1-1 lists the locations in the Replication Agent documentation of the tasks involved in installing, using, and troubleshooting Replication Agent. Each of the guides listed in this table includes a preface, this overview chapter, and a glossary of technical terms.

Table 1-1: Where to find Replication Agent documentation topics

Replication Agent for DB2 UDB Installation Guide for OS/390

Plan your Replication Agent installation

Includes information about installation preparation, including software requirements, operational considerations, and filling out the Replication Agent Installation Worksheet:

- Go to Chapter 2, “Understanding Replication Agent installation requirements”
- Go to Chapter 3, “Understanding Replication Agent operational considerations”
- Go to Chapter 4, “Preparing for installation”

Install Replication Agent

Includes instructions for installing the Replication Agent software, granting permissions, and configuring Replication Extract.

- Go to Chapter 5, “Installing Replication Agent”
- Go to Chapter 6, “Configuring Replication Extract”

Set up communications

Includes information about setting up TCP/IP communications between Replication Agent and Replication Server:

- Go to Chapter 7, “Configuring TCP/IP for Replication Agent”

Verify your installation

Includes information about setting required configuration parameter values, and testing connections with the PING and PINGRS utilities:

- Go to Chapter 8, “Verifying Your Replication Agent installation”

Reference information

- Appendix A, “LTM for MVS configuration parameters”
- Appendix B, “Replication Extract configuration parameters”
- Appendix C, “Replication Agent distribution tape contents”
- Appendix D, “Upgrading your Replication Agent software”
- Appendix E, “Running multiple Replication Agents”
- Appendix F, “Modifying Replication Agent for DB2 non-data-sharing mode”

Replication Agent for DB2 UDB User’s and Troubleshooting Guide for OS/390

Start and stop Replication Agent

Includes information about starting, stopping, and troubleshooting Replication Agent using trace commands:

- Chapter 4, “Managing Replication Agent”

Prepare for replication

Includes information about preparing your replication environment for Replication Agent operations, including Replication Server setup, materializing replicate tables, and marking source data for replication:

- Chapter 2, “Replication Server Setup”
- Chapter 3, “Replication Agent Setup”

Work with Replication Agent system tables Includes information about using the Replication Agent system tables on DB2 to coordinate bulk materialization and mark tables for replication: • Chapter 3, “Replication Agent Setup”
Work with datatype conversions Includes information about LTM for MVS datatypes, Adaptive Server datatypes, and Replication Agent datatype conversion handling: • Chapter 2, “Replication Server Setup”
Troubleshoot Replication Agent Includes information about basic troubleshooting procedures, start-up problems, restrictions, and failure and recovery: • Chapter 5, “Known Issues and Error Recovery”
Understand Replication Agent messages Includes information, error, and warning messages written to log files and to the MVS operator console, accompanied by recommendations for user responses: • Appendix B, “Replication Extract Messages” • Appendix C, “LTM for MVS Messages” • Appendix D, “MVS Console Messages”
Reference • Appendix A, “LTM for MVS Configuration Parameters”

Understanding basic concepts of replication systems

Transaction replication maintains data in separate databases called replicate databases. Replicate databases contain accurate, current copies or subsets of data from the primary database.

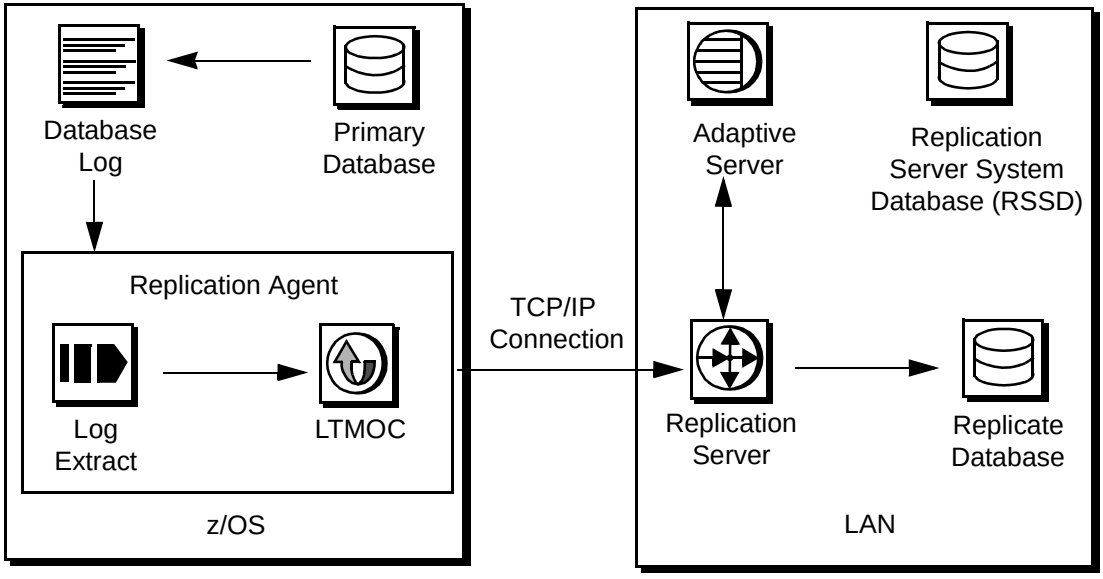
Replication of data allows slower, process-intensive decision support queries to run against the same data without slowing mission-critical, on-line transaction processing (OLTP) applications.

Figure 1-1 on page 3 shows a generic replication system and its components.

Figure 1-1: Replication system overview

The following sections introduce the basic concepts of transaction replication and describe the following elements of a typical replication system:

- Primary database
- Primary database log



- Replication Extract
- Log Transfer Manager for MVS
- Replication Server

Primary database

The primary database, DB2, contains your source data.

Primary database log

The DB2 log contains information about updates, deletes, and inserts on DB2 tables. The DB2 log contains information about additional changes and events; this additional information is not currently replicated.

DB2 logs changes to rows in DB2 tables as they occur. The information written to the log includes copies of the data before and after the changes. In DB2, these records are known as “undo” and “redo” records.

Control records are written for begins, commits, and aborts. These records are translated to begins, commits, and rollbacks.

The DB2 log is composed of a series of data sets. Replication Extract uses these log data sets to identify DB2 data changes. The DB2 log consists of three types of data sets:

- Active log data sets reside on a direct access storage device (DASD). As changes occur, they are first recorded in an active log data set. Eventually, the active log fills and DB2 copies the log to an archive log.
- Archive log data sets reside on DASD or tape.
- The bootstrap data set (BSDS) contains information about all of the active and archive data sets that make up the log records.

Since DB2 writes change records to the active log as they occur, Replication Extract can process the log records immediately after they are entered.

Typically, DB2 writes two copies of the logs for redundancy in the event of log data set corruption.

DB2 log changes

Replication requires an entire “before” and “after” image of the row that has been changed by an update. When you mark a table for replication using the Replication Agent interface dialog panels, the table is altered with the DATA CAPTURE CHANGES clause. As the number of DB2 tables marked for replication increases, so do the DASD space requirements for the DB2 active log data sets.

There are no increases in the amount of data written to the log for deletes or inserts, because for deletes and inserts, the entire “before” image and “after” image is captured in the log.

When a table receives an update, DB2 has to log before and after copies of the data in case it is necessary to restore that data. DB2 writes nine different types of update log records to minimize the amount of data written to the logs.

The additional DB2 log usage caused by using DATA CAPTURE CHANGES can be significant if the row size is large or if the number of updates is large.

DB2 also minimizes the amount of data written to a log record by writing only data from the first changed byte in the table row up to the end of the row. The entire copy of the table row before the change is then copied to the log record.

The effect of the change is minimal if the number of updates is small, if the row size is small, or if the updates are normally applied near the end of the row.

DB2 does not log transactions for an update in which the entire new table row matches the entire old table row. Replication Agent does not replicate transactions of this nature because the transactions do not appear in the DB2 logs.

Begin records can be written long before data records are written to the log file and transaction data can be interleaved. Because this is the case, begin records are not sent to the Replication Server until just before the first data record is sent. Begin records use information from the data record to keep the log sequence in ascending order.

There are multiple commit records and multiple abort records. A commit is sent to the Replication Server when the final commit is received; rollbacks are sent when the final rollback record is received.

DB2 active and archive logs

Be sure to protect DB2 active and archive logs against loss or corruption.

To optimize Replication Agent performance, be sure that the DB2 active logs are large enough to contain the number of log entries for a 12-hour period of update activity before reusing the log data set. This allows you to resume replication using log data following network outages that might interrupt replication, without requiring the log extract to read log entries from tape.

Replication Extract

Log extracts read and transfer relevant database transactions recorded in the database logs to LTM for z/OS. The log extract in Replication Agent is the Replication Extract.

A DB2 table named LTMOBJECTS is created during the installation of Replication Agent. This table contains a row corresponding to each table that is marked for replication. Replication Extract uses the LTMOBJECTS table to identify the data marked for replication.

To capture DB2 data changes, Replication Extract uses the DB2 IFI API to retrieve log records that may originate in the DB2 active and archive logs for inserts, updates, and deletes on tables marked for replication in LTMOBJECTS.

Replication Extract formats the extracted changes, stores the changes in multiple large buffers, then continues to read the DB2 logs.

A separate thread (LTMOC) running in the Replication Agent Address space waits for buffers to fill before transmitting buffers to the Replication Server. This allows the Log Extract and the LTMOC thread to be tuned separately, depending on z/OS, DB2, or network activity.

Replication Extract and LTM for MVS run simultaneously within a single MVS address space as an LTM for MVS started task, continuously reading the DB2 logs.

Replicating transactions from multiple DB2 subsystems can be accomplished in two ways:

- Use one Replication Agent started task for each DB2 subsystem.
- Use multiple members of a data-sharing group with a single Replication Agent.

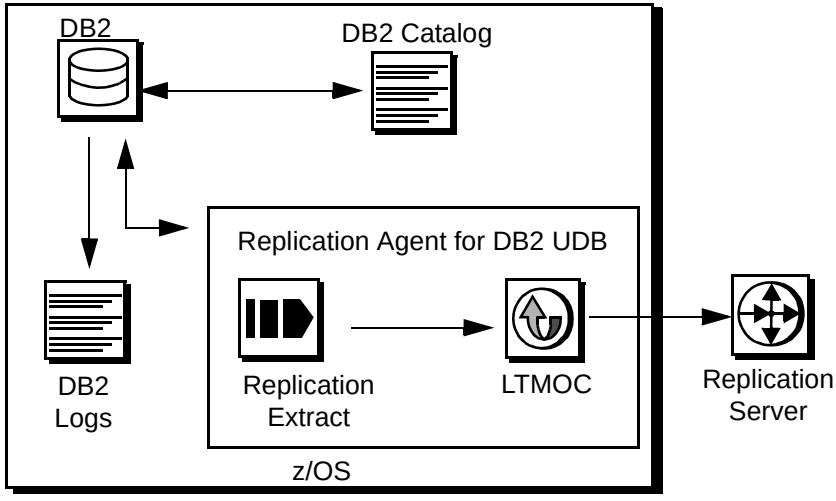
With the first option, each Replication Agent started task processes the log of a single DB2 subsystem.

With the second option, all logs are processed by DB2 and records are passed to the Replication Agent as if there were only one log.

Replication Extract processing flow

The following figure shows the processing flow between DB2, Replication Extract, LTM for MVS, and Replication Server:

Figure 1-2: Replication Extract processing flow



The processing flow for Replication Extract includes the following actions:

1 The MVS START command, which you issue from the MVS console, starts the LTM for MVS started task. (Replication Agent can also be started as a batch job.)

   a LTM for MVS processes the configuration file and retrieves information about Replication Server, Replication Extract, user exits, defaults, DB2, traces, and DB2 subsystems.

   b LTM for MVS opens a connection to Replication Server.

      If the connection to Replication Server fails, LTM for MVS retries the connection (according to the value of the retry and Maximum_connect_retries configuration parameters).

2 LTM for z/OS queries the Replication Definitions if configured to do so, and starts the Rep Server transmission thread using the z/OS ATTACH command.

3 Replication Extract connects to the DB2 subsystem using the DB2 CONNECT command. The DB2 subsystem must be active for the entire time the Replication Agent started task is active.

4 Replication Extract queries the LTMOBJECTS system table to identify the tables marked for replication.

5 Replication Extract queries the DB2 catalog to retrieve the definitions of all the tables recorded in the LTMOBJECTS table. These definitions are required to format the changes described in the DB2 log.

6 Replication Extract queries LTM for MVS for the origin queue ID (QID) of the last update that was successfully received by Replication Server, and writes a copy of the QID to the *hlq.TRUNCPT* data set.

   This QID is called the LTM Locator, and is stored in the Replication Server database, in the rs_locater table. From this QID, Replication Extract derives the point within the log from which processing is to resume.

   If the QID that LTM for MVS passes to Replication Extract and the QID in the *TRUNCPT* data set contain all binary zeros, Replication Extract begins processing from the end of the active log.

7 Replication Extract resumes processing at the point derived from the LTM Locator and begins scanning the log (or logs, if Replication Agent is running in data-sharing mode) for changes to DB2 tables.

8 Using the log records that describe inserts, updates, and deletes to primary tables, Replication Extract constructs a full-row "after" image for inserts, a full-row "before" image for deletes, and full "before" and "after" images for updates for each affected row.

The individual column values are converted from DB2 internal format to their external format. Replication Extract then passes the converted transaction operation information to LTM for MVS.

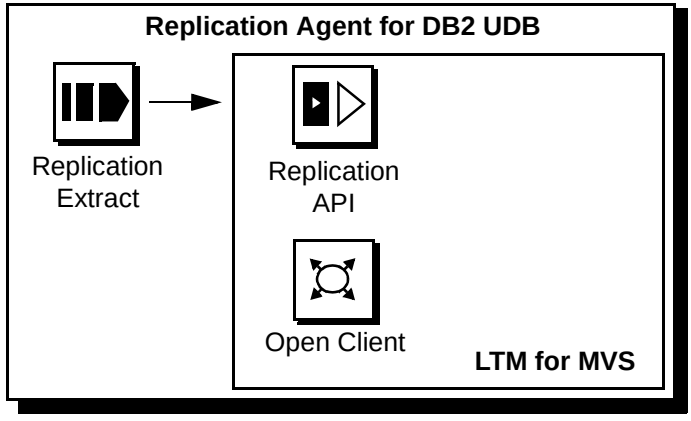
Log Transfer Manager for MVS

Log Transfer Manager receives relevant database transactions from log extracts. LTM for MVS translates this information into Log Transfer Language (LTL), and sends these LTL statements through a connection to Replication Server.

LTM for MVS contains the following interfaces:

- Replication API (Replication Application Program Interface, LTMAPI)
- Sybase Open ClientConnect™ routines that allow LTM for MVS to communicate with Replication Server

Figure 1-3: LTM for MVS components



LTM for MVS translates the transaction information received from Replication Extract into Log Transfer Language (LTL), and sends these LTL statements to the primary Replication Server.

Replication Server holds the LTL statements in a stable queue, transforms these transactions (usually into SQL, according to user instructions), and passes the transformed transaction information to the replicate Replication Servers or replicate database(s) for processing.

LTM for MVS processing flow

LTM for MVS processes as follows:

1 Logs in to the primary Replication Server.

2 Sends a connect source command to identify the session as a log transfer source and to specify the database log to be transferred.

3 Obtains the maintenance user ID from Replication Server. LTM for MVS uses the Replication Server maintenance user ID to filter operations executed by the maintenance user.

4 Requests the origin queue ID from Replication Server. The origin queue ID contains the LTM Locator value.

   Replication Extract (LTMEXT) uses the LTM Locator value to determine the point in the log at which LTM for MVS should begin transferring transaction operation records. Replication Server successfully received all log records up to this location.

5 Receives transaction operations from Replication Extract following the point in the log indicated by the LTM Locator value and formats them into LTL commands.

What you need to know

Maintenance user ID	The Replication Server get maintenance user command returns the Replication Server maintenance user ID. Replication Server uses the maintenance user ID to update tables with replicated data in the database monitored by Replication Agent. By default, Replication Agent does not pass transactions made by the maintenance user ID to Replication Server for replication. To replicate changes made by this user, set the value of the LTM for MVS LTM_process_maint_uid_trans configuration parameter to Y to specify that Replication Agent replicate all logged changes made by the Replication Server maintenance user.

LTM Locator and origin queue ID	The Replication Server get truncation command returns a value called the LTM Locator, which is the origin queue ID associated with the transaction operation last saved by Replication Server. Replication Extract uses the LTM Locator value to determine the location in the log where replication should begin. Replication Server has successfully saved all transactions up to this LTM Locator value in its stable queue; Replication Agent can safely ignore all transactions before this point in the log.

Replication Server

Replication Server transfers database transactions to the replicate database using a connection to the replicate database. Replication Server can replicate to a single database, multiple databases, or other Replication Servers.

Replication Server holds the transaction operations in a stable queue and delivers them as soon as possible to other Replication Servers (replicate Replication Servers) or replicate databases. These transactions are transformed (usually into SQL) according to the instructions you provide to Replication Server and are sent to the replicate database for processing.

Replication Server guarantees transaction delivery: Every transaction successfully received from a LTM is guaranteed to be delivered to appropriately subscribing replicate databases.

Replication Server systems are administered through a LAN-based interface. The System Administrator creates subscriptions for replication definitions on Replication Server.

The subscriptions and replication definitions define the columns and rows that Replication Server manages at the replicate sites. The System Administrator also defines destinations and routes to manage network traffic.

Materialization

Materialization is the process of initially populating the replicate database with a copy of the data from the primary database. Replication Agent requires that you set up and populate each replicate database so that it is in a state consistent with that of the primary database.

Replication Server then replicates incremental changes to the primary data from the primary database to the replicate database.

See also

- Chapter 2, "Replication Server Setup," for information about materializing replicate databases
- Replication Server *Administration Guide*

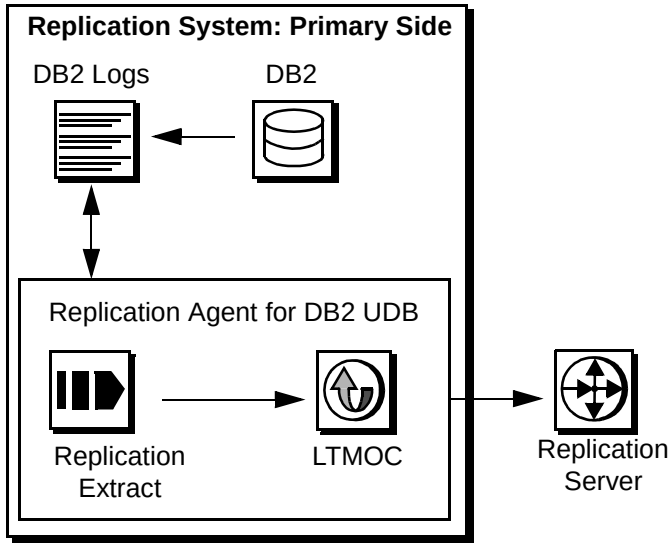
Understanding the role of Replication Agent

Replication Agent is a component of a replication system that moves primary database transactions from DB2 through Replication Server to replicate databases.

Replication Agent fits into the generic replication system, illustrated in Figure 1-4, with the following differences:

- With Replication Agent, the primary database is DB2, which runs as a subsystem in MVS. The database logs are DB2 logs.
- Replication Agent provides a log extract, called Replication Extract, that reads the DB2 logs and retrieves the relevant DB2 active and archive log entries for tables marked for replication.
- The LTMAPI send function receives the data marked for replication from Replication Extract and LTMOC transfers this data to Replication Server using the TCP/IP communications protocol.
- Replication Server then applies the changes to the replicate databases.

Figure 1-4: Replication Agent data flow



See also

- Replication Server *Administration Guide* and Replication Server *Design Guide* for information about additional replication system topics

Replication Agent data flow

The Replication Agent data flow consists of four primary steps:

Step 1: Replication Extract reads log data.

Step 2: Replication API translates log data into LTL.

Step 3: LTMOC (Open Client) delivers information.

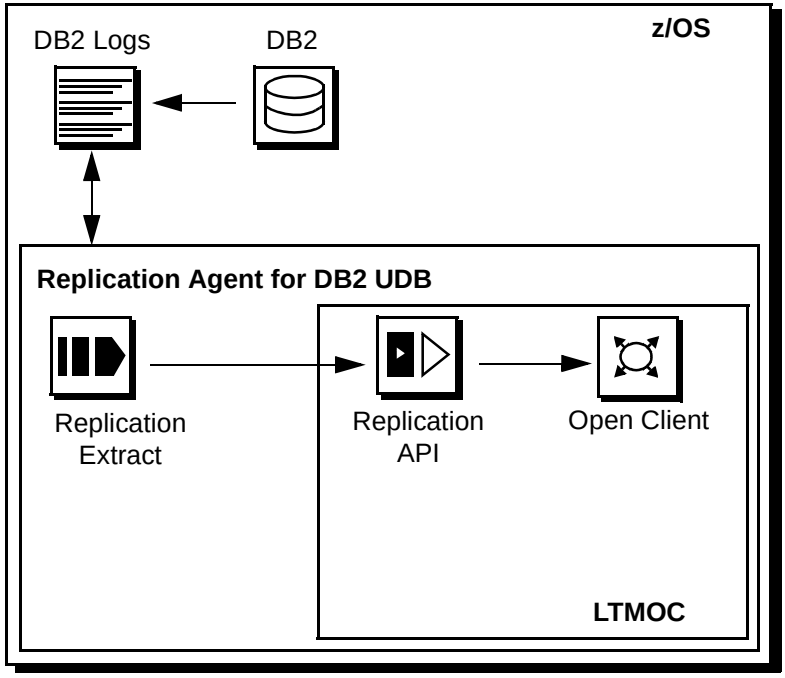
Step 4: Replication Server replicates information.

Step 1: Replication Extract reads log data

Replication Extract reads the DB2 active and archive log data sets for changes to DB2 tables marked for replication. It retrieves all the relevant information and delivers it to the Replication API, a component of LTM for MVS.

Figure 1-5 illustrates this process.

Figure 1-5: Replication Extract reads log data

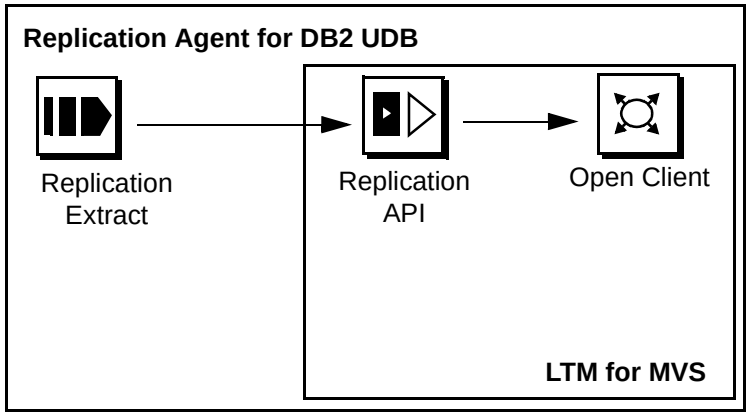


Step 2: Replication API translates log data into LTL

The Replication API receives information from Replication Extract, translates the command structures received from the log extract into Log Transfer Language (LTL) statements, and places the LTL in a buffer.

Figure 1-6 illustrates this process.

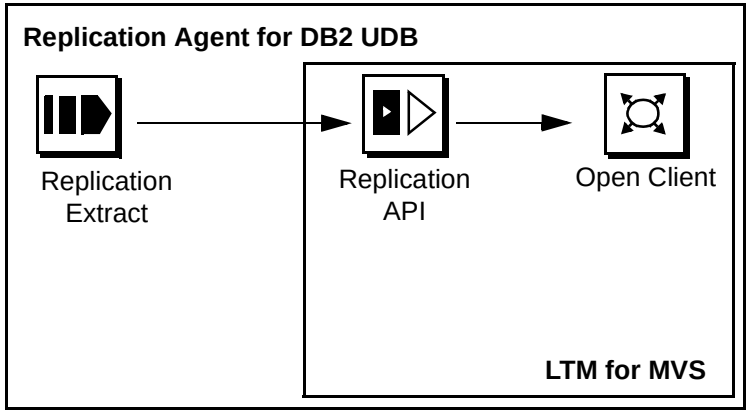
Figure 1-6: Replication API translates information



Step 3: LTMOC (Open Client) delivers information

The LTMOC delivers the buffer to the Open ClientConnect for MVS routines in LTM for MVS, as shown in Figure 1-7.

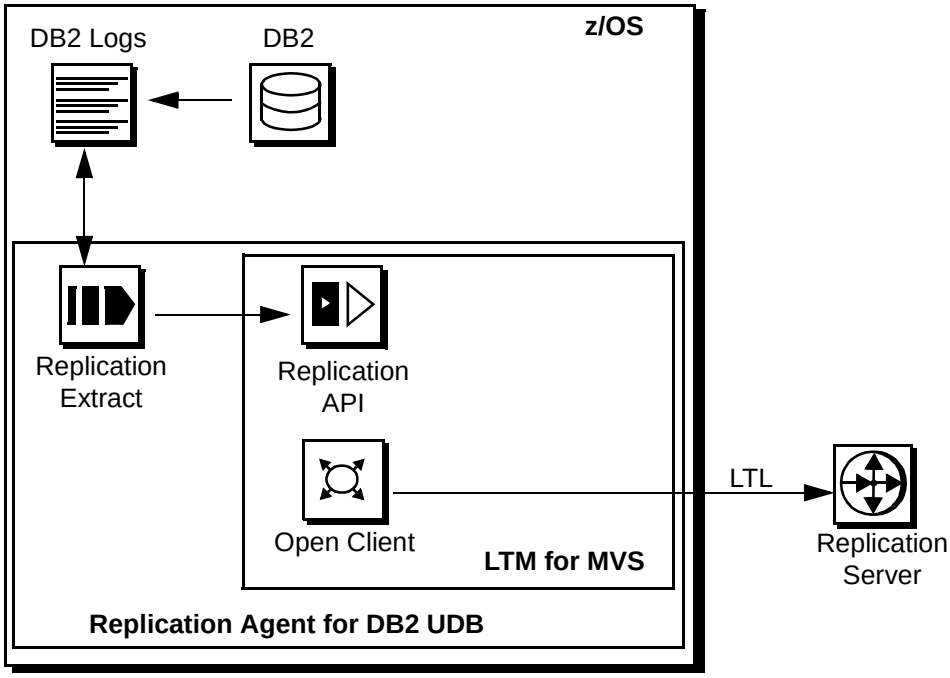
Figure 1-7: LTMOC translates information for Open Client



Step 4: Replication Server replicates information

LTM for MVS delivers LTL to Replication Server over the network. Replication Server translates the LTL statements into transactions that replicate the primary database changes to the replicate databases, as shown in Figure 1-8.

Figure 1-8: Replication Server receives transactions for replication



See also

- "Understanding connectivity in a replication system" for information on Replication Agent-to-Replication Server connectivity
- Replication Server *Administration Guide* and the Replication Server *Design Guide* for more information about Replication Server functionality

Understanding connectivity in a replication system

Replicating transaction operations from MVS to Replication Server requires communication between different operating systems. Each site has two sets of connectivity options:

- Replication Agent-to-Replication Server connectivity
- Replication Server-to-DB2 connectivity

Replication Agent-to-Replication Server connectivity

Replication Agent communicates with Replication Server using the MVS TCP/IP protocol.

Replication Server-to-DB2 connectivity

Replication from Replication Server-to-DB2 replicate databases can use a variety of gateways.

Three-tier architecture

You can use Sybase DirectConnect™ for DB2 for gateway connectivity to allow Replication Server to issue selects, inserts, updates, or deletes against the tables stored in DB2.

Two-tier architecture

You can set up Replication Server-to-DB2 connectivity without a DirectConnect for DB2 gateway. For details on how to accomplish this, see the Replication Server *Heterogeneous Replication Guide*.

See also

- Chapter 3, “Understanding Replication Agent Operational Considerations” in the Replication Agent for DB2 UDB *Installation Guide* for OS/390, for information about connectivity options for replication into DB2
- The World Wide Web address *http://www.sybase.com/Partners/certification/index.html* for information on replicating into DB2
- Chapter 2, “Replication Server Setup,” for information on bulk materialization

Replication Agent users

Several user identifiers are required for working with Replication Agent. The following table describes each one and its requirements.

Table 1-2: User IDs associated with Replication Agent

User ID	Description	Restrictions
LTMADMIN user	Sybase recommends that you create the LTMADMIN user ID, a TSO user optionally named LTMADMIN, to perform the following tasks: • Install, start, and stop Replication Agent • Manage the Replication Agent system tables on DB2 • Bind the log extract plan	This identifier can be a TSO user ID or a group identifier. MVS user ID restrictions apply to this identifier (eight characters maximum, all uppercase). This user ID should not be identical to the Replication Server maintenance user ID.
Replication Server maintenance user	The maintenance user is the Replication Server user in the create connection command used to create a connection from Replication Agent to Replication Server. Any updates applied to the primary database by the maintenance user are ignored for replication unless the value of the LTM_process_maint_uid_trans LTM for MVS configuration parameter has a value of Y. The Replication Server get maintenance user command returns the Replication Server maintenance user ID.	Any updates applied to the primary database by the maintenance user are ignored for replication unless the value of the LTM for MVS LTM_process_maint_uid_trans configuration parameter has a value of Y. Because this user ID is intended to compare against a user ID that can perform updates on DB2, it must conform to MVS user ID restrictions (eight characters maximum, all uppercase). **Note** To avoid cyclic replication, this user ID should not be identical to the value of the LTM for MVS RS_user configuration parameter (the RS user).
RS user	The RS user is the user ID specified in the value of the RS_user configuration parameter. This is the user ID that Replication Agent uses to connect to the primary Replication Server. See "Task 1a: Create the RS_user on Replication Server" on page 32 for details.	This user ID can be a maximum of 30 characters. **Note** This user ID should not be identical to the Replication Server maintenance user ID.
RSSD user	The RSSD user is the user ID specified in the value of the RSSD_user configuration parameter. This is the user name for the Adaptive Server user who has permissions to perform selects against the primary Replication Server's RSSD.	This user ID can be a maximum of 30 characters.

Understanding replication in a data-sharing environment

When you are using DB2 with data sharing enabled, you can replicate transactions from one or more DB2 subsystems within a data-sharing group.

This chapter contains the following tasks:

- Understanding data-sharing environment characteristics
- Replicating transactions in a data-sharing environment

Understanding data-sharing environment characteristics

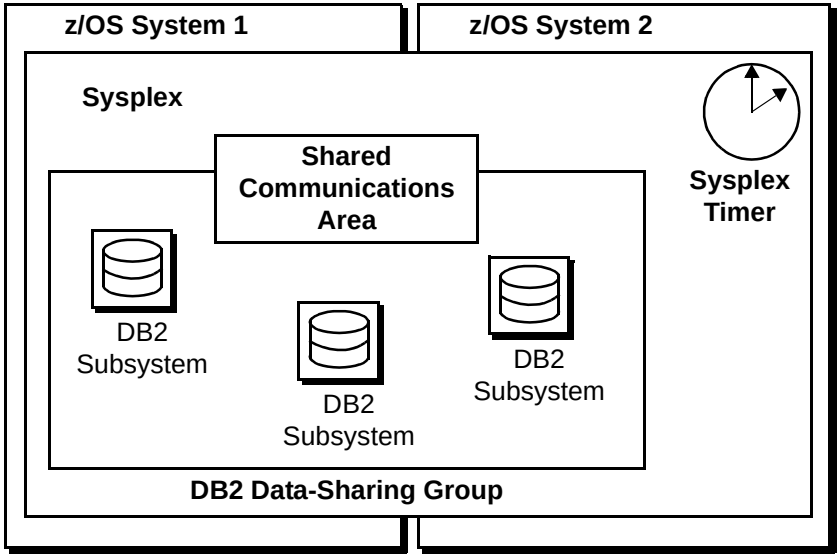
DB2 installations can have multiple MVS sysplexes. Each sysplex can contain multiple DB2 data-sharing groups. A DB2 data-sharing group is an MVS Cross-system Coupling Facility (XCF) group.

Each member of a DB2 data-sharing group is a DB2 subsystem, also called a member. The members within a single DB2 data-sharing group can share data. DB2 data sharing does not function across groups. The Sysplex Timer synchronizes timestamps between all DB2 subsystems in a data-sharing group.

The members of a DB2 data-sharing group can reside on the same or different MVS systems. Therefore, a DB2 data-sharing group can either span several MVS systems or reside on a single MVS system.

Each DB2 subsystem maintains its own recovery logs and bootstrap data set (BSDS). The BSDS for every DB2 member in a data-sharing group contains information about the other group members' BSDSs. The shared communications area (SCA) in the XCF for a given data-sharing group contains information about all BSDSs and all logs for DB2 subsystems within the group.

Figure 1-9: Illustration of data-sharing group components



Each log record header contains a log record sequence number (LRSN). The LRSN is a 6-byte value derived from the Store Clock timestamp. The LRSN is equal to or greater than the timestamp value truncated to 6 bytes. This value also exists in the header page of the tablespace. In the data-sharing environment, the LRSN replaces the relative byte address (RBA), although each subsystem continues to use the RBA for its own logs and BSDS.

Replicating transactions in a data-sharing environment

When data sharing is enabled, you can replicate transactions from a single member of a data-sharing group with a single Replication Agent started task. Replication Server allows a single Replication Agent to replicate transactions from multiple members of a data-sharing group.

To replicate transactions when using DB2's data-sharing environment, you must supply specific parameter values in the LTM for MVS Log_identifier configuration parameter.

See also

- Chapter 6, "Configuring Replication Extract," in the Replication Agent for DB2 UDB *Installation Guide* for OS/390 for information about configuring the Rep Agent and binding the log extract plan.

- Appendix A, “LTM for MVS Configuration Parameters,” for information on the LTM for MVS *Log_identifier*, *DataSharingOption*, and *DataSharingMember* configuration parameters. To replicate transactions using DB2’s data-sharing environment, you must supply specific values for these parameters.

CHAPTER 2

Replication Server Setup

This chapter explains how to prepare a system for replication and what you need to know about datatype conversions before you begin.

Replication considerations

This section provides information about the datatype conversions that occur when DB2 source data is replicated into replicate tables. Because Replication Server uses Adaptive Server datatypes, you must consider the transformations that can occur in mapping DB2 source data to Sybase LTM for MVS and ASE datatypes.

Mapping source data to LTM for MVS datatypes

LTM for MVS datatypes are nearly identical to DB2 datatypes.

Table 2-1 lists valid LTM for MVS datatypes.

Table 2-1: LTM for MVS datatypes and lengths

LTM for MVS datatype	Description	Length (bytes)
binary	Binary.	1–254
char	Character, fixed length.	1–255

LTM for MVS datatype	Description	Length (bytes)
char for bit data	Character, fixed length. Replication Agent replicates char for bit data as binary data. **Note** In this case, the target datatype must be binary or varbinary. The data is not transformed from EBCDIC to ASCII format, but is replicated as is (in binary format).	1–254
varchar	Character, varying length. Actual length = maximum + 2.	1–255
integer	Integer.	4
smallint	Integer.	2
decimal	Packed decimal. The length indicates the number of decimal digits, where the actual length = ((number of digits +1)÷2), rounded up.	1–16
float4	Floating decimal. **Note** LTM for MVS working with Replication Server can guarantee a precision of 15 digits for float. Depending on your system hardware, the 16th digit of float rounds unpredictably.	4
float8	Floating decimal.	8
date	Date.	10
time	Time.	8
timestamp	Date and time.	26
db2 graphic	DB2 graphic is replicated as binary. **Note** LTM for MVS truncates GRAPHIC fields at 127 characters (binary data contains 2 bytes per character).	254
db2 vargraphic	DB2 graphic is replicated as binary. **Note** LTM for MVS truncates graphic fields at 127 characters (binary data contains 2 bytes per character).	1–254
long varchar	Long character, varying length.	1–32704*

*Replication Agent limits long varchar field lengths to 32704 bytes when Minimal_cols=N. When Minimal_cols=Y, Replication Agent limits varchar fields to 250 bytes and long varchar fields to 32700 bytes.

Mapping LTM for MVS datatypes to ASE datatypes

Replication Server requires your data to be compatible with ASE datatypes. Specify ASE datatypes for target table columns as needed when you create replication definitions on Replication Server.

Figure 2-1 illustrates the process of mapping your LTM for MVS data to ASE datatypes.

Figure 2-1: Mapping LTM for MVS datatypes to ASE datatypes

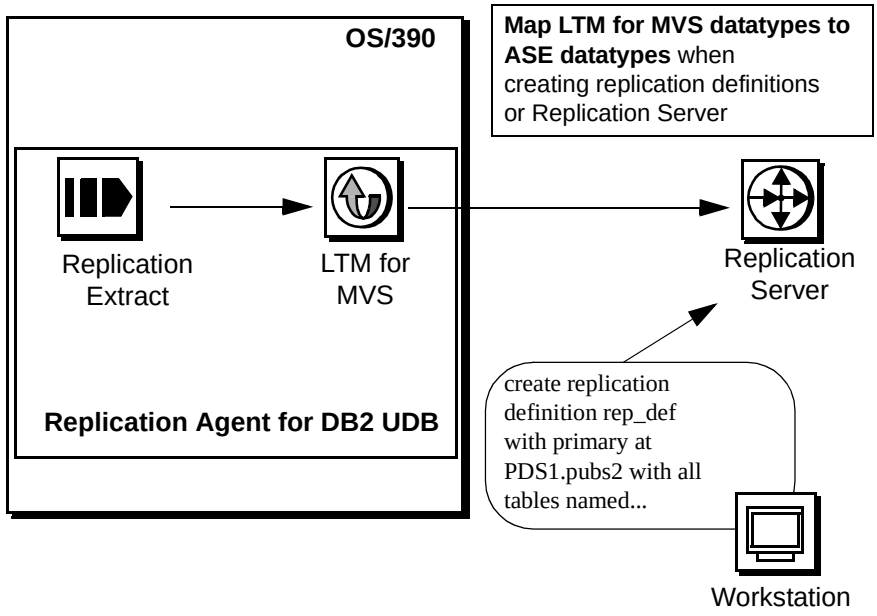


Table 2-2 shows the ASE datatypes you can specify in replication definitions.

Table 2-2: LTM for MVS to ASE datatype conversion

LTM for MVS datatype and example	Length (bytes)	ASE datatype and example
binary	250 or less	binary(), varbinary()
char hello	250 or less	char(), varchar() hello
char for bit data x'F1F2F3'	250 or less Truncates after 250 characters.	char(), varchar() 123
varchar hello	250 or less Truncates after 250 characters.	varchar() hello

LTM for MVS datatype and example	Length (bytes)	ASE datatype and example
smallint 22	2	smallint, int 22
integer -22	4	int -22
float 1.2E3	Either: • 4 (single precision) • 8 (double precision)	float 1.2E3 **Note** LTM for MVS working with Replication Server can guarantee a precision of 15 digits for float. Depending on your system hardware, the 16th digit of float rounds unpredictably.
decimal 1.200	31 or less	float or decimal 1.200
date 1994-01-01	10 Map dates below 1753 to char.	datetime + default time Jan 1 1994 12:00AM **Note** See "Support for heterogeneous datatypes" on page 27 for additional information on mapping the date datatype.
time 10:02:30	8	datetime + default date Jan 1 1990 10:02:30AM **Note** See "Support for heterogeneous datatypes" on page 27 for additional information on mapping the time datatype.

LTM for MVS datatype and example	Length (bytes)	ASE datatype and example
timestamp 1994-01-01-10.02.30.006123	26	datetime Jan 1 1994 10:02:30:006AM **Note** See "Support for heterogeneous datatypes" on page 27 for additional information on mapping the timestamp datatype.

See Appendix A, "LTM for MVS Configuration Parameters," for information about the replace_null_in_char and Codepage configuration parameters.

Support for heterogeneous datatypes

Replication Server version 12.0 and later includes Heterogeneous Datatype Support (HDS). This means that you can now use DB2 date, time, and timestamp datatypes without having to map them to ASE datatypes.

Note If you have an existing DB2 database configured as a replicate database with an earlier version of Replication Server, do not install the DB2 function strings provided with Replication Server version 12. The new function strings may not be compatible with earlier versions of these function strings, so you should continue using the earlier versions of the DB2 function strings with Replication Server version 12 and its HDS feature.

For more information on setting up Replication Server and the HDS feature, see the Replication Server *Configuration Guide*.

What you need to know

Default date ranges	Every time LTM for MVS starts, it checks default values for the Date_conv_default and Time_conv_default parameters in the LTM for MVS configuration file. The valid date and time value ranges include: • Year: 1753 – 9999 • Month: 01 – 12 • Day: 01 – 31 • Hour: 00 – 24 • Minute: 00 – 59 • Second: 00 – 59 **Note** ASE does not accept year values earlier than 1753. Set the value of the Date_in_char LTM for MVS configuration parameter to Y (yes) to pass year values outside this range as char data.
float approximate datatypes	LTM for MVS supports two versions of float in the standard MVS floating point representation. Different operating system platforms handle float datatypes differently. Depending on your system hardware, the 16th digit of float rounds unpredictably. Check the documentation for the source and target ASE operating systems and compare it to the LTM for MVS datatype handling documentation in this chapter.
decimal and Replication Server float datatypes	LTM for MVS does not convert decimal to float, but leaves it as LTM for MVS decimal. The float datatype on Replication Server is accurate to only 15 digits depending on the platform you use.
Replication Agent handling of char for bit data columns	Replication Agent replicates char for bit data in binary format.
graphic and vargraphic columns	graphic or vargraphic datatypes are replicated as BINARY. LTM for MVS truncates these fields at 127 characters (2 bytes per character).

Replication Agent datatype conversion considerations

This section describes how Replication Agent handles date, time, and timestamp data conversions and responds to datatype conversion errors.

Understanding LTM for MVS date and time conversions

When you map data to the ASE datetime datatype, be sure you understand the data type conversion that can occur between LTM for MVS date, time, and timestamp data and ASE datetime data.

Note Replication Server version 12.0 and later includes Heterogeneous Datatype Support (HDS). This means that you can now use DB2 date, time, and timestamp datatypes without having to map them to ASE datatypes. For more information on setting up Replication Server and the HDS feature, see the Replication Server *Configuration Guide*.

ASE datetime data is accurate only up to one thousandth of a second. For example, if you map a time column in your primary table that contains 10.02.30.006123 to the ASE datetime datatype, the conversion results in a loss of the last three digits: 10:02:30:006AM.

You can prevent this precision loss by:

- Converting LTM for MVS date, time, and timestamp columns in your primary table to char format.
- Setting the following LTM for MVS configuration parameter values:
  - Time_in_char=Y (time columns are sent as char(8) fields instead of mapping to the ASE datetime datatype.)
  - Date_in_char=Y (date columns are sent as char(10) fields instead of mapping to the ASE datetime datatype.)
  - Timestamp_in_char=Y (timestamp columns are sent as char(26) fields instead of mapping to the ASE datetime datatype.)
- Writing a user exit to convert timestamp data to char format. This approach is ideal for handling subsets of source data.

See Appendix B, “Creating User Exits” in the Replication Agent for DB2 UBD *Installation Guide* for further information.

Datatype conversion error handling in LTM for MVS

LTM for MVS responds to datatype conversion errors differently depending on the values of several LTM for MVS configuration parameters. Table 2-3 describes the actions LTM for MVS performs depending on the types of datatype conversion errors and the values of specific LTM for MVS configuration parameters.

Table 2-3: Conversion errors and LTM for MVS parameters

Error converting:	LTM for MVS configuration parameter settings
date, time, or timestamp data	The system performs the following action, depending on the value specified for the LTM for MVS DateTime_conv_err configuration parameter: • DateTime_conv_err=Null Substitutes nulls in columns encountering errors. Make sure the affected columns allow nulls. • DateTime_conv_err=Default If LTM for MVS encounters an error in a date column, it supplies the value you specified for Date_conv_default. • DateTime_conv_err=Datetime If the LTM for MVS encounters an error in a date column, it supplies the value you specified for Date_conv_default and the value you specified for the Time_conv_default.

Task 1: Configuring Replication Server

Configure Replication Server using Replication Command Language (RCL) commands. You can skip this section if Replication Server was configured before or during installation.

This section includes the following tasks:

- Task 1a: Create the RS_user on Replication Server
- Task 1b: Grant connect source permission
- Task 1c: Create a connection to Replication Server
- Task 1d: Specify Replication Server character sets
- Task 1e: Create the replicate database
- Task 1f: Create replicate tables

- Task 1g: Create a connection to the replicate database

Note This section assumes that you have one Replication Server in your replication system. If you have more than one Replication Server in your replication system, you must create routes between them. See the Replication Server *Commands Reference* for information on the create route command.

See also

- Replication Server *Commands Reference* and *Administration Guide* for information on RCL commands
- Replication Server *Installation Guide* for detailed instructions on configuring Replication Servers

What you need to know

LTM for MVS RS_source_ds *configuration parameter*	The value of the LTM for MVS RS_source_ds configuration parameter identifies a unique name for the MVS system that contains the primary data source. Make sure that the MVS system name you supply in the create connection command matches the value you specified for the RS_source_ds parameter in the LTM for MVS configuration file.
LTM for MVS RS_source_db *configuration parameter*	The value of the LTM for MVS RS_source_db configuration parameter identifies a unique instance of the Log Transfer Manager (LTM) you are running. It is the name of the logical data source that contains the primary data to be replicated. Make sure that the database name you supply in the create connection command matches the value you specified for the RS_source_db parameter in the LTM for MVS configuration file.
Replication Server maintenance_user_ID	The maintenance user ID is the Replication Server user specified in the create connection command when creating a connection to Replication Server. If you plan to replicate into DB2, to prevent cyclic replication, the maintenance_user_ID should not be identical to the value of the RS_user parameter, which was recorded in section 7c of the Replication Agent Installation Worksheet during installation.

Replication Server maintenance_password	The value of the maintenance_password identifies the password for the Replication Server maintenance_user_ID. This password can be 30 characters or less.
dsi_suspended	This option to the create connection command starts the connection with the DSI thread suspended. The DSI thread is an outbound queue connection to the target or replicate database. Sybase strongly recommends that you use this option with Replication Agent.

Note Perform the tasks in the next section in the order shown.

Task 1a: Create the RS_user on Replication Server

Replication Server requires a valid Replication Server user ID to make the connection from Replication Agent for DB2 to Replication Server.

Issue the Replication Server create user command to create the Replication Server user. Use the values of the RS_user and RS_pw LTM for MVS configuration parameters as the user ID and password.

❖ **To create the Replication Server user**

1 Obtain the value of the RS_user configuration parameter. This information was recorded during installation on the Replication Agent Installation Worksheet.

2 Obtain the value of the RS_pw configuration parameter. This information was recorded during installation on the Replication Agent Installation Worksheet.

3 Log into the primary Replication Server as the System Administrator (*sa*) and enter the following command:

```
create user RS_user
set password RS_pw
```

where RS_user and RS_pw are the Replication Server user ID and password you located in steps 1 and 2.

See also

- Appendix A, “LTM for MVS Configuration Parameters”

- Replication Server *Commands Reference* for information about the create user command

Task 1b: Grant connect source permission

After you create the Replication Server user, you must grant connect source permission to this Replication Server user ID.

To grant connect source permission:

1. Locate the Replication Server user ID you just created, which should match the value of the RS_user configuration parameter in the LTM for MVS configuration file.
2. Log in to the primary Replication Server as the System Administrator (*sa*) and enter the following command:

   ```
   grant connect source to RS_user
   ```

   where *RS_user* is the Replication Server user ID you located in the previous step.

See also

- Appendix A, “LTM for MVS Configuration Parameters”
- Replication Server *Commands Reference* for information about the grant connect source command

Task 1c: Create a connection to Replication Server

A connection is a message stream from a Replication Agent to a Replication Server. Connections allow the primary data sources identified on this connection to participate in replication.

Create a connection from the primary data source to the primary Replication Server using the create connection command.

To create a connection to the primary Replication Server:

1. Log in to the primary Replication Server as the System Administrator (*sa*) using isql:

   ```
   isql -Usa -Psa_password -Sservername
   ```

2 Create the connection using the following syntax:

```
create connection to RS_source_ds.RS_source_db
set error class error_class
set function string class function_class
set username maintenance_user_ID
set password maintenance_password
with log transfer on, dsi_suspended
```

The following example creates a connection for the *DSNA* DB2 subsystem in the *MVSA* data server. These values should match the data server and database names you specify in the RS_source_db and RS_source_ds parameters in the LTM for MVS configuration file. Replication Server uses the default rs_sqlserver_error_class error class to handle errors for the database.

In addition, Replication Server uses the function strings in the default rs_sqlserver_function_class function string class for data manipulation operations.

Example

```
create connection to MVSA.DSNA
 set error class rs_sqlserver_error_class
 set function string class rs_sqlserver_function_class
 set username SSUSERID
 set password SSPASSWD
 with log transfer on, dsi_suspended
```

To prevent replication of updates applied as a result of replication into DB2, updates performed by the Replication Server maintenance user ID are not replicated. To replicate transactions performed by this user ID, you must do the following:

- Change the value of the LTM for MVS configuration parameter to Y.
- Specify the maintenance user ID and password in the Replication Server create connection command.

Warning! To prevent cyclic replication, the Replication Server maintenance user ID should *not* match the value of the RS_user parameter, which was recorded on the Replication Agent Installation Worksheet during installation.

See also

- Appendix A, “LTM for MVS Configuration Parameters,” for information about the LTM_process_maint_uid_trans configuration parameter and filtering Replication Server maintenance user ID transactions
- Replication Server *Commands Reference* for information on the create connection command

Task 1d: Specify Replication Server character sets

You must configure the primary Replication Server to use the code page value specified by the Replication Agent RS_ccsid parameter. This value can be any ASCII code page listed in the *CODEPAGE* member of the *hlq.DOCS* library.

Note The code page used by Replication Server is controlled by the value of the RS_charset Replication Server configuration parameter. The default value of the RS_charset parameter varies by the platform, operating system, and language of the Replication Server and is determined by Open Server/Open Client. For more details on this parameter, see the documentation for Replication Server.

See also

- “Managing Replication Server,” in the Replication Server *Administration Guide*, for information about the RS_charset Replication Server configuration parameter
- “International Replication Design Considerations,” in the Replication Server *Design Guide*, for details about working with character sets when designing international replication systems

Task 1e: Create the replicate database

If you have not already done so, create the replicate database to which Replication Agent replicates your source data.

See the Adaptive Server Enterprise *Reference Manual* for instructions on creating a replicate database using the create database command.

Task 1f: Create replicate tables

If you did not do so, create the replicate tables to which Replication Agent replicates your source data.

Note For each replicate table you specify in a replication definition, you must grant permissions using the following command:

```
GRANT ALL on Table_name to DATABASE_MAINT_USER
```

See also

- "Task 2: Creating replication definitions" on page 37
- Adaptive Server *Reference Manual* for information on the create table command

Task 1g: Create a connection to the replicate database

To define the replicate database to the primary Replication Server, you must create a connection from the primary Replication Server to the replicate database. Replication Server manages and distributes transactions to the replicate database.

You can create a connection to the replicate database in two ways, depending on the type of database it is:

- Automatically, for ASE databases, using the Replication Server rs_init utility, or
- Manually, for all other databases, using the Replication Server create connection command

  For information on creating a connection to the non-ASE replicate database, see the Replication Server *Heterogeneous Replication Guide*.

See also

- Appendix A, "LTM for MVS Configuration Parameters," for information on filtering maintenance user ID transactions using the LTM_process_maint_uid_trans LTM for MVS configuration parameter
- Replication Server *Installation Guide* for detailed instructions on using the Replication Server rs_init utility to create a connection to the replicate database
- Replication Server *Commands Reference* for information on the create connection command

- Replication Server *Design Guide* for requirements for using a non-Sybase database

Note Perform the tasks in the next section in the order shown.

Task 2: Creating replication definitions

Create replication definitions using the Replication Server create replication definition command to define the source data available for replicate tables that have subscriptions to the DB2 tables.

You must create at least one replication definition for each DB2 table that contains data you want to replicate.

When you create a replication definition for a table in the primary Replication Server, the replication system makes this table available to any subscribing replicate Replication Servers or replicate databases.

Each replication definition can have multiple subscriptions.

Make sure that you specify valid ASE datatypes when creating replication definitions. For information on valid datatypes, see the appropriate documentation for ASE.

Note Replication Server version 12.0 and later include Heterogeneous Datatype Support (HDS). This means that you can now use DB2 date, time, and timestamp datatypes without having to map them to ASE datatypes. For more information on setting up Replication Server and the HDS feature, see the Replication Server *Configuration Guide*.

See the first part of this chapter for detailed information about mapping DB2 datatypes to ASE datatypes.

❖ **To create a replication definition**

1 Log in to the primary Replication Server as the System Administrator (*sa*) using isql:

```
isql -Usa -Psa_password -Sservername
```

2 Create the replication definition using the following syntax:

```
create replication definition
replication_definition
 with primary at data_server.database
 [with all tables named [table_owner.]'table_name' |
 [with primary table named
   [table_owner.]'table_name']
 [with replicate table named
   [table_owner.]'table_name']]
 (column_name [as replicate_column_name]
     [datatype [null | not null]]
   [, column_name [as replicate_column_name]
     [datatype [null | not null]]]...)
 primary key (column_name [, column_name]...)
 [searchable columns (column_name
   [, column_name]...)]
 [send standby [{all | replication definition}
   columns]]
 [replicate {minimal | all} columns]
 [replicate_if_changed
   (column_name [, column_name]...)]
 [always_replicate (column_name [, column_name]...)]
```

Example

This example creates a replication definition named *titles_repdef* for the *TITLES* table. The primary copy of the *TITLES* table is located on the *MVSA* data server, in the *DSNA* DB2 subsystem. All replicate copies of the table are named *TITLES*. All columns in the *TITLES* table are replicated.

Note The case of the table and column names you specify in your replication definition must exactly match the case of the DB2 source table and column names.

```
create replication definition titles_repdef
 with primary at MVSA.DSNA
 with all tables named 'TITLES'
 (
 TITLE_ID char(6),
 TITLE varchar(80),
 TYPE char(12),
 PUBDATE datetime,
 PRICE float
 )
```

```
primary key (TITLE_ID)
```

What you need to know

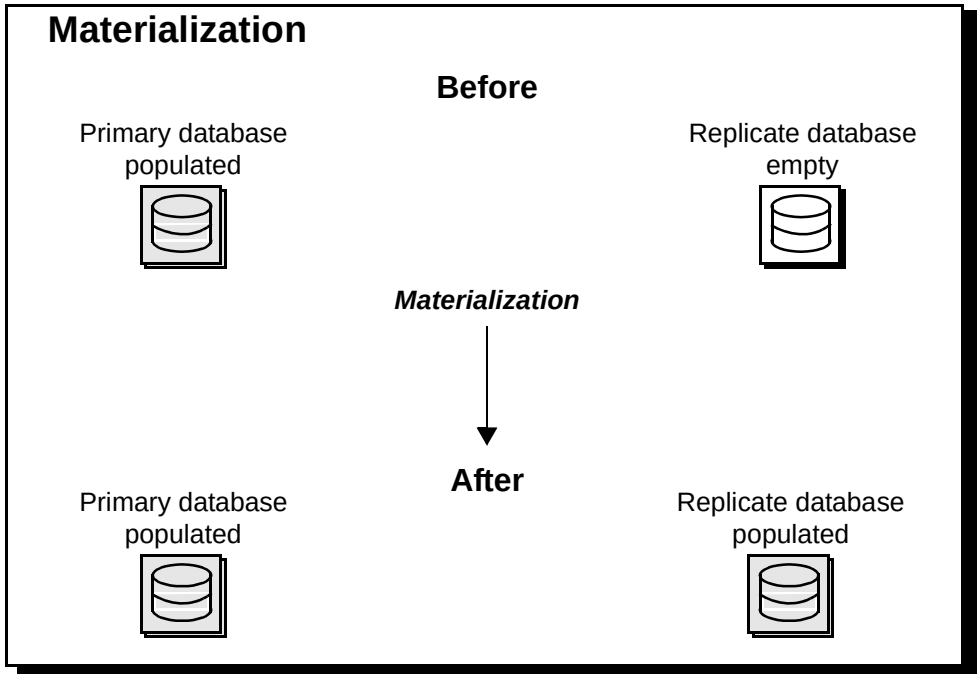
Table and column names	• If the suppress_col_names parameter is set to Y, column names you specify in replication definitions must appear in the same order that they appear in your DB2 tables. • The level of case-sensitivity for table and column names is determined by the LTL_table_col_case parameter.
Replication Server replicate minimal columns *option*	Replicates only the columns needed to perform update or delete operations at replicate databases. Use the replicate minimal columns option only with replication definitions that use the default function strings for the rs_update and rs_delete functions. If you specify replicate minimal columns for a replication definition, you cannot create a subscription using the nonatomic bulk materialization method because you cannot use the set autocorrection on option for the replication definition.

See the Replication Server *Commands Reference* for information about the Replication Server create replication definition command.

Task 3: Materializing replicate tables

Materialization is the process of initially populating the replicate database with a copy of the data from the primary database. LTM for MVS requires that you set up and populate each replicate database so that it is in a state consistent with that of the primary database. Figure 2-2 shows the effects of materialization on replicate databases.

Figure 2-2: Primary and replicate databases before and after materialization



There are two types of materialization:

- **Rep Agent materialization**

  Rep Agent materialization is the process of populating a replicate database using the unload utilities and the dataHandler=LTMMAT Replication Agent configuration. This method requires you to manually activate your subscriptions after the replicate database is populated.

- **Automatic materialization**

  Automatic materialization is the process of populating a replicate database using Replication Server commands. After the replicate database is populated with a copy of the primary data, replication is active.

Note Replication Agent for DB2 does not support automatic materialization.

See the Replication Server *Administration Guide* for further information on materialization.

Using Replication Agent materialization

When the IBM unload utility is used to unload data from a table space, a punch card file is produced that describes the data in a second file which contains all the data for that table space. The punch card file contains data in this format:

```
LOAD DATA INDDN SYSREC    LOG NO    REPLACE
 EBCDIC    CCSID(01027,05035,04396)
 INTO TABLE "username"."table_name     "
 WHEN(00001:00002 = X'00A1')
 ( "DATA1              " POSITION(    00004:00258)  CHAR MIXED(255)
                           NULLIF(00003)=X'FF'
 , "DATA2              " POSITION(    00260:00516)  VARCHAR MIXED
                           NULLIF(00259)=X'FF'
 , "DATA3              " POSITION(    00518:00772)  CHAR MIXED(255)
                           NULLIF(00517)=X'FF'
 )
```

The punch card file describes the rows and columns in the unloaded data file. The Replication Materialization process can use this information to reformat the unloaded data into log records and pass it to the Replication Extract as though it came from a DB2 log. In most cases, non-IBM utilities use this format. As long as the punch syntax and the unloaded data use the IBM format, then materialization will work with these utilities.

The only information about the table that cannot be found in the punch card file is the DBID (data base ID number) and the OBID (table ID number). These numbers are used by Replication Extract to find the LTMOBJECTS and system information for the table.

Materialization Server

A Replication Server is required to materialize the data. This server is referred to as the Materialization Server. This may be the main Replication Server in your system, but Sybase recommends that you use a separate server for materialization.

Before you begin

- Verify that the source table exists and contains data.
- Verify that the replicate table exists and contains the appropriate columns.
- Verify that you have succcessfully configured the Materialization Replication Server in your replication system. See “Task 1: Configuring Replication Server” on page 28 for details.

See also

- Replication Server *Commands Reference* for information on RCL commands
- Replication Server *Administration Guide* for more information about materialization methods

Replication Agent materialization configuration

To set up and configure the Replication Agent for materialization, follow these steps:

1 Add the source table to the LTMOBJECTS table for the Replication Agent.

2 Create a configuration file and configure it to communicate with the Materialization Server.

3 Change the dataHandler configuration to dataHandler=LTMMAT.

4 Unload the source table to get a punch card file and a data file.

5 Obtain the SYSIBM.SYSTABLES DBID and OBID information for the source table.

6 Create JCL to run the Materialization job and change it to use the configuration file from step 1.

7 Add the Maintenance JCL DD statements and data to the JCL in step 5.

Sample JCL DD statements and data

```
//SYSPUNCH DD DISP=SHR,DSN=Unload.Punch.Card.Dataset
//SYSREC   DD DISP=SHR,DSN=Unload.Table.Data.Dataset
//MATCMD   DD *
--MAT 257,125,2000
```

The MATCMD configuration statement syntax is:

```
--MAT dbid,obid,commit[,skip]
```

Commands start with two dashes. At this time, there is only one MAT command. The arguments in the materialization command are:

- dbid is the source table DBID from step 4, above.
- obid is the source table OBID from step 4, above.
- commit is the number of lines sent before a commit is sent.

- skip is a restart feature that gives the number of records to skip before replicating data.

Restarting materialization

DB2 uses a relative byte address (RBA) to identify log records. The RBA value is the numbeer of bytes from the beginning of the file plus the HIGH RBA value of the previous archived log file. This value is also referred to as a log record sequence number (LRSN).

The Materialization RBA begins with 1 and increments by 1, so that the materialization process can be continued if a problem occurs. When a problem does occur, the materialization command can be modified by adding the skip parameter. Records are ignored until the skip record value is reached, then replication begins again at the next record.

The truncation point or QID from the materialization server has the value of the oldest open transaction (OOT). There are three RBA values in the truncation point. The middle RBA is the record number of the OOT. skip should be set to the OOT value minus 1, unless you want to skip a bad record or records.

After materialization

The materialization process ends when the end of the unloaded data file is encountered. At this point, Replication Agent displays the following message:

```
LEXTRACE Read past end of log, check the truncpt.
```

When the materialization process is complete, you can add the source table to the LTMOBJECTS table for the Replication Agent that will be replicating changes.

See also Atomic and NonAtomic materialization in “Using bulk materialization.”

Using bulk materialization

When you use bulk materialization, you must:

- Unload the source data at the primary site
- Move unloaded data to the replicate servers
- Load data into the replicate tables

- Apply updates made at the primary site after the media unload is performed

Note Replication Agent for DB2 does not support use of the DB2 LOAD LOG YES utility to assist in replicate materialization.

Three bulk materialization methods are available. The method you use depends mainly on whether one of the following conditions exist at your site:

- A copy of the data already resides at the replicate site
- Applications using the primary data can tolerate interruptions

The following table summarizes bulk materialization methods:

Table 2-4: Summary of bulk materialization methods

Bulk materialization method	Summary of materialization process
Subscription data already exists in the replicate database	If the subscription data already exists at the replicate site, you must verify that it is consistent with the primary data and then define, activate, and validate the subscription.
Atomic materialization	This method prevents client applications from executing transactions against the primary data while the subscription data is unloaded. After defining the subscription, lock the primary data, retrieve the subscription data, activate the subscription, and unlock the primary data. Use this method to retrieve data from the primary database, if you are able to suspend updates to the primary data.
Nonatomic materialization	This method *allows* client applications to execute transactions against the primary data while the subscription data is unloaded. If you use this method, the data at the replicate database may be inconsistent with the data at the source database. Use this method only if you are unable to suspend transactions on the source database.

Performing atomic bulk materialization

The atomic bulk materialization method locks the primary database tables while materialization occurs. The data is copied to the replicate database without any temporary inconsistency.

This method ensures consistency between source and replicate data by retrieving the subscription data, activating the subscription, and suspending the data server interface (DSI) connection to the replicate database all in one transaction at the primary data server.

Before you begin

- Verify that the source database exists and contains data.
- Verify that the replicate table exists and contains the appropriate columns.
- Verify that you successfully configured every Replication Server in your replication system. See "Task 1: Configuring Replication Server" on page 30 for details.

See also

- Replication Server *Commands Reference* for information on RCL commands
- Replication Server *Administration Guide* for more information about materialization methods

What you need to know

with suspension *clause*

Use the with suspension clause with the activate subscription command to prohibit updates to the source data until after the subscription initializes.

When you use the with suspension clause, the system suspends the DSI thread after changing the subscription status to active. This prevents the replicate Replication Server from sending updates for the replicated table before the subscription data loads.

After the data loads, you can execute the resume connection command to apply the updates.

Note If you are activating multiple subscriptions (consecutively in a script, for example), to avoid issuing the resume connection command repeatedly, use the with suspension clause only with the first or last activate subscription command. Execute the resume connection command once after the multiple activate subscription commands.

❖ **To bulk materialize using the atomic method:**

1 Log in to the replicate Replication Server as the System Administrator (*sa*) using isql:

isql -U*sa* -P*sa_password* -S*servername*

2 Define the subscription at the replicate Replication Server using the following syntax:

```
define subscription subscription_name
for replication_definition
with replicate at dataserver.database
[where search_conditions]
```

3 Check the subscription at both the primary and replicate Replication Servers to verify that the subscription status is defined. Use the following syntax:

```
check subscription subscription_name
for replication_definition
with replicate at dataserver.database
```

4 Prevent updates to DB2 by using the following command to restrict it to read-only access during materialization:

```
START DATABASE (database_name) SPACENAM (space_name) ACCESS (RO)
```

where *database_name* is the DB2 database that contains the DB2 primary table, and *space_name* is the DB2 tablespace that contains the DB2 primary table.

5 Unload the source data at the primary site by retrieving the data from the primary database using your site's preferred database unload method to select or dump the subscription data.

6 Activate the subscription with suspension at the replicate Rep Server using the folowing syntax:

```
activate subscription subscription
for replication_definition
with replicate at dataserver.database
with suspension
```

7 Use the Activate and Validate Subscription panel to activate the subscription, using the with suspension clause.

The Activate and Validate Subscription panel activates and validates subscriptions by sending Replication Server commands to the DB2 log. Replication Agent for DB2 sends the log to Replication Server, which activates and validates subscriptions.

- From an ISPF command line, type:

  `TSO EXEC '`*`hlq`*`.ISPF(LTMCSET)'`

  (where *hlq* is the high-level qualifier and version number recorded in sections 3a of the Replication Agent Installation Worksheet).

8 Unlock the primary tables to restore read-write access using the following example syntax:

```
START DATABASE (database_name) SPACENAM (space_name) ACCESS (RW)
```

where *database_name* is the DB2 database that contains the DB2 primary table, and *space_name* is the DB2 tablespace that contains the DB2 primary table.

9 Wait for the subscription to become active at both the primary and replicate Replication Servers; then execute the check subscription command at both the primary and replicate Replication Servers to verify that the subscription status is active.

When the subscription status is active at the replicate Replication Server, the database connection for the replicate database is suspended.

10 Load the subscription data into the replicate database using the BCP utility or other load utility.

11 From the replicate Replication Server, resume the database connection for the replicate database using the following syntax:

```
resume connection to dataserver.database
```

12 Validate the subscription at the replicate Replication Server using the following syntax:

```
validate subscription subscription
for replication_definition
with replicate at dataserver.database
```

13 Wait for the subscription to become valid at both the primary and replicate Replication Servers; then, execute the Replication Server check subscription command (as in step 3 of this task) at both the primary and replicate Replication Server to verify that the status is valid.

Now the subscription is created and replication is active.

Performing nonatomic bulk materialization

This section describes how to bulk materialize data using the nonatomic materialization method.

The nonatomic bulk materialization method allows a period of flux at the replicate site during which the replicate data may be inconsistent with the primary data. By the time you validate the subscription data, however, the data should be consistent.

You must use the Replication Server set autocorrection on command during nonatomic bulk materialization so that inconsistencies resulting from continuing updates in the primary database can be resolved without errors.

Autocorrection determines how Replication Server processes inserts and updates to copies of replicated tables. When autocorrection is on, Replication Server converts each update or insert operation into a delete followed by an insert.

Before you begin

- Verify the following:
  - The data source exists and contains data.
  - The replicate table exists and contains the appropriate columns.
  - You successfully configured every Replication Server in your replication system. See “Task 1: Configuring Replication Server” on page 30 for details.

See also

- Replication Server *Reference Manual* for information on the set autocorrection command
- Replication Server *Administration Guide* for more information on materialization methods

What you need to know

Using the Replication Server set autocorrection *command*	This command prevents failures caused by missing or duplicate rows in a copy of a replicated table. Autocorrection is off by default for replication definitions you create. You should enable autocorrection only for replication definitions with subscriptions that use nonatomic bulk materialization. After materialization is complete and the subscription is VALID, issue the set autocorrection off command to improve system performance.
Using the Replication Server replicate minimal columns *command*	Do *not* use the nonatomic bulk materialization method if you are using the replicate minimal columns option in your replication definition. Nonatomic materialization *cannot* be used with data for which replication definitions have the replicate minimal columns option, because autocorrection cannot resolve the inconsistencies between source and replicate data. When you use the replicate minimal columns option for a replication definition, data is replicated from the minimum number of columns needed for delete or update operations. For rows to be deleted from replicate tables, Replication Server only replicates data from primary key columns. For rows to be updated in replicate tables, Replication Server only replicates data from primary key columns and columns that changed.
Using the Replication Server activate subscription *command with the* with suspension *clause*	You must use the with suspension clause with the activate subscription command to prohibit updates to the primary version until after the subscription initializes. When you use the with suspension clause, the system suspends the DSI thread after changing the subscription status to ACTIVE. This prevents the replicate Replication Server from sending updates for the replicated table before the subscription data loads. After the data loads at the replicate site, you can execute the resume connection command to apply the updates.

To bulk materialize using the nonatomic method:

1 Log in to the replicate Replication Server as the System Administrator (*sa*) using isql:

```
isql -Usa -Psa_password -Sservername
```

2 Issue the Replication Server define subscription command at the replicate Replication Server using the following syntax:

```
define subscription subscription_name
for replication_definition
with replicate at dataserver.database
[where search_conditions]
```

3 Check the subscription at both the primary and replicate Replication Servers to verify that the subscription status is DEFINED. Use the following syntax:

```
check subscription subscription_name
for replication_definition
with replicate at dataserver.database
```

4 Activate the subscription, using the with suspension clause, at the replicate Replication Server. Use the following syntax:

```
activate subscription subscription_name
for replication_definition
with replicate at dataserver.database
with suspension
```

5 Wait for the subscription to become active at the primary Replication Server and any replicate Replication Servers; then, execute the Replication Server check subscription command at the primary and replicate Replication Servers to verify that the subscription status is ACTIVE.

When the subscription status is ACTIVE at the replicate Replication Server, the database connection for the replicate database is suspended.

6 As soon as the subscription becomes active at the primary Replication Server, retrieve the data from the primary database using your site's preferred database unload method.

7 Validate the subscription at the replicate Replication Server using the following syntax:

```
validate subscription subscription_name
for replication_definition
with replicate at dataserver.database
```

8 Load the subscription data from the media into the replicate database.

9 From the replicate Replication Server, enable autocorrection for the replication definition at the replicate database using the following syntax:

```
set autocorrection on
 for replication_definition
with replicate at dataserver.database
```

10 From the replicate Replication Server, issue the Replication Server resume connection command to restore the connection to the replicate database using the following syntax:

```
resume connection
 to dataserver.database
```

11 Wait for the subscription to become valid at both the primary and replicate Replication Servers; then issue the Replication Server check subscription command at the primary and replicate Replication Servers to verify that the subscription status is valid.

When the subscription status is valid, the replicate data is consistent with the primary data.

12 From the replicate Replication Server, turn autocorrection off for the replicate database using the following syntax:

```
set autocorrection off
 for replication_definition
with replicate at dataserver.database
```

Now the subscription is created and replication is active.

CHAPTER 3

Replication Agent Setup

This chapter provides information about DB2 considerations that affect working with source tables, and instructions for marking tables for replication.

DB2 source table considerations

This section addresses DB2-related items that you must consider when using Replication Agent to replicate DB2 tables, including the following topics:

- DB2 table size
- DB2 table names and reserved keywords
- Restrictions on DB2 utilities

DB2 table size

Use the following calculation to identify the size of primary table update:

```
                          Message overhead +
                              2 * Row size +
2 * Sum of length of column names +
                      8 * Number of columns
_____________________________________________
= Inbound message size of the update
```

- *Message overhead* is 200 bytes. Each message includes a transaction ID, an origin queue ID, and other information.
- *Row size* is the ASCII representation of the table columns (float or decimal columns are also presented in ASCII). For example, a CHAR(10) column has a size of 10 bytes. Multiply row width by 2 because both the before and after images of the row are distributed to replication sites.
- *Sum of length of column names* is the sum of all the column name lengths in the table. Multiply this sum by 2.

  Note When you set the value of the Suppress_col_names parameter to Y and the value of the Use_repdef parameter to Y, you can eliminate the sum of the length of the column names and reduce the LTL by this amount.

- *Number of columns* is the number of columns in the updated table. Multiply this number by 8.

Add the four numbers to obtain an estimate, in bytes, of the message size of the largest possible update.

For example, if you had a table with three columns of CHAR(10) each, and column names of 8 bytes each, the calculation is:

```
200 bytes +
2 * 30 bytes +
2 * 24 bytes +
8 * 3

_______________
 = 332 bytes
```

What you need to know

Spaces	The calculation for table sizes in this section does not include spaces between command components.
Minimal_cols *configuration parameter*	If the value of the LTM for MVS Minimal_cols configuration parameter is set to Y, Replication Agent only replicates the after images of columns that changed and that contain key data for a changed row. This can reduce the volume of the Log Transfer Language (LTL) for some transactions, potentially increasing throughput.

Suppress_col_names *and* Use_repdef *configuration parameters*	Setting the values of the LTM for MVS configuration parameters Use_repdef to Y and Suppress_col_names to Y can eliminate the sum of the length of the column names calculated and reduce the LTL by that amount.

See Appendix A, "LTM for MVS Configuration Parameters," for more information on the Minimal_cols, Suppress_col_names, and Use_repdef configuration parameters.

DB2 table names and reserved keywords

Do not use table or column names in the source tables that are reserved words in the replicate database. You can create replication definitions and subscriptions, but you cannot create tables in the replicate database if columns or tables use reserved keywords.

What you need to know

Table names	If you must use table names in data that are reserved at the replicate site, you can write a user exit, create function strings, or use the REPLICATE_NAME field in the LTMOBJECTS table to change these names before Replication Server sends the transactions to the replicate database.
Column names	If you must use column names in the source tables that are reserved keywords at the replicate site, you can write a user exit or create function strings to change these names before Replication Server sends the transactions to the replicate database.

See the Sybase Adaptive Server *Reference Manual* for listings of reserved keywords in ASE.

Restrictions on DB2 utilities

Using some DB2 utilities on primary tables can jeopardize replicate databases, which may necessitate rematerializing the data.

Do *not* run the following DB2 utilities on the DB2 tables marked for replication:

- RECOVER to a prior point in time, using TOCOPY or TORBA. This requires rematerialization.
- REORG or REORG UNLOAD PAUSE (unless you delete rows after the UNLOAD phase)
- Any DB2 utility that makes changes to the primary tables or tablespaces during materialization.
- REPAIR

Replication Extract operating considerations

Read the operating considerations in this section carefully and make any necessary modifications to the environment before you use the Replication Extract component of Replication Agent.

You may need to change these areas of the system:

- Active and archive log and BSDS access
- Replicate table definitions
- DB2 utilities
- DB2 compression
- DATA CAPTURE CHANGES clause
- Editprocs and fieldprocs used
- LTMOBJECTS table updates
- Error handling

Active and archive log access

Replication Extract can access the first or second copy of the active and archive DB2 logs. If your active log is small and is archived several times a day, consider archiving the DB2 system to a data access storage device (DASD) instead of to tape. Archiving the log to DASD helps ensure that a DASD copy of any log data needed by Replication Extract is always available. This eliminates the need to mount tapes or to wait for a unit or a volume to become available.

You can then use DFHSM or a similar product to migrate the archive data sets to tape on a regular basis to reduce the amount of DASD devoted to your active and archive log data sets.

Warning! Never create archive logs from different DB2 subsystems or for more than one LTM for MVS on a single tape. Archive logs cannot be merged and read in parallel if different subsystems are using the same tape. This is especially important in data-sharing environments.

Replicate table definitions

Replication Extract does not support historical definitions of DB2 tables. It uses the current definition of a table as recorded in the DB2 catalog when extracted log data is formatted. Therefore, use caution when you alter the definition of a DB2 table to be replicated.

If the alteration requires dropping and re-creating the table (for example, if you are moving a table from a simple tablespace to a partitioned tablespace), you must ensure that all data changes are replicated and passed to the Replication Server before dropping and re-creating the table. This ensures that Replication Extract does not need to reread the portion of the log that contains data changes made to the original version of the table.

Before you drop the table, turn off replication for the primary table. This prevents Replication Extract from extracting any more changes made to a table having this DBID.OBID combination. (*DBID* and *OBID* are the database ID and the object ID, respectively, that uniquely define the database to DB2.)

After you re-create the table, start replication by using the Replication Agent interface dialog panels to enable replication for the primary table.

See “Working with Replication Agent system tables” on page 59 for more information.

DB2 compression

Replication Extract supports the use of DB2 compression. If any of the primary tables reside in a DB2 compressed tablespace, the DB2 log can contain compressed data. To expand this data, the DB2 IFI API requires access to the DB2 compression dictionary.

It is very important for Replication Extract that the DB2 IFI API is able to access the appropriate DB2-built compression dictionary. DB2 stores this dictionary in the compressed tablespace. Whenever possible, the API return expanded data. If the dictionary cannot be retrieved from the compressed tablespace, Replication Extract attempts to retrieve the dictionary from an appropriate full image copy.

The DB2 LOAD, REORG, RECOVER TOCOPY, RECOVER TORBA utilities can change the dictionary stored in the compressed tablespace. When you use any of these utilities, make a full image copy before the tablespace is made available for updates. This ensures Replication Extract can always access the appropriate compression dictionary.

These utilities may also stop, start, lock, or unlock a tablespace. Occasionally, a log record is written before a matching image copy is registered with DB2. When this occurs, Replication Extract waits until the tablespace or image copy is available before processing continues.

Large DB2 tablespaces

You can create large tablespaces with DB2 version 6 and later using the DB2 DSSIZE parameter. Replication Agent can replicate these tables, even when they are compressed, in DB2 version 7 and later.

DATA CAPTURE CHANGES clause

The DATA CAPTURE CHANGES table attribute ensures that a full image of all changed rows is written to the DB2 log. Use the ALTER TABLE statement to activate DATA CAPTURE CHANGES on existing tables.

See also

- "Working with Replication Agent system tables" on page 59 for more information about using the Replication Agent interface panels
- *IBM DB2 SQL Reference* for information on using the ALTER TABLE statement

DB2 editprocs and fieldprocs

Replication Extract does not support the use of DB2 edit procedures (editprocs) and field procedures (fieldprocs).

Error handling

If Replication Extract encounters an error while trying to process a log record that describes a change to a primary table, replication ceases on that table. By default, Replication Extract terminates processing if this occurs. You can, however, request that Replication Extract continue replicating changes made to other primary tables. This behavior is controlled by the value of the LogExtractError parameter in the configuration file.

Note If Replication Extract encounters errors while attempting to process log records that describe changes to primary tables, Sybase recommends that you use the LogExtractError parameter value to prevent replication from continuing on other tables.

Working with Replication Agent system tables

During installation, Replication Agent creates several system tables on DB2. This chapter contains detailed information about using the Replication Agent system tables on DB2 to coordinate bulk materialization and mark tables for replication. Use this chapter for the following tasks:

The SQLINIT member, located in the *hlq.JCL* data set, contains a set of example SQL statements that you can use to create the LTMOBJECTS table in a DB2 system.

The LTMADMIN user (this identifier was recorded during installation, on section 1 of the Replication Agent Installation Worksheet) should insert and delete rows into the LTMOBJECTS table representing tables marked for replication.

LTMOBJECTS tracks the DB2 tables marked for replication.

Understanding and using translation tables

Replication Agent translates character data from a source Coded Character Set Identifier (CCSID) to a target CCSID using IBM translation tables. These tables are used from within the LTM and can be viewed by querying DB2. The following query displays all CCSID conversions supported by DB2 and the translation tables used in these conversions:

```
SELECT * FROM SYSIBM.SYSSTRINGS
```

If no CCSID value is specified for the target CCSID, Replication Agent uses the default value shown in this translation table.

Verifying CCSID settings

You can verify the CCSID setting for a database by executing the following query:

```
SELECT * FROM SYSIBM.SYSDATABASE WHERE NAME = 'database_name'
```

where *database_name* is the name of the database for which you are verifying the CCSID setting.

You can also verify the CCSID settings for translation tables used by the LTM in character data conversions by executing the following query:

```
SELECT * FROM SYSIBM.SYSSTRINGS WHERE INCCSID = inccsid AND
OUTCCSID = outccsid
```

Note Replication Agent versions 12.6 and later support all DB2 CCSIDs, including ASCII and double-byte CCSIDs.

Migration considerations

In translating character data to the target CCSID, there are two major differences between Replication Agent version 12.0 and earlier versions:

1 The target CCSID for Replication Agent versions earlier than 12.0 is CP819. Replication Agent version 12.0 supports all Replication Server character sets as target character sets, including double-byte character sets.

2 For the source CCSID value, Replication Agent versions earlier than 12.0 use the value of the Codepage configuration parameter. Replication Agent version 12.0 and later uses the CCSID value of the DB2 database in which a table was created as the source CCSID, unless this value is zero. (To obtain the CCSID value of the DB2 database in which a table was created, see your DB2 database administrator). If the source CCSID value is zero, Replication Agent version 12.0 and later obtains the source CCSID from the Replication Agent Codepage configuration parameter.

   If the value of the source CCSID is a non-zero value for Replication Agent version 12.0 and you want to override this value, you must reset the CCSID value in the Codepage configuration parameter value using the force option, as shown in the following example:

   ```
   Codepage=CP037,force
   ```

   Warning! If you had an earlier version of Replication Agent with the Codepage configuration parameter set to a value different from the DB2 database value, you may see differences in character translations with the new version.

Table 3-1 illustrates how Replication Agent obtains the source CCSID with respect to the DB2 CCSID value for the DB2 database in which the primary table was created (referred to here as the DB2 CCSID value) and the value of the Replication Agent Codepage parameter. The value of the Codepage parameter is represented as *cpNNN*.

Table 3-1: CCSID values used by Replication Agent

Codepage parameter	DB2 version 5, version 6, and version 7
`Codepage=`*`cpNNN`*	• If DB2 CCSID value is zero, Replication Agent uses *cpNNN* as source CCSID value. • If DB2 CCSID value is greater than zero, Replication Agent uses DB2 CCSID value for source CCSID value.
`Codepage=`*`cpNNN`*`,force, utf8`	• If DB2 CCSID value is zero, Replication Agent uses *cpNNN* as source CCSID value. • If DB2 CCSID value is greater than zero, Replication Agent uses *cpNNN* as source CCSID value. • If the replication definition for the column in the primary database specifies a Unicode datatype (unichar or univarchar) then the utf8 option for the Codepage configuration parameter enables replication from DB2 using a Unicode character set.
Codepage parameter commented out of the configuration file or not present in the configuration file	• If DB2 CCSID value is zero, Replication Agent uses default source CCSID value of cp500. • If DB2 CCSID value is greater than zero, Replication Agent uses DB2 CCSID value for source CCSID value .

With DB2 versions 5, 6, and 7, the value Replication Agent uses for the source CCSID is obtained from the CCSID of the DB2 database for the table being replicated. In most cases, this CCSID value will be greater than zero.

See Appendix A, “LTM for MVS Configuration Parameters” for more information on the Codepage configuration parameter.

Understanding and using the LTMOBJECTS table

The LTMOBJECTS table provides Replication Extract with information about the DB2 primary tables marked for replication.

When Replication Agent starts, Replication Extract uses the information in the LTMOBJECTS table to obtain table metadata from the DB2 catalog and stores an internal list of tables to be replicated. When the contents of the LTMOBJECTS table change, Replication Extract updates this internal list.

LTMOBJECTS table uses

Use the LTMOBJECTS table to accomplish the following:

- To prevent replication of tables containing errors. To stop replication of DB2 tables containing errors, set the value of the ERROR_CODE column in the LTMOBJECTS table to 20.
- To make a replicate table name different from the corresponding DB2 table name. To do this, edit the REPLICATE_NAME column.
- To enable replication to a fully qualified table name. To do this, edit the QUALIFY column.

LTMOBJECTS table contents

This Replication Agent system table is stored on DB2. It tracks the DB2 primary tables marked for replication. The LTMOBJECTS table contains one row for each table that is marked for replication. Replication Extract reads this table at start-up to determine which tables to replicate. When this table changes, Replication Extract dynamically implements the changes.

Table 3-2 describes the contents of the LTMOBJECTS table.

Table 3-2: LTMOBJECTS table schema

Column name	Datatype	Length (bytes)	Description
CREATOR	char not null with default	8	Creator of the table to be replicated.
NAME	char not null	30	Name of the table to be replicated.
QUALIFY	char not null with default, which is N	1	This column is used to indicate whether the name of the table being replicated is qualified with its creator ID.
QID	char not null with default	32	The queue ID at the time of failure. Disregard unless error code is non-zero. Replication Extract makes changes to this column.
ERROR_CODE	smallint not null with default	2	The Replication Extract error code. Non-zero means the log extract encountered a problem replicating this table and Replication Agent stopped replication. Replication Extract makes changes to this column.

Column name	Datatype	Length (bytes)	Description
REPLICATE_NAME	char not null with default	128	Blank spaces in this column indicate primary and replicate table names are the same. Edit the entry in this field to assign a different name to the replicate table name. See Chapter 4, "Managing Replication Agent" for more information. **Note** Replication Agent does not verify that the entry is a valid table name.

LTMOBJECTS table creator permissions

The MVS user ID that submits the SQLINIT job creates the LTMOBJECTS table. Sybase recommends that you create an *LTMADMIN* user ID during Replication Agent installation and using the ID to install Replication Agent.

See also

- Chapter 4, "Managing Replication Agent" for information about changing replicate table names
- Replication Agent for DB2 UDB *Installation Guide* for OS/390 for information about granting permissions to the *LTMADMIN* user ID and binding the log extract plan

CHAPTER 4

Managing Replication Agent

This chapter explains how to manage Replication Agent.

Startup and shutdown

This section explains how to start LTM for MVS as a started task or as a batch job. It also explains how to stop LTM for MVS.

Starting LTM for MVS

You can run the LTM for MVS start-up procedure (LTM*xxxxx*) on your system in one of two modes:

- Started task

  LTM for MVS replicates continuously. As changes are recorded in the DB2 log, LTM for MVS processes the changes and distributes them to Replication Server.

- Batch processing

LTM for MVS pulls all currently logged changes from the DB2 log at one time. For example, the LTM for MVS start-up procedure can be included in your daily backup procedure so that changes are applied to replicate databases only during off-peak hours.

Note To prevent Replication Agent from falling behind in reading the primary database transaction log, make sure the LTM*xxxxx* procedure starts as soon as possible after the source data is marked for replication.

What you need to know

Running multiple Replication Agents	Each instance of LTM for MVS can pull data from only one Replication Extract. Make sure each instance of LTM for MVS contains the following unique set of files: • LTM for MVS configuration file • Truncation point file • *LTLOUT* file • *LTMLOG* file
Restarting Replication Agent	Sybase recommends that before restarting Replication Agent, you set the value of the API_QID_request_interval configuration parameter to 30 or another low value. For production, set the value of API_QID_request_interval to a higher value, between 2000 and 10000.

See also

- "Running Multiple Replication Agents" in the Replication Agent for DB2 UDB *Installation Guide* for OS/390 for information on how to set up your system to run with multiple Replication Agents

Note To prevent Replication Agent from falling too far behind in reading the primary database transaction log, make sure that the LTM for MVS task starts as soon as possible after you start DB2, and that your DB2 tables are available for update.

To start LTM for MVS as a started task

1 Start Replication Server according to the instructions in the Replication Server *Installation Guide*.

2 Issue the START (S) command on the MVS operator console:

```
S LTMxxxxx
```

where LTM*xxxxx* is the LTM start-up procedure name you defined at installation.

3 Check the LTMLOG log data set to verify that the system wrote the configuration file parameter values to the LTMLOG log data set. (You can find the log name by checking the LTMLOG DD name in the LTM*xxxxx* procedure.) This indicates that you successfully started LTM for MVS.

To start LTM for MVS as a batch job

1 Start Replication Server.

2 Submit a job that calls the LTM*xxxxx* procedure.

3 Check the LTMLOG log data set to verify that the system wrote the configuration file parameter values to the LTMLOG log data set. (You can find the log name by checking the LTMLOG DD name in the LTM*xxxxx* procedure.) This indicates that you successfully started LTM for MVS.

See also

- "Installing Replication Agent" in the Replication Agent for DB2 UDB *Installation Guide* for OS/390 for information about defining the LTM for MVS start-up procedure (LTM*xxxxx*)

Stopping LTM for MVS

Shut down LTM for MVS using the MVS STOP (P) command.

To stop LTM for MVS

Enter the following command on the MVS operator console using the following syntax:

```
P jobname
```

where *jobname* is the LTM for MVS start-up PROC name (LTM*xxxxx*) or batch procedure name, depending on whether LTM for MVS was initiated as a started task or batch job.

See also

- "Installing Replication Agent" in the Replication Agent for DB2 UDB *Installation Guide* for OS/390 for information about the LTM for MVS start-up procedure (LTM*xxxxx*)

Replication Agent start-up problems

This section suggests ways to troubleshoot problems at startup.

Checking the logs

Check these items for the following types of information:

Replication Agent for DB2 log	Check error message numbers against those listed in Appendix C, “LTM for MVS Messages,”and check the symptoms against the information contained in the remainder of this chapter.
MVS operator console	Check error message numbers from the MVS operator console against those listed in Appendix C, “LTM for MVS Messages,”and Appendix D, “MVS Console Messages,”, and check the symptoms against the remainder of this chapter.
Replication Server logs	Check the Replication Server logs for information about the nature of the problem. Check error messages from Replication Server against those listed in the Replication Server *Troubleshooting Guide*.

TCP/IP connectivity failure

The following subsections describe common symptoms and error messages that provide clues to the resolution of a TCP/IP connectivity problem.

Messages with EZY prefix and IBM TCP/IP

Messages with the EZY prefix can indicate problems with IBM TCP/IP.

The following error message (in the LTM for MVS log file or, if you run Replication Agent with the PING JCL parm, in the *SYSOUT* file) can indicate TCP/IP connectivity failure:

```
EZY3744E RC=xxxx on IUCV_CONNECT TO TCPIP
```

❖ **To resolve the TCP/IP connectivity failure**

1 Check the values of the Replication Server parameters in the configuration file.

2 Check the values of the Replication Definitions Connection in the configuration file.

See also

- "Installing Replication Agent" in the Replication Agent for DB2 UDB *Installation Guide* for OS/390.
- IBM TCP/IP documentation

If you receive abends when using TCP/IP

You can receive a SOC1 or SOC4 abend if you specify IBMTCP as the value of the Communications_Protocol configuration parameter and you do not have IBM TCP/IP running on MVS, or IBM TCP/IP terminates while Replication Agent is running.

❖ **To correct the abend condition when using TCP/IP**

1 Determine which TCP/IP software is installed on MVS.

2 Edit the LTM for MVS configuration file to provide the correct value for the Communications_Protocol parameter.

3 Restart Replication Agent after ensuring TCP/IP is running.

See Appendix A, "LTM for MVS Configuration Parameters," for more information on the Communications_Protocol parameter.

Troubleshooting TCP/IP failure

If running Replication Agent with the PING JCL parameter indicates a failure to connect to Replication Server, and you are using IBM TCP/IP for MVS, follow this procedure to determine the source of the failure.

❖ **To determine the source of the TCP/IP failure**

1 Run the TSO PING command. If the results indicate success, continue to the next step.

2 Test connectivity to your Replication Server using ISQL or SYBPING from a machine that does not contain DB2 or your Replication Server.

If the connection fails, the problem is on the machine containing Replication Server. Follow troubleshooting procedures for Replication Server (and Open ClientConnect, if necessary). After you correct those problems, repeat Step 1.

If the connection to Replication Server succeeds, continue to Step 3.

3 Run the PINGRS program for TCP/IP. The JCL for this job is in the *PINGRS* member, in the *hlq.JCL* data set.

- Make sure that the *STEPLIB* or *JOBLIB* DD statements point to the SAS/C runtime library.
- Include the *SYSPRINT* DD statement in the JCL.

4 Run the Replication Agent with the PING JCL parameter. If it indicates failure, check the following output:

- All Open ClientConnect messages. Record message numbers and save the output.
- *SYSPRINT* DD contents for any messages from TCP/IP. Check the message guides for the appropriate product for further explanation. Save the output.
- The Replication Server error log on the Replication Server machine. Error or information messages in this log for Replication Agent indicate that the TCP/IP connection succeeded and that there may be a problem with the Replication Server configuration.

5 Check the JES2 or JES3 system logs for PING messages. Messages generated by RACF, ACF2, and TCP/IP are most common; however, investigate any message that is not security-related.

Remember that MVS tasking can cause messages to appear 10 to 15 seconds before or after the “STARTED” and “ENDED” messages are issued for the job. Save the output. If steps 1 through 4 do not indicate the component(s) that failed, save all output and continue the TCP/IP diagnosis with assistance from your MVS TCP/IP systems programmer.

6 Verify that the value of the port configurations matches the value of *port_number* in the interfaces file on the server that contains the primary Replication Server. The port values must be identical in both files.

If either of these values are incorrect, change them and return to Step 1 of “To determine the source of the TCP/IP failure” on page 71. If they are correct, proceed to the next step.

7 Continue testing with the TSO command NETSTAT ALL. The output from the NETSTAT ALL command shows whether the socket is in use by other applications. Use the output from this command to determine whether the port specified in the Replication Agent configuration file and in the interfaces file is in use by another task on z/OS.

Note According to IBM technical support, when an IUCV socket starts on the source system, it uses the socket or port number from the replicate system.

8 The IP address is resolved in the HOST.ADDRINFO and HOST.SITEINFO data sets. To test these data sets and verify the IP address, issue the TESTSITE command. Save the output from the command.

If this command output fails to show the correct address, the problem may be an incorrect entry in these data sets (the input to these data sets is the HOST.LOCALS data set). Correct this entry, and repeat this procedure starting with Step 1 of "To determine the source of the TCP/IP failure" on page 71.

If the command shows the correct address, proceed to the next step.

9 Run the TSO command TRACERTE specifying the server name. The results of the TRACERTE command show the paths that the IP address takes to be resolved. (This command requires that the issuing user ID be listed in the *OBEYFILE* on TCP/IP. Contact your TCP/IP administrator to make this change.)

If the output from this command indicates failure, ask your TCP/IP systems programmer to check the routes to the replicate server. This may involve LAN, bridge, and router connections.

If the output from this command indicates success, verify that the IP address is correct and save the output. Continue to the next step.

10 Execute the traceroute command from the UNIX machine. Specify the MVS server name. This shows the paths that the IP address takes to be resolved.

If the output from this command indicates failure, have the UNIX Systems Administrator check the routes to the MVS server. This may involve LAN, bridge, and router connections.

If the output from this command indicates success, verify that the IP address is correct, and save the output. Continue to the next step.

11 Ask the TCP/IP systems programmer to run the TRACE ALL and MORETRACE SOCKETS commands.

Warning! TRACE commands can create a high volume of output. Make sure that your output file is large enough to handle this output. TCP/IP can abend if a *data set full* condition exists.

Record the following information about the system on which the trace ran:

- TCP/IP address space name (job name)
- MVS IP address
- Primary Replication Server machine IP address and name
- Replication Server machine port
- TCP/IP version and maintenance level
- MVS version and maintenance level

12 Send all information collected in steps 1 through 11 in electronic format to Technical Support.

See also

- Chapter 8, "Configuring TCP/IP for Replication Agent" in the Replication Agent for DB2 UDB *Installation Guide* for OS/390 for information on editing the *SYGWHOST* macro
- Sybase Replication Server documentation
- IBM documentation for TCP/IP for OS/390

Replication Agent connectivity failure

A message similar to the following in the Replication Agent log indicates that you are not connecting to Replication Server:

```
Failure to connect source lti
```

❖ **To resolve Replication Agent connectivity failure**

1 Log in to Replication Server.

2 Execute a Replication Server admin who command to make sure the *dataserver*.*database* variables in the connection name match the values of the RS_source_ds and RS_source_db LTM for MVS configuration parameters.

3 Make sure the values of the RS_user and RS_pw configuration parameters are correct in the Replication Agent configuration file.

4 Make sure that the Replication Server user ID, identified in the values of the RS_user and RS_pw configuration parameters, is created and granted connect source authority on the Replication Server to which you are trying to connect.

5 If this does not resolve the error condition, issue the create connection command with the dsi_suspended clause to any Replication Agent or Mainframe ClientConnect listed in the interfaces file on the machine that contains the primary Replication Server.

6 Start Replication Agent.

See the Replication Agent for DB2 UDB *Installation Guide* for OS/390 for information about permissions in replication systems.

Replication Agent started task failure

The following message indicates that the Replication Agent started task failed for security reasons:

```
USER      IS NOT AUTHORIZED TO EXECUTE PLAN
LEDB2RELLEDB2RELLEDB2REL
```

where LEDB2REL is the name of the log extract plan.

To correct this error, be sure the LTM for MVS startup PROC is associated with the LTMADMIN user ID in the security started procedure table.

❖ **To correct Replication Agent started task failure**

- Have the MVS systems programmer associate the LTMADMIN user ID with the LTM for MVS PROC by adding them to the security system started task authorization table. These names were recorded on the Replication Agent Installation Worksheet during installation.

  You should also grant execute permission to the plan using SQL syntax like the following:

  `GRANT EXECUTE ON PLAN` *plan_name* `TO` *authorization_name*

  where *plan_name* is the name of your log extract plan (LEDB2RELLEDB2REL, for this release), and *authorization_name* is the name of the user to whom you are granting permission.

Replication Agent configuration errors

Check this list for possible Replication Agent configuration errors:

- If any parameter is set to insert a null in a replicate column, make sure the column allows nulls in all the replicate tables.
- The Replication Agent configuration file is case sensitive, so make sure the case of all parameter values in the file is consistent.
- Make sure that any Replication Agent configuration parameters listed as optional have a value entered, are commented out, or are deleted. Leaving a parameter name without a value in an uncommented state can cause a syntax error.
- If the value of the retry configuration parameter is too low, your CPU usage can be unnecessarily high during network outages. This is because the host retries the connection every *n* seconds, where *n* is the value specified in the retry configuration parameter. If the value is too high, delays in processing can occur following network outages. Valid values range from 1 (one second) to 86400 (one day).

See Appendix A, “LTM for MVS Configuration Parameters,” for more information on the retry configuration parameter.

DB2 authorization errors

If you are running Replication Agent as a started task and the security system started task table is not updated to associate the task with the *LTMADMIN* user ID, this error may occur:

```
USER IS NOT AUTHORIZED TO EXECUTE PLAN log_extract_plan_name
```

If you encounter this error, follow the instructions in the section about granting LTMADMIN permissions in Chapter 4, “Preparing for Installation” in the Replication Agent for DB2 UDB *Installation Guide* for OS/390.

Monitoring LTM for MVS and Replication Server

While Replication Agent is operating, you might need to monitor or troubleshoot its components and processes. This section describes how to monitor the status of your replication system by entering operator commands from the MVS console.

Monitoring LTM for MVS includes the following tasks:

- Entering operator commands
- Using trace commands for troubleshooting

Entering operator commands

Enter operator commands from the MVS console using the MODIFY command.

The following is the syntax of the MODIFY command (include the commas in this command as shown in the example):

```
F jobname,target,command
```

where:

- F is the MODIFY command keyword.
- *jobname* is the LTM for MVS started task name or batch job name, depending on whether LTM for MVS was initiated as a started task or batch job.
- *target* is either of the following options:
  - C specifies that the command is passed to LTM for MVS.
  - E specifies that the command is passed to Replication Extract.
- *command* is the MVS operator command you want to execute.

Displaying the software version

To view the Replication Agent software version information, start LTM for MVS and view the VERSION DD statement, which should contain a string similar to the following example:

```
Replication Agent for DB2
LTM FOR MVS/R12.6/P/IBM S390/ESA/0/OPT(1)/CDR19990917 Mar 02 2003 14:23:07
```

You can also view the version string by entering the following command at the system console:

```
f jobname,c,version
```

If you do not see a version string resembling the one above, notify the Technical Support contact for your site.

Using trace commands for troubleshooting

Use the MVS operator commands documented in this section to obtain information about LTM for MVS network performance.

The LTM for MVS trace command starts the function, which dynamically turns on and off the trace parameters in the LTM for MVS configuration file. Use the LTM for MVS notrace command to stop sending information to the *SYSPRINT* file.

Warning! Remember to turn off trace functions after you obtain the necessary information. If you allow trace functions to continue, the trace output files can fill and consume disk space, causing abends or impaired LTM for MVS performance.

Tracing LTM for MVS LTL output

The following trace command sends data that is passed to Replication Server to the *LTLOUT* file. This enables you to examine *LTLOUT* to determine whether LTM for MVS is performing properly.

❖ **To trace LTL output**

1 Start this trace by entering the following command from the MVS operator console:

```
F job_name,C,trace=LTLebcdic
```

2 End this trace by entering the following command from the MVS operator console:

```
F job_name,C,traceoff=LTLebcdic
```

See Appendix A, “LTM for MVS Configuration Parameters,” for more information on the trace=LTLebcdic and trace=LTLascii parameters.

Tracing LTM for MVS performance

The following trace command sends timestamped messages, which indicate task begin and end times, and a network response time trace to the *LTMTRACE* log file. This can help you obtain information about LTM for MVS and network performance.

❖ **To trace LTM for MVS performance**

1 Start this trace on a job by entering the following command from the MVS operator console:

```
F job_name,C,trace=Network
```

2 End this trace by entering the following command from the MVS operator console:

```
F job_name,C,traceoff=Network
```

See Appendix A, "LTM for MVS Configuration Parameters," for more information on the trace=Network parameter.

Verify whether Replication Server is running

Verify whether your primary and intermediate Replication Servers are running.

Check the Replication Server log

The Replication Server log contains error and warning messages. The most recent messages are at the end of the log.

Display information about stable queues

You can check the Replication Server queues to determine which transactions are being processed or ignored, and to determine whether transactions are open or not being committed.

To display information about SQM and SQT threads

1 Execute the admin who, sqm command against the primary Replication Server.

2 View the results to determine the number of duplicate messages being detected and ignored, and the number of blocks being written in the Replication Server stable queues.

3 Execute the admin who, sqt command against the primary Replication Server.

4 View the results to find open transactions.

See the Replication Server *Commands Reference* for information about the admin who commands and results.

Replication failure troubleshooting tips

Use the procedures in this section to diagnose the source of replication failure.

Verify names in Replication Server log

Check the Replication Server log to verify whether the values for *dataserver* and *database* you specified in your create connection command exactly match the values you specified in the RS_source_db and RS_source_ds configuration parameters.

Verify the existence of the RS user ID

Executing the connect source lti command accomplishes three things:

- Verifies that the connection to Replication Server exists
- Verifies that the RS user ID has permission to connect to Replication Server
- Returns a version string that displays the Replication Agent software version

❖ **To verify that the RS user ID exists and has appropriate permissions**

1 Log in to Replication Server as the user name you specified as the value of the RS_user configuration parameter (see the Replication Agent Installation Worksheet for this value).

2 Execute the connect source lti command using the following example syntax:

```
connect source lti data_server.database 101
```

where *data_server* is the value you specified for the RS_source_ds LTM for MVS configuration parameter and *database* is the value you specified for the RS_source_db configuration parameter (see the Replication Agent Installation Worksheet for these values).

See the Replication Server *Design Guide* for information about the connect source lti command.

Check the LTMOBJECTS table for error information

You can examine the contents of the LTMOBJECTS system table on DB2 to determine causes of replication errors.

❖ **To check the contents of the LTMOBJECTS table**

1 Execute a SELECT * on the ERROR_CODE column in the LTMOBJECTS system table on DB2 using the following syntax:

```
SELECT ERROR_CODE FROM LTMADMIN.LTMOBJECTS
```

where *LTMADMIN* is the LTMADMIN user ID you recorded on the Replication Agent Installation Worksheet during installation.

2 If the column ERROR_CODE contains a value other than zero, check the contents of message LEX0047. This message contains information about the table containing the error, and the type of error that stopped replication on that table.

See also

- "LTMOBJECTS table contents" on page 64

Check the LTM logs

The LTM logs contain warning and error messages, as well as information about the connections to Replication Server and DB2.

Check your target databases

Issue the admin who_is_down command to determine whether any of the connections to the replicate databases are down or have been suspended by Replication Server.

Check your replication definitions and subscriptions

Verify that you created replication definitions with the appropriate information.

Verify that you defined and activated subscriptions for each of your replication definitions.

Verify that table names specified in the replication definition match table names and or replicate names defined in *LTMOBJECTS*.

Check the Rep Server error log

The Rep Server error logs contain warning and error messages.

Check the ASE error log

The ASE error logs contain warning and error messages about your Adaptive Server and RSSD.

Verify whether maintenance user ID owns transactions

If the value of the LTM_process_maint_uid_trans configuration parameter is set to N (its default value), LTM for MVS does not process source database changes made by the Replication Server maintenance user ID.

See “LTM_process_maint_uid_trans” on page 135 for more information.

Replication Extract and error handling

When Replication Extract detects a replication error, it updates the row associated with the table in the LTMOBJECTS table with diagnostic information that describes the error. Replication Extract follows this procedure:

1 Replication Extract sets the contents of the ERROR_CODE column in the LTMOBJECTS table to a value greater than 0.

2 Replication Extract places the QID of the record containing the error in the QID column.

Note You can specify Terminate as the value of the LogExtractError parameter in the configuration file to force Replication Extract to terminate when it detects replication errors.

This section contains information on using the Replication Agent LTMOBJECTS system table to troubleshoot and correct replication errors.

See also

- "Understanding and using the LTMOBJECTS table" on page 62

Restarting replication on primary tables containing errors

When Replication Extract detects an error, it updates the row in the LTMOBJECTS table corresponding to the table you are replicating. At this time, the ERROR_CODE column contains a non-zero number and replication of that table discontinues until you correct the problem.

After you correct the problem, restart replication on the table using *one* of the following procedures. Sybase recommends using the first procedure.

❖ **To restart replication on the primary table using SPUFI**

1 Use SPUFI to issue a DELETE on the LTMOBJECTS table row associated with the primary table.

2 Use SPUFI to reinsert the row you deleted in step 1 into the LTMOBJECTS table.

3 In the LTMOBJECTS table:

   a Set the contents of the ERROR_CODE column to 0.

   You can find sample SQL in the *SETRPERR* member of the *hlq.DOCS* data set.

b Replace the contents of the QID column with blanks (spaces).

See also

- "Replication Agent Distribution Tape Contents," in the Replication Agent for DB2 UDB *Installation Guide* for OS/390 for information about the *SETREPERR* member of the *hlq.DOCS* data set
- Chapter 2, "Replication Server Setup"
- Chapter 4, "Managing Replication Agent"
- Chapter 5, "Known Issues and Error Recovery"

Log extract plan authorization error

If you run Replication Agent as a started task and the security system started task table is not updated to associate the task with the LTMADMIN user ID, you may encounter the following error message:

```
USER      IS NOT AUTHORIZED TO EXECUTE PLAN LEDB2REL
```

where LEDB2REL is the name of the log extract plan.

To correct the security problem:

Have your MVS systems programmer add the *LTMxxxxx* PROC and the *LTMADMIN* user ID to the security systems started task authorization table.

You should also grant execute permission to the plan using SQL syntax like the following:

`GRANT EXECUTE ON PLAN` *plan_name* `TO` *authorization_name*

where *plan_name* is the name of your log extractor plan (LEDB2RELLEDB2REL, for this release), and *authorization_name* is the name of the user to whom you are granting permission.

For the values of *LTMxxxxx*, see section 2b of the Replication Agent Installation Worksheet in the Replication Agent for DB2 UDB *Installation Guide* for OS/390).

LTMADMIN user ID permissions error

In data-sharing environments, if Replication Extract is unable to determine the high LRSN value, you can receive the following Replication Extract error messages:

```
LEX0073 DB2 Authorization Error rc=00000008 reason=00E60824. Check user
privileges.
LEX00027 Sybase Log Extract unable to retrieve DB2 high written RBA.
LEX0059 Ftch for high LRSN failed.
```

This condition can indicate that the LTMADMIN user ID does not have the required permission to execute MONITOR2 commands on DB2.

❖ **To grant LTMADMIN permission to execute MONITOR2 commands**

- Grant the LTMADMIN user permission to execute MONITOR2 commands on DB2 using the following syntax:

  ```
  GRANT MONITOR2 to LTMADMIN_user_ID
  ```

  where *LTMADMIN_user_ID* is the LTMADMIN user name you recorded in section 1a on the Replication Agent Installation Worksheet.

CPU and latency

The following sections describe possible causes of high CPU usage and high latency.

CPU

High CPU can result from any of the following factors:

- A low value setting for the PollInterval parameter — a low PollInterval value increases CPU usage and the number of I/O operations Replication Extract performs.
- A low value for the CIMAX parameter — a low CIMAX value increases the number of I/O operations issued. The effect of the CIMAX parameter is similar to the PollInterval parameter.
- A low value for the retry configuration parameter — if the value of the retry configuration parameter is too low, your CPU usage can be unnecessarily high during network outages. This is because the host retries the connection every *n* seconds, where *n* is the value specified in the retry configuration parameter. If the value is too high, delays in processing can occur following network outages.

Latency

High latency can result from any of the following factors:

- A high value setting for the POLL_INTVL parameter — a high POLL_INTVL value increases latency time.
- A high value setting for the CI_COUNT parameter — the effect of the CI_COUNT parameter is similar to the POLL_INTVL parameter.

Note For low-activity periods, use the Low_activity_latency parameter to control the length of time Replication Agent is quiescent during a low-activity period. This parameter can be used to conserve resources when the demand on Replication Agent is low.

High latency may also result from factors described in "CPU."

DB2 SQL errors

DB2 SQL-911 errors can indicate that the system tables are in the same tablespace as the primary server.

SQL-911 errors are similar to the following:

```
-911 THE CURRENT UNIT OF WORK HAS BEEN ROLLED BACK
     DUE TO DEADLOCK OR TIMEOUT.
```

See IBM DB2 documentation for complete information on DB2 SQL errors.

See also

- Documentation for the software component that issued the error
- IBM DB2 documentation

CHAPTER 5

Known Issues and Error Recovery

This chapter explains the known restrictions on Replication Agent and provides error recovery procedures. The information in this chapter allows you to verify that an error is not a result of known restrictions or issues.

Known issues and restrictions

This section describes known issues and datatype restrictions.

MVS issues

This section describes MVS-related restrictions on the operation of Replication Agent.

DASD utility conflict when archiving to disk

Using the STOPX37 DASD space management product can cause errors when using logs archived to disk.

Replication Server Manager restrictions

Sybase Replication Server Manager (RSM) is not designed to manage Replication Agent for DB2. Replication Agent is not an Open Server application, which is required for Replication Server Manager compatibility.

DB2 restrictions

The following subsections detail restrictions on DB2 functionality.

Table schema changes

Replication Agent does not detect data definition language (DDL) changes, such as dropping a table or adding a column to an existing table, in tables marked for replication.

Dropping and re-creating a table can change its object ID (OBID). Replication Agent does not recognize log entries for the new OBID until Replication Agent restarts.

Making DDL changes to existing tables

You can make data description language (DDL) changes to existing tables.

❖ **To make DDL changes to a table marked for replication**

1 Ensure delivery of any in-flight transactions for the table.

2 Turn replication off for the table.

3 Make DDL changes to the table.

4 Use the DB2 COPY utility to make an image copy of the table.

5 Run the DB2 REORG utility on the table.

6 Use the DB2 COPY utility to make an image copy of the table.

7 Reconcile the replication definition and replicate tables.

8 If the value of the Use_repdef configuration parameter is set to Y, stop and restart Replication Agent.

9 Turn replication on for the table.

DB2 utilities

If you run the DB2 RECOVER (to a previous point in time) utility against your DB2 primary database, you can jeopardize the integrity of the replicate databases. You may need to rematerialize your replicate database as a result. Replication Agent does not recognize or warn the user when this utility executes.

Do not run any DB2 utilities against your primary tables or primary tablespace during materialization. These DB2 utilities change the DB2 database without providing information about those changes to the log, thus threatening replication integrity.

DB2 referential integrity rules

ASE does not support the DB2 referential integrity verbs ON DELETE CASCADE and ON DELETE SET NULL. ASE does support the referential integrity verb ON DELETE RESTRICT.

Do not define referential integrity rules in the replicate tables. DB2 enforces referential integrity rules on the primary tables. The updates and deletes that result are correctly replicated.

System tables location

Put the Replication Agent LTMOBJECTS system tables in a different database or tablespace from your primary tablespace. If the system tables share a tablespace with your primary table, locks can prevent updates to the primary table.

See Chapter 5, “Known Issues and Error Recovery” for more information on these tables.

DB2 logs and transaction replication

DB2 does not log update transactions in which the values are identical to the values in DB2 before the update. Replication Agent does not replicate transactions of this nature because the transactions do not appear in the DB2 logs.

Datatype restrictions

This section contains restrictions that apply to DB2 datatypes when using Replication Agent.

CHAR for bit data support

Replication Agent replicates DB2 char for bit data columns as binary data.

Graphic and binary datatype support

Replication Agent can replicate DB2 graphic and vargraphic columns as binary data.

Replication Agent cannot replicate DB2 long vargraphic data but can replicate nulls in primary columns of this format.

Datatype length restriction

Replication Agent limits long varchar to a length of 32704KB.

Stored procedure restrictions

You cannot replicate stored procedures against DB2 tables.

See also

Replication Server *Commands Reference* for information about Replication Server stored procedures

Error recovery procedures

Replication Agent recovers automatically in some situations and requires intervention for recovery in other situations.

Automatic recovery

Automatic recovery occurs for these events:

- Network failures

- Machine crashes of the processing unit (not of a disk)
- Data server failures where the log transactions are not lost
- Replication Server shutdown
- Replication Agent shutdown

Intervention recovery

These recovery situations require intervention:

- Incorrect or missing server entries in the RSIPAddress and RSPort configurations
- Full stable device or other stable device failure
- Lost primary database log
- Unrecoverable primary database or RSSD failure

Current recovery

A current recovery due to a hardware or database failure is less serious. The replicated table is usually accurate and healthy. Recovering the primary table brings the replicated table back in synchronization.

The RECOVER TABLESPACE utility prevents updates to the table during recovery. Do not suspend replication in this case.

Note To recover to a current point in time using the RECOVER TABLESPACE utility, run the DB2 RECOVER TABLESPACE utility with no options.

DB2 subsystem crash

If the DB2 subsystem crashes, restart it. DB2 issues rollbacks for all uncommitted transactions. Replication Server replicates only committed database changes. Recovery is not required.

Primary table recovery

If logical or physical DB2 errors occur, you must run the DB2 RECOVER TABLESPACE utility to rectify a point-in-time recovery.

❖ **To recover to a prior point in time using the RECOVER TABLESPACE utility**

1 Stop replication on the primary DB2 table by selecting option 2 on the Activate and Deactivate Replication panel.

2 Drop the subscription.

3 Run the DB2 RECOVER TABLESPACE utility.

4 Redefine the subscription.

5 Rematerialize all the subscriptions using data from the recovered DB2 table.

6 Start replication on the primary DB2 table by selecting option 1 on the Activate and Deactivate Replication panel.

See Chapter 4, “Managing Replication Agent” for more information on primary tables.

Primary table drop

If the primary table is dropped, you may need to rematerialize or re-create the replication definitions and subscriptions. Then, you should manually verify that the tables are synchronized again.

LTMOBJECTS system table loss or corruption

If the LTMOBJECTS system table is either lost or corrupt, you must recover the table by using the DB2 RECOVER utility.

See “Primary table recovery” on page 92 for more information.

Bootstrap data set loss or clearance

If the bootstrap data set (BSDS) is lost or cleared, the Replication Extract task cannot locate the log records for replication beyond the current log or any entries not removed from the BSDS. This is not a problem unless you need older logs for recovery.

DB2 log deletion or loss

If the DB2 log is deleted or lost, you must rematerialize all replicate tables.

DB2 RECOVER utility

A logical or physical DB2 error can occur. Correct the error by running the DB2 RECOVER utility. For instructions, see “Primary table recovery” on page 92.

No unique index on primary table

If there is not a unique index on the primary table, duplicate rows can exist. Duplicate rows can fail to insert at replicate sites.

❖ **To delete duplicate rows**

1 If you did not change the error function class for the Replication Server connection, close the connection.

2 Check error codes to find the duplicate rows and delete extra copies.

QIDs, transaction IDs, and interleaving transactions

The origin queue ID is a unique, ascending binary number associated with each log record. Origin queue IDs must always be ascending. Replication Server ignores operations that have origin queue IDs equal to or lower than the queue ID of the most recent transaction operation received.

The transaction ID contains part of the DB2 subsystem ID, the URID, the MEMID, and a DB2 timestamp.

Log records for multiple transactions can be interleaved. The following example shows valid interleaving of log records (where *Qxxx* is the queue ID, and *Txxx* is the transaction ID):

```
Q200T200update, Q201T200insert, Q202T202insert,
Q203T200delete, Q204T202update, Q205T205insert,
Q206T200insert, Q207T205delete...
```

See the Replication Server *Design Guide* for origin queue IDs and transaction IDs.

Replication Server errors

Check the Replication Server *Troubleshooting Guide* for the most accurate and up-to-date error resolution information. This section is provided for your convenience.

Binary zero in primary table data

If your primary table contains a binary zero in the mainframe supported datatype char, and the value of the LTM for MVS Minimal_cols configuration parameter is N, Replication Server reports an error similar to this:

```
No terminating quote found.
```

This error can also occur if an ending quote is missing.

❖ **To correct a binary zero in primary table data error**

1. Make sure the binary zero does not pass from the primary table if you cannot find a normal source for this error.
2. If the binary zero does pass from the primary table, replace the binary in the replicate data so this error does not occur.

See also

- The section about the replace_null_in_char parameter in Appendix A, “LTM for MVS Configuration Parameters”
- Appendix F, “Creating User Exits,” for information about creating a user exit that removes binary zeros from DB2 tables

Sybase Open ClientConnect error

The following Open ClientConnect error indicates that you did not successfully connect to Replication Server:

```
Failure to connect source lti
```

See Sybase Open ClientConnect documentation for more information on Open ClientConnect.

DB2 failure and recovery

If DB2 fails in any of the following ways, use the recommended recovery procedure.

DB2 SQL locks

If you receive DB2 SQL-911 errors, be sure that the LTMOBJECTS system table does not share a tablespace with a primary table.

❖ **To correct DB2 SQL-911 errors**

1 Use the same user ID and password that originally created the system tables (usually LTMADMIN).

2 Shut down Replication Agent.

3 Save the data from the system tables using your site's normal backup procedures.

4 Drop all three system tables.

5 Create all three system tables in a different tablespace or table blocks.

6 Restore the contents of the system tables saved in step 3.

Replication failure

If replication stops on a table incorrectly, make sure that you do not drop and re-create the table while Replication Agent is running. Replication Extract uses the *OBID* of the table when it scans the log for items to replicate. This *OBID* can change when a table is dropped and re-created, and replication ceases.

❖ **To avoid dropping and re-creating tables during replication**

1 Allow Replication Agent to complete processing log records for the table containing the replication error.

2 Stop Replication Agent before dropping and re-creating primary tables.

❖ **To correct errors caused by dropping and re-creating tables**

Use *either* of the following procedures:

1 Stop and restart Replication Agent.

2 Use the Replication Agent interface panels to delete the table name from LTMOBJECTS; then replace the table name.

Either of these procedures cause Replication Extract to associate the new *OBID* with the table to replicate.

Warning! Missed transactions can occur during the time the *OBID* is not recognized, so you may need to rematerialize the table.

See also

- Chapter 2, “Replication Server Setup”
- Chapter 4, “Managing Replication Agent”

Replication Extract failure and recovery

If Replication Extract fails in any of the following ways, use the recovery procedure recommended for that situation.

Replication Extract task abend

If the Replication Extract task abends, the LTM task controller shuts down the other tasks in the address space and ends the started task. The MVS Operator must restart the LTM task.

Inspect the LTMLOG data set to determine the cause of the abend.

DB2 log failure

If a failure occurs on the DB2 log that requires the source of the error to be removed before reliable operation can resume, the Replication Extract task issues a message to the MVS console indicating that the DB2 log cannot be read. The Replication Extract task then shuts down. You can restart the task when the problem with the log is corrected.

Full output queue

If the output queue becomes full, the LTM task issues a return code 4 to Replication Extract, and Replication Extract sleeps for the amount of time specified by the LTM for MVS retry configuration parameter before attempting again to add data to the output queue.

Replication Extract and Replication Server

Replication Extract ensures reliable recovery of the DB2 changes for tables marked for replication. Replication Agent coordinates with Replication Server by using the truncation-point protocol for managing stable queues.

Replication Server failure and recovery

If Replication Server fails in any of the following ways, use the recovery procedures recommended for that situation.

Replication Server termination

If Replication Server terminates, Replication Extract relies on the LTM Locator protocol to maintain a consistent state with Replication Server through all terminations.

Replication Server connection failure

If the network link to Replication Server fails, Replication Agent retries the connection as many times as specified by the value of the Maximum_connect_retries configuration parameter. Each retry occurs at an interval specified by the retry configuration parameter.

If Replication Agent cannot connect to Replication Server, LTM for MVS fails.

To correct a connection problem:

- Check the Replication Server logs for information about the nature of the problem.
- Refer to TCP/IP documentation for your site.

See also

- Appendix A, "LTM for MVS Configuration Parameters," for information about the LTM for MVS Stop_on_error configuration parameter

Replication Server-to-Gateway connection failure

If the Replication Server-to-gateway software network link fails, automatic materialization also fails. Issue an admin who or admin who_is_down on the primary Replication Server to determine the status of the communications link. Then re-establish the communications link. Replication Server resumes normal operation.

Connection cycling repeatedly

If your connection terminates and tries to connect repeatedly, the cause may be a log record that cannot replicate. Such a record causes Replication Agent to terminate, reconnect, and go to the queue ID specified by Replication Server that precedes the log record it cannot replicate. Replication Agent terminates the connection and begins again.

To correct this problem, find the name of the table encountering the error by examining the Replication Server or Replication Agent error log. Then, choose one of the following courses of action, depending on the circumstances of the failure.

Note Message LEX0025 is normal upon starting Replication Agent.

❖ **If message LEX 0025 displays on the console repeatedly**

1 Shut down Replication Agent.

2 Determine whether open transactions still exist in the stable queue by issuing the admin who, sqt command.

3 If open transactions exist in the stable queue, contact your Replication Server Administrator for advice, and use either sysadmin purge_all_open or sysadmin purge_first_open.

Note Use the Replication Server documentation to solve stable queue problems. This does not imply that this is the only circumstance under which open transactions can exist in the stable queue.

4 If open transactions do not exist in the stable queue, prevent replication on the table in error.

For the column in the LTMOBJECTS system table representing the table containing the error, insert a value of 20 in the ERROR_CODE column.

You can edit and run the sample SQL in the SETREPERR member of the *hlq*.DOCS data set to make this change in your LTMOBJECTS system table. (See “Understanding and using the LTMOBJECTS table” on page 62 for details.)

5 Rematerialize the table that contained the error.

6 Restart replication on the table that contained the error by resetting the ERROR_CODE column value to 0 (zero).

7 Start Replication Agent.

See also

- Chapter 5, “Known Issues and Error Recovery”
- Replication Server *Commands Reference* for information about Replication Server commands

❖ **If message LEX 0025 does not display on the console repeatedly**

1. Stop replication on the primary DB2 table encountering the error by inserting a value of 20 in the ERROR_CODE column for that table in LTMOBJECTS.
2. Start Replication Agent and let it read past the log record that it cannot replicate.
3. Resynchronize the primary and replicate tables either by issuing individual updates, inserts, or deletes or by rematerializing the affected table.
4. Start replication on the primary DB2 table by selecting option `1` from the Activate and Deactivate Replication panel.

See also

- "Replication Server failure and recovery" on page 97

Replication Server stable device failure

If the Replication Server stable device fails, Replication Agent returns to the earliest missing log record and begins processing log records from that point. There is no Replication Agent recovery mode because Replication Agent restores the missing log records.

A sufficient number of log records exist because of the archiving technique of DB2. The DB2 database administrator is responsible for maintaining the archive log data sets for the recovery period required by your site.

Understanding Replication Server and DB2 termination

Any change to data marked for replication on a DB2 table must be conveyed to the associated Replication Server when it is in any of the following modes:

- In active mode (as it is read from the log)
- Termination mode (as DB2 is terminating)
- Initialization mode (as DB2 is initializing, possibly after an incomplete or abnormal termination)

Understanding a connection-to-Replication Server failure

For information about the cause of a failure in the connection to Replication Server, check the Replication Server error log.

Replication Server buffer problem

You may encounter Replication Server error message 14081 ("Command of length *xxxxx* too long for SQL write"), indicating that Replication Server rejected a 16KB buffer sent by Replication Agent as an 18KB buffer, if your site meets all of the following conditions:

- You are using Replication Server release 11.*x*.
- You are using TCP/IP connectivity.
- The value of LTM for MVS Minimal_cols configuration parameter is N.
- You are using large tables with long data names.

For assistance with this error, contact Sybase Technical Support, and reference problem 1005-8033.

Recovering from Replication Agent failure

This section describes sources of Replication Agent failure and recommends procedures for recovery.

Handling STOP conditions

A STOP condition is generated by MVS STOP and MODIFY console commands so that the Replication Extract program and Replication Agent can terminate replication. The shutdown of the Replication Extract program and Replication Agent must be coordinated to allow processing to end in an orderly fashion.

Generating STOP conditions

You can generate a STOP condition by entering either of the following MVS console commands:

```
P jobname
F jobname,C,STOP
```

where *jobname* is the LTM for MVS start-up PROC name (LTM*xxxxx*) or batch procedure name, depending on whether LTM for MVS was initiated as a started task or batch job.

Processing STOP commands

When it receives a STOP command, Replication Extract must perform the actions necessary to shut down in an orderly fashion. This includes freeing memory, closing files, issuing messages, and other tasks necessary for cleanup when shutting down.

See also

- Chapter 2, “Replication Server Setup”
- “Installing Replication Agent” in the Replication Agent for DB2 UDB *Installation Guide* for OS/390 for information about the LTM for MVS start-up procedure (LTM*xxxxx*)

Replication Server transaction failure

If a transaction failure occurs between Replication Agent and Replication Server, Replication Extract and LTM for MVS use an Open ClientConnect-style API with inherent session and transaction protocols for recovery.

LTM for MVS termination

Replication Extract relies on the Replication Server to maintain a recoverable, consistent state. When you restart Replication Agent, Replication Extract uses the Replication Server queue to obtain the origin queue ID (also called the LTM Locator) of the last committed transaction operation. The origin queue ID serves as a starting point from which to read the DB2 logs.

Character translation errors

An invalid syntax error message or a Replication Server core dump can indicate problems translating special characters. If you encounter these errors, contact Technical Support.

See “Understanding and using translation tables” on page 60 for more information on translation of character data.

See “If you need help” on page xv for instructions on using Technical Support.

Using tracing options

Run traces to obtain information about troubleshooting your LTM for MVS configuration file and user exit file.

See “Monitoring LTM for MVS and Replication Server” on page 77 for information about monitoring Replication Agent.

See also

- Appendix B, “Replication Extract Messages”
- Appendix C, “LTM for MVS Messages”

APPENDIX A

LTM for MVS Configuration Parameters

This chapter contains information about the LTM for MVS configuration file and the parameters that control the behavior of the LTM for MVS component of Replication Agent.

Understanding the LTM for MVS configuration file

When LTM for MVS starts, it reads the LTM for MVS configuration file, which contains parameters that control the behavior of the LTM for MVS component of Replication Agent.

The configuration file contains parameters and their associated values. Each line containing a configuration parameter is followed by the = (equals) character, followed by the value. LTM for MVS ignores blank lines and lines beginning with the # or * character.

Sample LTM for MVS configuration file

The following example shows the contents of the sample configuration file, located in the *LTMCNFG* member in the *hlq.JCL* data set.

Example

```
#-------------------------------------------------------------

*---------------------RS configs------------------------------
*Parameter names are not case sensitive.
```

```
*-------------------------------------------------------------
LTL_test_only=N                 Y or N to turn off connection to RS
RS=DevUTF8_RS                   Rep Server (RS) to connect to
RS_user=sa                      RS User used for RS connection
RS_pw=                          RS password for RS User
RS_source_ds=tinker             What the RS calls the DB2 server and
RS_source_db=db4x               the database; that is, bc6, dbc7, db1x
RS_ccsid=819                    The RS Codeset default is 819
RSCsetname=iso_1                Charset to log in to Rep Server
RSHost=Perro                    The name that is looked up with DNS
RSIPAddress=usehost             The IP nn.nn.nn.nn for the RS or usehost
TCPName=TCPIP                   Name of the TCPIP address space
RSPort=6001                     and port; for example, 6030
Communications Protocol-IBMTCP  Currently the only supported protocol
*-------------------------------------------------------------
*
*---------------RSSD configs if using REPDEFS-----------------
RSSD_server=RSSD_DevUTF8       The ASE Server
RSSD_user=sa                   The ASE user
RSSD_pw=                       ASE password
Use_repdef=Y                   Y or N to turn on/off use of repdefs
RSSDCsetname=iso_1             Charset to log in to RSSD Server
RSSD_database=DevUTF8_RS_RSSD  The ASE database for Repdefs
RSSDHost=Perro                 The name of the ASE host
RSSDIPAddress=usehost          The IP for the ASE or usehost
RSSDPort=5010                  and port; for example, 6030
*-------------------------------------------------------------
#
#-------------------------------------------------------------
#
#Log Extract Configuration Parameters

#
#-------------------------------------------------------------
DataSharingoption=Multi
BSDS=DB1X,DSNDB0X.DB1X.BSDS01
BSDS=DB2X,DSNDB0X.DB2X.BSDS01
BSDS=DB3X,DSNDB0X.DB3X.BSDS01
DetectDataCapture=Yes
Log_extractor=LTMEXT
Log_identifier=DBC7
LogTrace=N
GenID=1
CIMax=20
LogExtractError=Term
Buffers=25
PollInterval=0.0.20.0
```

```
LTMPlan=LEDB2REL
SetTruncPoint=47000
Creator=DSMITH
retry=10

#------------------------------------------------------------------
#
# LTM Configuration Parameters
#
#------------------------------------------------------------------
LTM_admin_user=noneyet
LTM_admin_pw=noneyet
Codepage=500,force,utf8
API_QID_request_interval=1000
max_messages=200
batch_ltl_cmds=on
Minimal_cols=Y
Use_repdef=Y
Stop_on_error=Y
Suppress_col_names=Y
Support_DB2_comp_rec=N
API_com_test=N
LTL_test_only=N
replace_null_in_char=_
Packet_size=32k
Maximum_connect_retries=10
Minimum_year=1998
Long_varchar=Y
Low_activity_latency=10
Ltl_table_col_case=lower
DateTime_conv_err=Default
Date_conv_default=1998-12-25
Time_conv_default=11:22:33
#User_exit=HOMEDPX

*------------------------------------------------------------------
* Trace Configuration Parameters
*------------------------------------------------------------------
*trace=Calls
*trace=General
*trace=RSCommand
*trace=QIDs,25           (operator QIDs or QIDs,nn); for example, QID,25
*trace=TruncPoint
*trace=Connections
*trace=LTLebcdic
*trace=LTLascii
```

What you need to know

Case sensitivity	The keywords in the LTM for MVS configuration file are not case sensitive. You can verify whether LTM for MVS accepted the correct values from the configuration file by examining the *LTMLOG* file. After you restart Replication Agent, *LTMLOG* displays the configuration parameters and values, and configuration errors, if applicable.
One line per configuration parameter	Each configuration parameter entry must be on its own line; therefore, the LTM for MVS configuration file has no line continuation character.
Duplicate parameter entries	LTM for MVS processes duplicate entries for a single parameter in the configuration file inconsistently. In most cases, the second entry overrides the first entry. Sybase recommends that you avoid using multiple entries for single parameters in the configuration file.
Optional parameters	For optional configuration parameters you do not want to use, comment out the line in the LTM for MVS configuration file by inserting the "#" (pound) symbol in the first column of the line. Several changes were made to the syntax of the LTMCFG dataset: • Comments are now permitted after the configuration value. • Entire lines may be commented out by inserting "*" or "#". • Configuration parameter names are not case sensitive. • Most configuration values are not case sensitive. **Note** A comment cannot follow a "blank" parameter. For example, `RSSDPort= comment` is not valid.

Required parameters	Do not delete or comment out any required configuration parameters. LTM for MVS supplies default values for most configuration parameters. Change the values for these parameters for your site as needed.
Configuration errors	After modifying configuration parameters and before starting LTM for MVS, check this list for potential errors: • A line beginning with a blank is treated as a comment. • Provide valid values for all optional configuration parameters that are not commented out of the configuration file. Leaving a parameter name in an active state without an appropriate value causes syntax errors. • If a parameter is set to insert a null in a replicate column, make sure that column allows nulls. • Do not delete lines containing unused parameters. Your site may need those parameters in the future.
Adaptive Server LTM configuration parameters	LTM for MVS configuration parameters differ from Adaptive Server LTM configuration parameters.

Understanding the LTM for MVS configuration parameters

This section lists the LTM for MVS configuration parameters in alphabetical order with their valid values. Parameter descriptions include information about performance and tuning considerations where appropriate.

Note Be sure to read the information in this section before you change configuration parameter values so that you understand the consequences of the changes.

The default values for most of the parameters in the LTM for MVS configuration file enable the operation of most replication systems.

You can change parameter values to accommodate unusual environments or situations. For example, you can adjust parameter values if an error message indicates that your system has run out of configurable resources.

Table A-1 lists the configuration parameters described in this section according to their function.

Table A-1: LTM for MVS configuration parameters

Configuration parameter type	Parameter name and location
Required parameters	"Log_extractor" on page 129
	"RS" on page 143
	"RS_source_db" on page 146
	"RS_source_ds" on page 147
	"RS_user" on page 148
	"RS_pw" on page 148
	"LTM_admin_pw" on page 134
	"LTM_admin_user" on page 135
	"Communications_Protocol" on page 120
	"LTMPlan" on page 136
Performance parameters	"Low_activity_latency" on page 132
	"RSSD_database" on page 151
	"RSSD_pw" on page 152
	"RSSD_server" on page 153
	"RSSD_user" on page 153
	"Minimal_cols" on page 137
	"Suppress_col_names" on page 159
	"Use_repdef" on page 165
	"Support_DB2_comp_rec" on page 159
	"API_QID_request_interval" on page 115
	"Packet_size" on page 139
	"trace=LTLascii" on page 164
	"RS_ccsid" on page 143
	"batch_ltl_cmds" on page 116
	"RSCsetname" on page 146
	"RSSDCsetname" on page 151
Numeric and data conversion parameters	"replace_null_in_char" on page 141
	"Codepage" on page 118

Configuration parameter type	Parameter name and location
	"GraphicType" on page 128
Date and time conversion parameters	"Date_in_char" on page 124
	"Time_in_char" on page 161
	"Timestamp_in_char" on page 162
	"DateTime_conv_err" on page 125
	"Date_conv_default" on page 124
	"Long_varchar" on page 131
	"Time_conv_default" on page 160
	"Minimum_year" on page 138
Processing parameters	"Stop_on_error" on page 156
	"User_exit" on page 167
	"LTL_table_col_case" on page 132
Log extract parameters	"Creator" on page 121
	"Log_identifier" on page 130
	"DataHandler" on page 121
	"GenID" on page 128
	"TableHandler" on page 158
Communication parameters	"LTM_process_maint_uid_trans" on page 135
	"retry" on page 142
	"Maximum_connect_retries" on page 136
	"RSIPAddress" on page 149
	"TCPName" on page 158
	"RSPort" on page 150
	"RSSDIPAddress" on page 154
	"RSSDPort" on page 155
	"RSHost" on page 149
	"RSSDHost" on page 152

Configuration parameter type	Parameter name and location
Testing and debugging parameters	“API_com_test” on page 113
	“LTL_test_only” on page 133
	“LogTrace” on page 131
	“trace=Calls” on page 163
	“trace=LTLebcdic” on page 163
	“trace=QID,n” on page 165
Log extract parameters	“BSDS” on page 117
	“Buffers” on page 117
	“DataSharingMember” on page 122
	“DataSharingOption” on page 122
	“DetectDataCapture” on page 127
	“LogExtractError” on page 129
	“LTMPlan” on page 136
	“PollInterval” on page 140
	“SetTruncPoint” on page 155

API_com_test

Assists with debugging log extract or user exit applications you create. The API_com_test parameter is *optional*.

Example syntax

```
API_com_test=N
```

Default value

- N

Valid values

- N

  - Allows LTM for MVS to connect to Replication Server
  - Does *not* create *APICOM* nor *UELECMD* files
- Y
  - Allows LTM for MVS to connect to Replication Server
  - Sends Replication API call structures to *APICOM* file
  - If a user exit is loaded, sends Replication API call structures to *UELECMD* file after the user exit is invoked
- O
  - Sends Replication API call structures to *APICOM* file
  - If a user exit is loaded, sends Replication API call structures to the *UELECMD* file
  - Reads *LTMTRUNC* file for the LTM Locator value
  - Sets the maintenance user ID `FREDDY##` (where the # sign represents a space)
  - Prevents LTM for MVS from creating LTL
  - Prevents LTM for MVS from connecting to Replication Server

Comments

Use the output in the *APICOM* file to monitor Replication Extract's calls to the Replication API in the following cases:

- When a user exit is not installed
- When a user exit is installed, but before it is invoked

Use the *UELECMD* file to examine the manner in which the user exit altered the LE Command structure.

Set the value of API_com_test to Y or O, depending on whether you want to process transaction operations while debugging. Use the output in the *APICOM* file to monitor Replication Extract's calls to the Replication API without a user exit installed or before it invokes, if a user exit is installed. Use the *UELECMD* file to examine Replication Extract's calls to the Replication API after the user exit is invoked.

When you set the value of the API_com_test to N, Replication Agent does not write output to the *APICOM* or *UELECMD* files, but it allows Replication Agent to connect to Replication Server.

When the API_com_test parameter is set to Y, memory allocations and file input and output result. Sybase recommends that you set the API_com_test parameter to N for normal operation. The correct space allocation depends on the amount of data involved.

If you set the value of API_com_test to Y or O, uncomment the *APICOM* DD statement in the JCL that starts the LTM for MVS started task on MVS:

```
//APICOM DD DSN=hlq.APICOM,DISP=(,CATLG),
// UNIT=SYSDA,DCB=(LRECL=4096,RECFM=VB),
// SPACE=(CYL,(1,1))
```

Uncomment the *UELECMD* DD statement *only* if you installed a user exit in the JCL that starts the LTM for MVS task on MVS:

```
//UELECMD DD DSN=hlq.UELECMD,DISP=(,CATLG),
// UNIT=SYSDA,DCB=(LRECL=4092,RECFM=VB),
// SPACE=(CYL,(1,1))
```

See "Creating User Exits" in the Replication Agent for DB2 UBD *User's and Troubleshooting Guide* for z/OS for information about using a user exit to replace binary zeros.

See Appendix B, "Creating User Exits" in the Replication Agent for DB2 UBD *Installation Guide.*

API_QID_request_interval

Controls the number of messages Replication API sends to the LTI before requesting a new LTM Locator from Replication Server.

The API_QID_request_interval parameter is *optional*.

Example syntax

```
API_QID_request_interval=1000
```

Default value

- 1000

Valid values

- Any integer from 1 to 10000

Note In Replication Agent for DB2 UDB for OS/390 version 12.6 and later, the default value of the API_QID_request_interval parameter changes from 200 to 1000. The maximum value is currently 10000; formerly it was 1000.

Comments

The API_QID_request_interval configuration parameter controls the number of messages LTM for MVS sends to Replication Server before issuing a get truncation command to request a new LTM Locator value from Replication Server.

Issuing a get truncation command forces Replication Server to update rs_locater with the current LTM Locator for the primary database. This prevents Replication Extract from unnecessarily reprocessing transactions upon restart.

See the Replication Server *Reference Manual* for information about the rs_locater Replication Server system table.

batch_ltl_cmds

Controls whether LTM for MVS collects transactions (including inserts, updates, and deletes) in a 16K buffer and either ships them to Replication Server when the buffer is full, or sends the transactions one at a time.

The batch_ltl_cmds parameter is *required*.

Example syntax

```
batch_ltl_cmds=on
```

Default value

- on

Valid values

- on

Batches transactions in a 16K buffer and ships to Replication Server when buffer is full.

- off

  Places a single transaction in each 16K buffer it sends to Replication Server.

Performance and tuning considerations

Set the value of batch_ltl_cmds to on to reduce network traffic and increase throughput.

BSDS

If you want theBSDS data listed in the operator commands, the BSDS parameter must be added for each member of a data-sharing group.

Example syntax

```
BSDS=member,dsn
```

Example

```
BSDS=DB1X,DSNDB0X.DB1X.BSDS01
BSDS=DB2X,DSNDB0X.DB2X.BSDS01
BSDS=DB3X,DSNDB0X.DB3X.BSDS01
```

where DB1X, DB2x, and DB3X belong to data-sharing group DB0X.

Buffers

The Buffers parameter determines how many buffers the Log Extract uses to retrieve data from the log.

The Buffers parameter is not required.

Example syntax

```
Buffers=nn
```

Default value

25

Valid values

1-50

Codepage

Identifies the name of the code page that DB2 is currently using. To identify the code page DB2 is using, check the value of the CCSID initialization parameter in DB2.

The Codepage parameter is *optional*.

Warning! If you change the DB2 CCSID setting in the DSNTIPF panel, you must also do the following to ensure LTM resumes processing at the end of the DB2 log using the proper CCSID value:

1 Change the values for the Replication Agent Codepage and RS_ccsid parameters in the *LTMCNFG* member of the *hlq.JCL* data set to reflect the new CCSID value.

2 Reset the truncation point and rs_locater values to zero. See "Basic Replication Agent troubleshooting tips" on page 81 for instructions.

Options

- The force option for the Codepage configuration parameter forces Replication Agent to use the code page value provided in the Codepage configuration parameter instead of the CCSID value provided from DB2.

  Note Replication Agent now supports all valid DB2 CCSIDs, including ASCII and double-byte CCSIDs, in addition to single-byte character sets.

- The utf8 option for the Codepage configuration parameter enables replication from DB2 using a Unicode character set.

Example syntax

```
Codepage=CP500,force,utf8
```

Default value

CP500

Valid values

To determine the appropriate value for the Codepage parameter, refer to the table provided in the *CODEPAGE* member of the *hlq.DOCS* library.

Here is an excerpt of the *CODEPAGE* member:

```
|-------------------------------------------------------|
| This table lists the values to use for the Rep Agent  |
| Codepage= and RS-ccsid= parameters.  The 'DB2 CCSID'   |
| column lists the settings used on the DB2 installation|
| panel 'DSNTIPF'.  This is usually the same as the     |
| CCSID for the user database in which the primary table|
| was created.  The Rep Server/ASE column lists         |
| character set values.  The RS_ccsid value should      |
| reflect the setting of Replication Server's RS_charset|
| configuration parameter.                              |
|-------------------------------------------------------|
|           | Rep Server/ASE | Rep Agent  |  Rep Agent  |
| DB2 CCSID | ---------------| Codepage=  |  RS_ccsid=  |
|           | character set  |            |             |
|-----------|----------------|------------|-------------|
|     37    |    cp437       |     37     |     437     |
|-----------|----------------|------------|-------------|
     ...          ...             ...          ...
|-----------|----------------|------------|-------------|
|    500    |    cp437       |    500     |     437     |
|-----------|----------------|------------|-------------|
|    500    |    iso_1       |    500     |     819     |
|-----------|----------------|------------|-------------|
|    500    |    cp850       |    500     |     850     |
|-----------|----------------|------------|-------------|
|    500    | iso15(8859-15) |    500     |     923     |
|-----------|----------------|------------|-------------|
     ...          ...             ...          ...
|-----------|----------------|------------|-------------|
|   5348    |    cp1252      |   5348     |    1252     |
|-----------|----------------|------------|-------------|
```

This table matches DB2 CCSIDs with their appropriate Replication Agent Codepage and RS_ccsid configuration parameter values.

Comments

- You can verify the CCSID setting for a database by executing the following query:

  ```
  SELECT * FROM SYSIBM.SYSDATABASE
  WHERE NAME = 'database_name'
  ```

  where *database_name* is the name of the database for which you are verifying the CCSID setting.

You can also verify the CCSID settings for translation tables used by the LTM in character data conversions by executing the following query:

```
SELECT * FROM SYSIBM.SYSSTRINGS
WHERE INCCSID = inccsid AND OUTCCSID = outccsid
```

Character translation errors can occur when the value of the Codepage parameter value does not match the code page that DB2 is currently using.

See the *IBM Character Data Representation Architecture Reference and Registry*, document #SC09-2190-00, for the Character Data Representation Architecture conversion tables.

- To support replication with a Unicode character set, you must:
  - Specify the utf8 option in the LTM for MVS Codepage parameter:

    ```
    Codepage=CP500,,utf8
    ```

  - Specify a Unicode datatype (unichar or univarchar) in the replication definition for the column in the DB2 database

---

Warning! If Codepage is specified as CP500,force,utf8, then unprintable EBCDIC characters may not be replicated to char or varchar datatypes. This is not a problem when replicating to the unichar datatype.

Check for the latest EBF to address this issue (QTS #303468).

Communications_Protocol

Tells Replication Agent which protocol option your system is using:

- IBM TCP/IP

The Communications_Protocol parameter is *required*.

Example syntax

Communications_Protocol=IBMTCP

Valid values

- IBMTCP

Creator

Identifies the creator name for the LTMOBJECTS table, which is created on DB2 by the *LTMADMIN* user ID during installation.

The Creator parameter is *required*.

Example syntax

```
Creator=LTMADMIN
```

Default value

- LTMADMIN

Valid values

- Maximum of 8 characters

Note Obtain your site's value for the Creator parameter from the Replication Agent Installation Worksheet in the Replication Agent for DB2 UDB *Installation Guide* for z/OS. This value should have been recorded on the worksheet during installation.

DataHandler

Specifies the program that will process the log records to be replicated. By default, this program is LTMIFI. This configuration is set to LTMMAT when DB2 unload data is used to materialize a remote table. In a replication Toolkit implementation of the Replication Agent, you can subsitute another program that reads and builds log records from a data source other than DB2.

Example syntax

```
LTMIFI
```

Default value

- LTMEXT

DataSharingMember

Specifies the DB2 subsystem that returns log records when the DataSharingOption is specified as Single and Log_Identifier specifies a DB2 data sharing group name.

Example syntax

```
DataSharingMember=DB1X
```

where DB1X belongs to data-sharing group DB0X and DataSharingOption is Single. If DataSharingOption is not Single, this value is ignored.

Default value

- DataSharingMember=

Valid values

Any valid DB2 subsystem member name (maximum of 4 characters).

DataSharingOption

Specifies the type of data sharing environment in use in the DB2 subsystem.

If Data Sharing is not used with DB2, you must use this setting in the configuration file:

```
DataSharingOption=None
```

When Data Sharing is used, two configuration settings can be used:

```
DataSharingOption=Multi
DataSharingOption=Single
```

If the DataSharingOption is Single, you must provide the name of the Data Sharing Member to be used with the DataSharingMember configuration parameter. For example:

```
DataSharingMember=DB2X
```

Note This requirement does not apply to the Multi and None options.

Log identifiers

The Log Identifier (SSID) must be entered for all configurations. If the log identifier is set incorrectly, rollback records may not correspond to the correct LRSN/RBAs when `Support_DB2_comp_rec=Y` is used.

Log identifier examples

Log_identifier=DB0M

a group that has DB1M, DB2M, and DB3M as members

Log_identifier=DB2N

a specific SSID for the DB2, or an SSID within the data sharing group

If `DataSharingOption=Single` or `DataSharingOption=None`, the Log_identifier can be the SSID of the member or the data sharing group.

Example syntax 1

```
Log_identifier=DB0X
DataSharingOption=Multi
BSDS=DB1X,DSNDB0X.DB1X.BSDS01
BSDS=DB2X,DSNDB0X.DB2X.BSDS01
BSDS=DB3X,DSNDB0X.DB3X.BSDS01
```

Example syntax 2

```
Log_indentifier=DB1X
DataSharingOption=Single
DataSharingMember=DB1X
```

Default value

- DataSharingOption=None

Valid values

DataSharingOption=None

```
DataSharingOption=Multi
DataSharingOption=Single
```

Date_conv_default

Performs three functions:

- If the value of the DateTime_conv_err parameter is set to Default, this parameter supplies the value that LTM for MVS inserts in columns that encounter date conversion errors.
- Supplies date values during conversion of LTM for MVS date data to Adaptive Server datetime format.
- If your data contains year values less than the value of the Minimum_year parameter, then LTM for MVS inserts the value of the Date_conv_default parameter in the data you are replicating.

The Date_conv_default parameter is *optional*.

Example syntax

```
Date_conv_default=2000-12-31
```

Default value

- 1900-01-01

Valid values

- Any valid ISO date format value (*YYYY-MM-DD*, where *YYYY*=year, *MM*=month, and *DD*=day of the month)

See Chapter 6, “Working with datatype conversions” in the Replication Agent for DB2 UDB *User’s and Troubleshooting Guide* for z/OS for examples of valid date, time, and timestamp formats.

Date_in_char

Controls whether LTM for MVS sends date columns as char(10) fields or converts them to Adaptive Server datetime format.

The Date_in_char parameter is *optional*.

Example syntax

```
Date_in_char=N
```

Default value

- N

Valid values

- N

  LTM for MVS converts LTM for MVS date datatype columns to Adaptive Server datetime.

- Y

  When you set Date_in_char to Y, LTM for MVS converts dates that are out of range for the replicate table to a char(10) column.

  LTM for MVS sends LTM for MVS date datatype columns as char(10) fields (ISO format: *CCYY-MM-DD*) instead of converting them to Adaptive Server datetime format.

See Chapter 6, "Working with datatype conversions" in the Replication Agent for DB2 UDB *User's and Troubleshooting Guide* for z/OS for examples of valid date, time, and timestamp formats.

DateTime_conv_err

Controls the action LTM for MVS takes when it encounters conversion errors during conversion of LTM for MVS date, time, or timestamp data to Adaptive Server datetime.

The DateTime_conv_err parameter is *optional*.

Example syntax

```
DateTime_conv_err=Default
```

Default value

- Default

Valid values

- Default

  If the error is a date error, LTM for MVS places the value stored in the Date_conv_default parameter in the column encountering the error.

- Null

  LTM for MVS places a null in the column encountering the error.

- Datetime

  If the error is a date error, LTM for MVS places the value stored in the Date_conv_default and Time_conv_default parameters in the column encountering the error.

Note If you set the value of DateTime_conv_err to null, be sure the affected replicate columns allow nulls.

See Chapter 6, "Working with datatype conversions" in the Replication Agent for DB2 UDB *User's and Troubleshooting Guide* for z/OS for examples of valid date, time, and timestamp formats.

DateTime_conv_err

Controls the action LTM for MVS takes when it encounters conversion errors during conversion of LTM for MVS date, time, or timestamp data to Adaptive Server datetime.

The DateTime_conv_err parameter is *optional*.

Example syntax

```
DateTime_conv_err=Default
```

Default value

- Default

Valid values

- Default

  If the error is a date error, LTM for MVS places the value stored in the Date_conv_default parameter in the column encountering the error.

- Null

  LTM for MVS places a null in the column encountering the error.

- Datetime

  If the error is a date error, LTM for MVS places the value stored in the Date_conv_default and Time_conv_default parameters in the column encountering the error.

Note If you set the value of DateTime_conv_err to null, be sure the affected replicate columns allow nulls.

See Chapter 6, “Working with datatype conversions” in the Replication Agent for DB2 UDB *User's and Troubleshooting Guide* for z/OS for examples of valid date, time, and timestamp formats.

DetectDataCapture

Use the DetectDataCapture parameter to display a message when datacapture changes | none is toggled for a replicating table. Two examples of messages that could be generated are:

```
User USERID altered table TABLENAME to DATA CAPTURE
NONE.

User USERID altered table TABLENAME to DATA CAPTURE
CHANGES
```

Example syntax

DetectDataCapture=No

Default value

N

Valid values

- Y
- N

GenID

Used to increment the QID sent to a Replication Server. When a zOS Replication Sysem is first implemented, GenID is usually set to 1. The first two bytes (a short integer) of the QID value are set to this value. At restart, you may want to increment GenID by one so that Replication Server does not view the QID as lower than any QID received in the past.

Example syntax

```
Genid=1001
```

Valid values

- 1 to 32767

GraphicType

Determines whether a graphic type column in a DB2 row is replicated as a char value or as a hex string.

Example syntax

```
GraphicType=char
```

Valid values

- char
- bin

Log_extractor

Provides LTM for MVS with the name of the Replication Extract load module to be attached at start-up.

The Log_extractor parameter is *required*.

Example syntax

```
Log_extractor=LTMEXT
```

Default value

- LTMEXT

Valid value

- LTMEXT

Warning! Do not change the default value of the Log_extractor parameter; doing so causes program failure.

LogExtractError

The LogExtractError parameter determines whether the Log Extract terminates or returns a message when a log record error is encountered. When message is specified the Log Extract continues to execute after sending the message.

The LogExtractError parameter is not required.

Example syntax

```
LogExtractError=terminate
```

Default value

Terminate

Valid values

Terminate or message

Log_identifier

Provides LTM for MVS with the identifier of the subsystem containing the primary tables on DB2. This is also the subsystem on which the Replication Extract component of Replication Agent is running.

The Log_identifier parameter is *required*.

Example syntax

```
Log_identifier=DSNA
```

Default value

- none

Valid values

- A DB2 subsystem ID or group attach name (see "The Log_identifier parameter and data sharing" on page 131 for additional details)
- Maximum of 4 characters

Note Obtain your site's value for the Log_identifier parameter from the Replication Agent Installation Worksheet in the Replication Agent for DB2 UDB *Installation Guide* for z/OS. This value should have been recorded on the worksheet during installation.

Comments

When changing the value of the Log_identifier parameter, Replication Agent should have processed all records up to the end of the DB2 log. Changing the value causes Replication Extract to process transactions on a different DB2 subsystem. When Replication Extract stops processing the log for the subsystem you originally specified, open transactions on that subsystem may never reach the replicate database.

The Log_identifier parameter and data sharing

Replication Extract uses the value of the Log_identifier parameter to connect to DB2. The Log_identifier value identifies either the DB2 group attach name or a single DB2 subsystem (Sybase recommends using the group attach name as data sources for replication.

In DB2, the group attach name functions as an alias for a group of transaction identifiers. Replication Agent can use the group attach name to replicate transactions from multiple members of a data-sharing group.

LogTrace

When the LogTrace parameter is set to Y the Log Extract produces a trace in the LOGTRACE file.

The LogTrace parameter is not required.

Example syntax

```
LogTrace=N
```

Default value

Defaults to N meaning no trace will be put in the *LOGTRACE* file.

Valid values

- Y
- N

Long_varchar

Specifies size limits for char and varchar datatypes.

The Long_varchar parameter is *required*.

Example syntax

```
Long_varchar=Y
```

Default value

- N

Valid values

- Y

  Replication Agent limits char field lengths at 254 bytes and varchar fields at 32,704 bytes.

- N

  Replication Agent uses old char and varchar field limits.

Low_activity_latency

Controls the length of time Replication Agent is quiescent during a low-activity period. This parameter can be used to conserve resources when the demand on Replication Agent is low.

The Low_activity_latency parameter is *optional*.

Example syntax

```
Low_activity_latency=5
```

Default value

- 10

Valid values

- Any integer from 1 to 99.

LTL_table_col_case

Specifies the case of replicate table and column names.

The LTL_table_col_case parameter is *optional*.

Example syntax

```
LTL_table_col_case=asis
```

Default value

- asis

Valid values

Values for this parameter must be entered in lowercase.

- asis

  The replicate table name is the same as the primary table name.

- upper

  The replicate table name is uppercase.

- lower

  The replicate table name is lowercase.

Note The table names produced as a result of this parameter setting must match the table names in the replication definition and in the target database. Table and column names must be of the same case if this parameter is used.

LTL_test_only

Controls whether LTM for MVS connects to Replication Server and sends transaction operations for replication.

The LTL_test_only parameter is *optional*.

Example syntax

```
LTL_test_only=N
```

Default value

- N

Valid values

- N

  Allows LTM for MVS to connect to Replication Server and send transaction operations for replication, in normal production mode.

- Y

  Setting the value of LTL_test_only to Y results in the following actions:

  - Prevents LTM for MVS from connecting to Replication Server
  - Sends LTL to the *LTLOUT* file (in EBCDIC format)
  - Obtains the LTM Locator value from *LTMTRUNC* file
  - Sets the maintenance user ID to FREDDY## (the # symbol represents a space)

- A

  Causes Replication Agent to write data to the *LTLOUT* file in ASCII format instead of in EBCDIC format. This value is otherwise the same as a value of Y.

Comments

Set the value of LTL_test_only to Y only when debugging Replication Agent.

LTM_admin_pw

The LTM_admin_pw parameter is *required* but not used in this software version.

Although this parameter is not used, you must provide a value. Do not comment out or delete.

Example syntax

```
LTM_admin_pw=noneyet
```

Valid value

- Any value

LTM_admin_user

The LTM_admin_user parameter is *required* but not used in this software version.

Although this parameter is not used, you must provide a value. Do not comment out or delete.

Example syntax

```
LTM_admin_user=noneyet
```

Valid value

- Any value

LTM_process_maint_uid_trans

Controls whether LTM for MVS forwards transactions performed by the Replication Server maintenance user to Replication Server.

The LTM_process_maint_uid_trans parameter is *optional*.

Example syntax

```
LTM_process_maint_uid_trans=N
```

Default value

- N

Valid values

- N

  Transactions performed by the Replication Server maintenance user are *not* sent to Replication Server.

- Y

  Transactions performed by the maintenance user are sent to Replication Server for replication.

Comments

Most replication installations do not need to replicate transactions performed by the Replication Server maintenance user.

Set the value of this parameter to Y only if you are certain you need to replicate maintenance user transactions.

LTMPlan

The LTMPlan parameter is used by the Log Extract when it connects to the DB2 Subsystem.

The LTMPlan parameter is required.

Example syntax

```
LTMPlan=LEDB2REL
```

Default value

LEDB2REL

Valid values

The 1 to 8 character plan name that was bound for the Log Extract.

Maximum_connect_retries

Controls the number of times LTM for MVS attempts to restore a failed connection to Replication Server.

The Maximum_connect_retries parameter is *optional*.

Example syntax

```
Maximum_connect_retries=10
```

Default value

- 10

Valid values

- Integers from 0 (none) to 9999 (infinite). A blank value will be accepted and set to 0.

Comments

- If the LTM for MVS message queue does not contain messages to send, LTM for MVS does not attempt to restore the connection to Replication Server.

Minimal_cols

Controls whether transaction operation records for updates include the entire before image of a row and only those columns in the after image that change as a result of the update.

The Minimal_cols parameter is *optional*.

Example syntax

```
Minimal_cols=Y
```

Default value

- Y

Valid values

- Y

  Causes transaction operation records for updates to include the entire before image of a row but only those columns in the after image that change as a result of the update. If `Minimal_cols=Y`, then you must set `Use_Repdef=Y`.

- N

  Allows transaction operation records for updates to include the entire before image of a row and the entire after image of the row.

Performance and tuning considerations

Setting the value of Minimal_cols to Y can reduce network traffic, especially at sites with tables containing a high volume of columns.

Note Replication Agent limits char and varchar field lengths to 255 bytes and long varchar field lengths to 32704 bytes Replication Agent allows a maximum of 126 double-byte characters when Minimal_cols=N and 124 double-byte characters when Minimal_cols=Y.

Minimum_year

The value you supply is the minimum year value that Replication Agent can accept.

The Minimum_year parameter is *optional*.

Example syntax

```
Minimum_year=1753
```

Default value

- 0000

Valid values

- Any four-digit integer

Comments

If a date or timestamp in the source data marked for replication contains a year below the value you specify in the Minimum_year configuration parameter, then LTM for MVS inserts the value of the Date_conv_default parameter in the data you are replicating.

Updated data can fail (and Adaptive Server can fail, depending on the settings of the Adaptive Server parameters that control error behavior) under the following conditions:

- The source data contains dates before 1753.

- You specify a Minimum_year value of less than 1753.

See “Date_conv_default” on page 124 for information on the Date_conv_default parameter.

Packet_size

Packet size determines how many bytes are sent in each Replication Server command buffer. See Table A-2 for packet size information.

Table A-2: Packet size information

Packet size	Description
Packet_size=512	Minimum size; typically only useful for test purposes.
Packet_size=32K	Packet sizes in increments of 1024 bytes are required when packet size is greater than 1024 bytes.
Packet_size=32000	Use either whole numbers or abbreviate, using the K (1000) symbol.
Packet_size=1024K	Maximum packet size.

Packet size is useful when tuning Replication Agent for efficient operation in a network environment. Packet size must be larger than the largest command sent to Replication Server. A DB2 row can be as large as 32K. Determining the best packet size depends on configurations that change the size of data in a command, such as minimal columns. Inserts and deletes have a maximum size of ~32K; however, the before and after columns of an update make the maximum data size ~64K.

A minimum size of 175 bytes is needed for a typical Replication Server command. You must also allow enough space for table names, column names, punctuation, and other information necessary for a Replication Server command. If packet size is exceeded by a single command, the log extract terminates with a user abend of 1408.

Configuration packet size should be tested in a production-like environment. When testing, use the same version of Replication Server and a substantive amount of data. Changing the setting should not affect Replication Agent functionality, but it is important to notice how the entire replication environment is affected. Increasing packet size sends data over the network in larger packets, with fewer sends and receives. Latency, the frequency with which truncation points are requested, and other configuration issues should also be considered when modifying packet size.

Example syntax

Packet_size=32k

Default value

- 32K

Valid values

- Multiples of 512 from 512 to 1024000 bytes. A value in this range ending with the letter k or K is also valid.

PollInterval

The PollInterval parameter determines how often the Log Extract wakes up to check for more log records after it has reached the end of the log.

The PollInterval parameter is not required.

Example syntax

```
PollInterval=0.nn.nn.nnn
```

Default value

0.00.20.000

Valid values

Hours.Minutes.Seconds.Milleseconds or 0.00.20.0 to 0.60.60.999 (the hours value is ignored even if a non zero value is supplied).

replace_null_in_char

Specifies the value that replaces binary zero (x'00') characters in LTM for MVS char or varchar data columns, except when the value of replace_null_in_char is blank or the parameter is commented out of the configuration file.

The replace_null_in_char parameter is *optional*.

Note This parameter is commented out of your LTM for MVS configuration file. Leave it commented out unless you know you have binary zeros in LTM for MVS char or varchar columns.

Example syntax

```
replace_null_in_char= blank
```

Default value

- Null

Valid values

- Any value *except* x'00'
- To replace the null characters with blanks, set this parameter value as follows:

  ```
  replace_null_in_char=#blank
  ```

  where the # symbol represents a space.

Comments

To make this parameter active, uncomment the parameter in the LTM for MVS configuration file.

When uncommented, this parameter can impair performance. It forces LTM for MVS to scan each char or varchar field and replace each binary zero. You can use a user exit to perform this function more efficiently.

See Appendix D, “Creating user exits” in the Replication Agent for DB2 UDB *User’s and Troubleshooting Guide* for z/OS for information about using a user exit to replace binary zeros.

See Appendix B, “Creating User Exits” in the Replication Agent for DB2 UBD *Installation Guide* for further information.

retry

Controls the number of seconds LTM for MVS waits before attempting to restore a failed connection to Replication Server.

The retry parameter is *optional*.

Example syntax

```
retry=10
```

Default value

- 10

Valid values

- Integers between 1 (one second) and 86400 (one day)

Comments

If the LTM for MVS message queue does not contain messages to send, LTM for MVS does not attempt to restore the connection to Replication Server.

Performance and tuning considerations

If the value of this parameter is too low, your CPU usage can be unnecessarily high during network outages. If this value is too high, delays in replication can occur following network outages.

RS

Name of the machine where the primary Replication Server resides.

The RS parameter is *required*. The maximum length for this parameter is 30 alphanumeric characters.

Example syntax

```
RS=LTMIDRS1
```

Valid values

The value of the RS parameter for your site must exactly match the value of the *SERVERNAME* parameter in the SYGWHOST macro entry for Replication Server.

The value of the RS parameter must contain a maximum of 30 alphanumeric characters.

Note Obtain your site's value for the RS parameter from the Replication Agent Installation Worksheet in the Replication Agent for DB2 UDB *Installation Guide* for z/OS. This value should have been recorded on the worksheet during installation.

RS_ccsid

Indicates the Coded Character Set Identifier (CCSID) used by the Replication Server to which Replication Agent is connected.

The RS_ccsid parameter is *optional*.

Warning! If you change the DB2 CCSID setting, you must also do the following to ensure LTM resumes processing at the end of the DB2 log using the proper CCSID:

1 Change the values for the Replication Agent Codepage and RS_ccsid parameters in the *LTMCNFG* member of the *hlq.JCL* data set to reflect the new CCSID value.

2 Reset the truncation point and rs_locater values to zero. See "Basic Replication Agent Troubleshooting Tips" in Chapter 7, "Basic Troubleshooting Procedures" of the Replication Agent for DB2 UDB *User's and Troubleshooting Guide* for z/OS for instructions.

Example syntax

```
RS_ccsid=819
```

Default value

- 819

Valid values

- The range of valid values for the RS_ccsid parameter is between 0 and 64534 inclusive. A value in this range starting with the prefix cp or CP is also valid. For example, the user may enter 1258 or CP1258 to indicate the code page value for Vietnamese.

Note Replication Agent version 12.6 and later supports all valid DB2 CCSIDs, including ASCII and double-byte CCSIDs.

To determine the appropriate value for the RS_ccsid parameter, see the table provided in the *CODEPAGE* member of the *hlq.DOCS* library. Here is an excerpt of the *CODEPAGE* member:

```
|-------------------------------------------------------|
| This table lists the values to use for the Rep Agent  |
| Codepage= and RS-ccsid= parameters.  The 'DB2 CCSID'   |
| column lists the settings used on the DB2 installation|
| panel 'DSNTIPF'.  This is usually the same as the     |
```

CCSID for the user database in which the primary table was created. The Rep Server/ASE column lists character set values. The RS_ccsid value should reflect the setting of Replication Server's RS_charset configuration parameter.			
DB2 CCSID	Rep Server/ASE character set	Rep Agent Codepage=	Rep Agent RS_ccsid=
37	cp437	37	437
37	iso_1	37	819
...	...	...	...
5026	cp932	5026	932
5035	cp932	5035	932
5348	cp437	5348	437
5348	iso_1	5348	819
5348	cp1252	5348	1252

This table matches DB2 CCSIDs with their appropriate Replication Agent Codepage and RS_ccsid configuration parameter values.

Comments

You can verify the CCSID setting for a database by executing the following query:

```
SELECT * FROM SYSIBM.SYSDATABASE WHERE NAME = 'database_name'
```

where *database_name* is the name of the database for which you are verifying the CCSID setting. You can also verify the CCSID settings for translation tables used by the LTM in character data conversions by executing the following query:

```
SELECT * FROM SYSIBM.SYSSTRINGS WHERE INCCSID = inccsid AND
OUTCCSID = outccsid
```

RSCsetname

The character set name used by the Replication Server to which Rep Agent connects.

Example syntax

```
RSCsetname=cp850
```

Default value

- iso_1

RS_source_db

The value of the RS_source_db parameter represents a logical database name for the tables being replicated. Sybase recommends using the DB2 subsystem or group name (for example, DSNA).

You must select a name to represent the DB2 database, and use it consistently to represent the DB2 database.

Make sure that value you supply for the RS_source_db parameter matches the value you specify for the *database* variable in the create connection command used to connect Replication Agent to the primary Replication Server.

The RS_source_db parameter is *required*.

Example syntax

```
RS_source_db=DSNA
```

Valid values

- The value of the RS_source_db parameter must match the primary database name in the Replication Server create connection command that LTM for MVS issues to connect the primary database to Replication Server.

Note Obtain your site's value for the RS_source_db parameter from the Replication Agent Installation Worksheet in the Replication Agent for DB2 UDB *Installation Guide* for z/OS. This value should have been recorded on the worksheet during installation.

RS_source_ds

Identifies the MVS system that contains the DB2 source database. You can choose this parameter value arbitrarily, as long as you use it consistently to represent the DB2 data server.

Be sure that the value you supply for the RS_source_ds parameter matches the value you specify for the *data_server* variable in the create connection command used to connect Replication Agent to the primary Replication Server.

The RS_source_ds parameter is *required*.

Example syntax

```
RS_source_ds=MVSA
```

Valid value

- The value of the RS_source_ds parameter must match the name of the primary data server in the Replication Server create connection command that LTM for MVS issues to connect the database containing the primary data to Replication Server.

- Maximum length: 30 characters.

Note Obtain your site's value for the RS_source_ds parameter from the Replication Agent Installation Worksheet in the Replication Agent for DB2 UDB *Installation Guide* for z/OS. This value was recorded on the worksheet during installation.

RS_pw

Provides the password for RS_user, which is the user ID that LTM for MVS uses to log into the primary Replication Server.

The RS_pw parameter is *required*.

Example syntax

```
RS_pw=ltmdevrs_pw
```

Valid values

- # (blank space)

  Setting this value to a blank space inserts a null password.
- Change this value for your site.
- Maximum length: 30 characters.

Note Obtain your site's value for the RS_pw parameter from the Replication Agent Installation Worksheet in the Replication Agent for DB2 UDB *Installation Guide* for z/OS. This value should have been recorded on the worksheet during installation.

RS_user

Identifies the Replication Server user ID that LTM for MVS uses to log in to the primary Replication Server. This user must have connect source permission on Replication Server.

The RS_user parameter is *required*.

Example syntax

```
RS_user=ltmdevrs
```

Valid value

- A valid Replication Server user ID
- Maximum length: 30 characters

Note Obtain your site's value for the RS_user parameter from the Replication Agent Installation Worksheet in the Replication Agent for DB2 UDB *Installation Guide* for z/OS. This value should have been recorded on the worksheet during installation.

RSHost

The DNS name of the platform that Replication Server is on. RSHost overrides the Replication Server hostname parameter in the SYGWXCPH module.

Example syntax

```
RSHost=perro
```

Valid values

- The name used must be a valid DNS name.

RSIPAddress

The RSIPAddress parameter overrides the Rep Server IP Address in the SYGWXCPH module.

In order to use this override parameter, the TCPName and RSPort parameters must also be specified.

The RSIPAddress parameter is not required.

The RSIPAddress and RSPort parameters must be commented out or deleted from the configuration file to use the entries in the SYGWXCPH module.

Example syntax

```
RSIPAddress=nnn.nnn.nnn.nnn
```

Default value

Defaults to the value of the IP Address specified for the RS machine name configuration value that has been encoded in the SYGWXCPH module.

Valid values

The series of 4 *nnn* strings must be replaced by values from 0 to 255 and the four numbers separated by periods must represent the IP Address of the machine specified in the RS configuration parameter. If `RSIPAddress=usehost`, RSHost is used to access Replication Server.

RSPort

The RSPort parameter overrides the Rep Server port number in the SYGWXCPH module.

In order to use this override parameter, the TCPName and RSIPAddress parameters must also be specified.

The RSPort parameter is not required.

The RSIPAddress and RSPort parameters must be commented out or deleted from the configuration file to use the entries in the SYGWXCPH module.

Example syntax

```
RSPort=nnnnn
```

Default value

Defaults to the value of the Rep Server port number specified for the RS machine name configuration value that has been encoded in the SYGWXCPH module.

Valid values

The port number on which the Rep Server specified in the RS configuration parameter is listening must replace the *nnnnn* string.

RSSDCsetname

The character set name used by the Replication Server to which Replication Agent connects.

The default value is iso_1. Use RSSDCsetname if the charset of Replication Server is not iso_1.

Example syntax

```
RSSDCsetname=cp850
```

Default value

- iso_1

RSSD_database

Provides the name of the database within the Adaptive Server that contains the Replication Server System Database (RSSD) for the primary Replication Server.

The RSSD_database parameter is *required* if you set the value of the Use_repdef parameter to Y.

LTM for MVS *ignores* the value of the RSSD_database parameter if you set the value of the Use_repdef parameter to N.

Example syntax

```
RSSD_database=rssd_name
```

Valid values

- The name of the Adaptive Server that contains the primary Replication Server's RSSD

Comments

You must set the value of the Use_repdef parameter to Y to use the RSSD_database parameter.

RSSDHost

The DNS name of the Adaptive Server that contains the primary Replication Server RSSD. This parameter overrides the RSSDServer hostname parameter in the SYGWXCPH module.

Example syntax

```
RSSDHost=perro
```

Valid values

- The name used must be a valid DNS name.

RSSD_pw

Provides the password for the Adaptive Server user ID specified in the RSSD_user parameter value.

The RSSD_pw parameter is *required* if you set the value of the Use_repdef parameter to Y.

LTM for MVS *ignores* the value of the RSSD_pw parameter if you set the value of the Use_repdef parameter to N.

Example syntax

```
RSSD_pw=password
```

Valid values

- Password of the Adaptive Server user ID specified in the RS_user parameter value
- # (blank space)

  Setting this value to a blank space inserts a null password.

RSSD_server

Provides the name of the Adaptive Server that contains the primary Replication Server's RSSD.

The RSSD_server parameter is *required* if you set the value of the Use_repdef parameter to Y.

LTM for MVS *ignores* the value of the RSSD_server parameter if you set the value of the Use_repdef parameter to N.

Example syntax

```
RSSD_server=sql_server_name
```

Valid value

- The value of the *SERVERNAME* variable in the SYGWHOST macro entry for the Adaptive Server that contains the primary Replication Server's RSSD.

Comments

If the value of the Use_repdef parameter is set to Y, you must specify a value for the RSSD_server parameter.

Note Obtain your site's value for the RSSD_server parameter from the Replication Agent Installation Worksheet in the Replication Agent for DB2 UDB *Installation Guide* for z/OS. This value should have been recorded on the worksheet during installation.

RSSD_user

Provides the identifier that Replication Agent uses to log on to the Adaptive Server that contains the primary Replication Server RSSD. This user must have permission to perform selects against the Adaptive Server.

The RSSD_user parameter is *required* if you set the value of the Use_repdef parameter to Y.

LTM for MVS *ignores* the value of the RSSD_user parameter if you set the value of the Use_repdef parameter to N.

Example syntax

```
RSSD_user=sql_server_user
```

Valid value

- The user ID of the Adaptive Server user who has permission to perform selects against the Adaptive Server RSSD.

RSSDIPAddress

The RSSDIPAddress parameter overrides the IP Address for the ASE server that has the RSSD database in the SYGWXCPH module.

In order to use this override parameter, the TCPName and RSSDPort parameters must also be specified.

The RSSDIPAddress parameter is not required.

LTM for MVS *ignores* the value of the RSSDIPAddress parameter if you set the value of the Use_repdef parameter to N.

The RSSDIPAddress and RSSDPort parameters must be commented out or deleted from the configuration file to use the entries in the SYGWXCPH module.

Example syntax

```
RSSDIPAddress=nnn.nnn.nnn.nnn
```

Default value

Defaults to the value of the IP Address specified for the RSSD machine name configuration value that has been encoded in the SYGWXCPH module.

Valid values

The series of 4 *nnn* strings must be replaced by values from 0 to 255 and the four numbers separated by periods must represent the IP Address of the machine specified in the RS configuration parameter.

If `RSSDIPAddress=usehost`, RSSDHost is used to access Adaptive Server.

RSSDPort

The RSSDPort parameter overrides the port number for ASE server that has the RSSD database in the SYGWXCPH module.

In order to use this override parameter, the TCPName and RSSDIPAddress parameters must also be specified.

The RSSDPort parameter is not required.

LTM for MVS *ignores* the value of the RSSDPort parameter if you set the value of the Use_repdef parameter to N.

The RSSDIPAddress and RSSDPort parameters must be commented out or deleted from the configuration file to use the entries in the SYGWXCPH module.

Example syntax

```
RSSDPort=nnnnn
```

Default value

Defaults to the value of the Rep Server port number specified for the RS machine name configuration value that has been encoded in the SYGWXCPH module.

Valid values

The port number on which the Rep Server specified in the RS configuration parameter is listening must replace the *nnnnn* string.

SetTruncPoint

The SetTruncPoint parameter determines how often the Log Extract requests a truncation point from the Rep Server.

The SetTruncPoint parameter is not required.

Example syntax

```
SetTruncPoint =99999
```

Default value

47000

Valid values

1 to 99999

Note A low SetTruncPoint value can cause extra CPU usage and network traffic.

Stop_on_error

Controls whether the detection of specific errors shuts down LTM for MVS.

The Stop_on_error parameter is *optional*.

Example syntax

```
Stop_on_error=N
```

Default value

- N

Valid values

- N

  LTM for MVS continues processing despite detection of specific errors. LTM for MVS logs errors in the *LTMLOG* file.

- Y

  - Shuts down LTM for MVS upon detection of specific errors.
  - LTL is sent to *LTLOUT* without passing data to Replication Server.

Comments

Use this parameter to troubleshoot specific errors, including invalid replication definitions. Set the value of this parameter to Y to assist in troubleshooting Replication Extract or a user exit application you created.

Table A-3 shows the behavior of LTM for MVS when it detects specific error conditions.

Table A-3: Error conditions that shut down LTM for MVS

Message number	Condition	How Stop_on_error value affects LTM for MVS
14021	Invalid login. **Note** This message appears only when using Replication Server 11.*x*.	LTM for MVS shuts down regardless of Stop_on_error value.
14025	LTM for RS_source_ds. RS_source_db is not configured to connect.	LTM for MVS shuts down regardless of Stop_on_error value.
14027	LTM for RS_source_ds. RS_source_db is already connected.	LTM for MVS shuts down regardless of Stop_on_error value. LTM for MVS retries the connection once.
14039	Log Transfer for LTM for RS_source_ds. RS_source_db is suspended. The connect source is refused because Log Transfer is suspended.	LTM for MVS shuts down regardless of Stop_on_error value.
20014	Invalid login.	LTM for MVS shuts down regardless of Stop_on_error value.
32020	Table *table_name* is not defined.	LTM for MVS does not shut down; retries connection regardless of Stop_on_error value.
32032	Replication definition does not exist for table *table_name*.	LTM for MVS shuts down only when Stop_on_error=Y.

Upon detecting other errors, LTM for MVS continuously attempts to retry the connection to Replication Server.

TableHandler

Specifies the program that processes the LTMOBJECTS table in DB2. By default, this program is LTMINFO. LTMINFO gets its information from the DB2 database. In a replication toolkit implementation of Replication Agent, you can substitute another program that builds the table from a data source other than DB2.

Example syntax

```
TableHandler=LTMINFO
```

Default value

- LTMINFO

TCPName

The TCPName parameter overrides the TCPName value in the *SYGWXCPH* module.

In order to use this override, the RSIPAddress and RSPort, or RSSDIPAddress and RSSDPort parameters must also be specified.

The TCPName parameter is not required.

Example syntax

```
TCPName=TCPIP
```

Default value

Defaults to the value of the TCPName specified for the RS machine name configuration value that has been encoded in the *SYGWXCPH* module.

Valid values

This name must represent the name of a TCPIP subsystem name that is currently active in the z/OS system when the Rep Agent is started.

Support_DB2_comp_rec

With the value of the Support_DB2_comp_rec parameter set to Y, Replication Agent suppresses DB2 delete compensation records from the transaction operation information sent to Replication Server.

Default value

- N

Valid values

- N

  LTM for MVS passes DB2 delete compensation records to Replication Server.

- Y

  LTM for MVS ignores DB2 delete compensation records. Delete compensation records are not passed to Replication Server.

Suppress_col_names

The Suppress_col_name parameter determines whether LTM for MVS suppresses column names from the LTL that is sent to Replication Server. You can use this parameter to reduce network traffic.

The Suppress_col_name parameter is *optional*.

Example syntax

```
Suppress_col_names=N
```

Default value

- N

Valid values

- N

Includes column names in the LTL that LTM for MVS sends to Replication Server.

- Y

  Suppresses column names from the LTL that LTM for MVS sends to Replication Server.

Warning! If you set the value of Suppress_col_names to Y, be sure that the order of the columns in your replication definition exactly matches the order of your DB2 columns. Failure to do so can result in a mismatch between your source and replicate data.

Comments

If you set the value of Suppress_col_names to Y, you must set the values of the following parameters as follows:

- Minimal_cols=Y
- Use_repdef=Y

Performance and tuning considerations

Setting the value of Suppress_col_names to Y, with Minimal_cols set to Y and Use_repdef set to Y, can dramatically increase throughput.

Time_conv_default

Performs two functions:

- Controls the value DateTime_conv_err supplied in columns that encounter date conversion errors when the DateTime_conv_err parameter is set to Datetime.
- Supplies time values during conversion of LTM for MVS date to Adaptive Server datetime format.

The Time_conv_default parameter is *optional*.

Example syntax

```
Time_conv_default=00:00:00
```

Default value

- 00:00:00

Valid values

- Any valid ISO time format value (*HH:MM:SS*, where *HH*=hour, *MM*=minute, and *SS*=second)

See "Working with Datatype Conversions" in the Replication Agent for DB2 UDB *User's and Troubleshooting Guide* for z/OS for examples of valid date, time, and timestamp formats.

Time_in_char

Controls whether LTM for MVS time columns are sent as char(8) fields or converted to Adaptive Server datetime format.

The Time_in_char parameter is *optional*.

Example syntax

```
Time_in_char=N
```

Default value

- N

Valid values

- N

  Causes LTM for MVS to convert LTM for MVS time columns to Adaptive Server datetime format.

- Y

  Causes LTM for MVS to send unmodified time columns as char(8) fields (format: *HH:MM:SS*) instead of converting them to Adaptive Server datetime format.

See Chapter 6, “Working with datatype conversions” in the Replication Agent for DB2 UDB *User’s and Troubleshooting Guide* for z/OS for examples of valid date, time, and timestamp formats.

Timestamp_in_char

Controls whether unmodified timestamp columns are sent as char(26) fields or converted to Adaptive Server datetime format.

The Timestamp_in_char parameter is *optional*.

Example syntax

```
Timestamp_in_char=N
```

Default value

- N

Valid values

- N

  LTM for MVS converts LTM for MVS timestamp columns to Adaptive Server datetime format. The datetime format contains only milliseconds, whereas timestamp contains microseconds.

- Y

  LTM for MVS sends timestamp columns as char(26) fields (example: 1994-01-01-10.02.30.006123) instead of being converted to Adaptive Server datetime format.

Comments

LTM for MVS timestamp data has 6 digits of precision. Set the value of Timestamp_in_char to Y to prevent precision loss during conversion of LTM for MVS timestamp data to Adaptive Server datetime format. Use when LTM for MVS timestamp data needs to be replicated and to remain unique.

See Chapter 6, “Working with datatype conversions” in the Replication Agent for DB2 UDB *User’s and Troubleshooting Guide* for z/OS for examples of valid date, time, and timestamp formats.

trace=Calls

Traces Replication Extract's calls to the Replication API and sends the output to the *SYSPRINT* file.

Warning! Run this trace only on the advice of Sybase Technical Support.

The trace=1,4 parameter is *optional*.

Example syntax

To enable this facility, insert the statement trace=Calls in the LTM for MVS configuration file.

Note Be sure to comment out trace statements in the configuration file after you obtain the desired information. Traces require significant MVS resources and can seriously impair LTM for MVS performance.

Comments

This trace may be dynamically enabled or disabled from the MVS operator console, using these commands:

```
F REPJOB,C,TRACE=Calls
F REPJOB,C,TRACEOFF=Calls
```

where `REPJOB` is the jobname of the Replication Agent.

See also

- Chapter 2, "Managing Replication Agent"

trace=LTLebcdic

Writes Log Transfer Language (LTL) that is passed to Replication Server to the *LTLOUT* trace file.

The trace=1,11 parameter differs from the trace=1,31 parameter in that it translates the *LTLOUT* file data into EBCDIC, while trace=1,31 translates the *LTLOUT* file data into ASCII.

The trace=1,11 parameter is *optional*.

Example syntax

To enable this trace facility, insert the statement trace=1,11 in the LTM for MVS configuration file.

Note Be sure to comment out trace statements in the configuration file after you obtain the desired information. Traces require significant MVS resources and can seriously impair LTM for MVS performance.

Comments

This trace may be dynamically enabled or disabled from the MVS operator console, using these commands:

```
F REPJOB,C,TRACE=LTLebcdoc
F REPJOB,C,TRACEOFF=LTLebcdic
```

where `REPJOB` is the jobname of the Replication Agent.

See also

- Chapter 2, “Managing Replication Agent”

trace=LTLascii

Starts a trace that sends data passed to Replication Server to the *LTLOUT* file. The user can then determine whether LTM for MVS is functioning properly by examining the *LTLOUT* file.

The trace=1,31 parameter differs from the trace=1,11 parameter in that it translates the *LTLOUT* file data into ASCII, while trace=1,11 translates the *LTLOUT* file data into EBCDIC.

The trace=LTLascii parameter is *optional*.

Example syntax

To enable this trace facility, insert the statement trace=1,31 in the LTM for MVS configuration file.

Comments

This trace may be dynamically enabled or disabled from the operator console, using these commands:

```
F REPJOB,C,TRACE=LTLascii
F REPJOB,C,TRACEOFF=LTLascii
```

where `REPJOB` is the jobname of the Replication Agent.

trace=QID,*n*

Traces the next *n* QIDs processed by Replication Extract. The output is sent to the *SYSPRINT* file. trace=QID is optional.

Warning! Run this trace only on the advice of Sybase Technical Support.

Example syntax

```
trace=QID,25
```

traces the first 25 QIDs processed. To enable this facility, insert the statement `trace=QID` into the LTM for MVS file.

Note Be sure to comment out trace statements in the configuration file after you get the information you need. Traces require significant MVS resources and can seriously impair LTM for MVS performance.

Comments

This trace can also be turned on and off dynamically from the MVS console.

Use_repdef

With TCP/IP connectivity, allows LTM for MVS to send LTL to Replication Server that contains only the columns specified in the replication definition.

The Use_repdef parameter is *optional*.

Example syntax

```
Use_repdef=N
```

Default value

- N

Valid values

- N

  LTM for MVS sends LTL to Replication Server that contains all DB2 columns defined in your primary table.

- Y

  With TCP/IP connectivity, LTM for MVS logs in to the Adaptive Server that contains the RSSD for the primary Replication Server. LTM for MVS then obtains the replication definition and includes only the columns specified in the replication definition in the LTL it sends to Replication Server.

  Note Because LTM for MVS logs in to the Adaptive Server that contains the RSSD for the primary Replication Server, the Adaptive Server user ID used by LTM for MVS must have permissions to perform selects against the Adaptive Server RSSD. This Adaptive Server user ID, specified by the value of the RSSD_user parameter, must have a minimum of SELECT privileges.

  If you set the value of Use_repdef to Y and use TCP/IP, you *must* do the following:

  - Set the value of Minimal_cols to Y.
  - Add a SYGWHOST macro entry for the Adaptive Server that contains the RSSD for the primary Replication Server. The *SYGWHOST* macro entry must contain valid values for this Adaptive Server in the *SERVERNAME*, *IPADDR*, and *LSTNPORT* parameters.

    Note As an alternative, these values can be specified in the RSSDIPAddress and RSSDPort parameters in the LTM configuration file.

- Provide valid values for each of the following configuration parameters:

  RSSD_database

  RSSD_pw

  RSSD_server

  RSSD_user

Performance and tuning considerations

If you specify only a subset of the tables in the replication definition, and the tables have long column names, setting the value of Use_repdef to Y can help reduce network traffic and can therefore improve performance when you use TCP/IP.

User_exit

Supplies the name of your user exit to Replication Agent upon start-up.

The User_exit parameter is *optional*.

Example syntax

```
User_exit=RADAEX1
```

Default value

- Null

Valid values

- The value of the User_exit parameter must be identical to the name of your user exit module and is restricted to the following:
  - Must be a maximum of 8 characters, all uppercase

  Note When processing the configuration file, Replication Agent automatically changes the case of the User_exit parameter value to upper case.

- Must begin with an alphabetic character
- Must be a valid module name on MVS
- Must be the name of the user exit module located in your *LINKLIB*

Comments

If you have a user exit, you must specify the name in the User_exit configuration parameter; otherwise the system will not load it.

APPENDIX B

Replication Extract Messages

This appendix lists and describes the messages issued by Replication Extract. Replication Extract displays these messages on the MVS operator console and writes them to the job log. Use this chapter to help understand and resolve error conditions.

If Replication Extract issues these or any other undocumented messages with the "LEX" prefix, contact Sybase Technical Support for assistance.

What you need to know

Message format	The messages in this chapter are listed in ascending order by message number. The message number format consists of the prefix "LEX," followed by a numeric message identifier and a message level.
Message text	The message text that follows the message number is in uppercase text. After each message is an explanation and, if appropriate, a suggested action to take.
Variables within messages	Message variables appear in lowercase italic text. Actual values replace the variables when the message displays. For example, in the following message, "*A bad switch value of switch_value*," the variable is represented by *switch_value*.
If you need help	If you need help from Sybase Technical Support when using this chapter, follow the instructions in "If you need help" on page xv.

See also

- Replication Agent for DB2 UDB *Installation Guide* for OS/390
- Replication Server documentation

Replication Extract LEX messages

This section lists Replication Extract messages.

Table B-1: Replication Extract messages

Message	Description	User action recommended
LEX0001 Invalid parameters passed to Sybase Log Extract.	Invalid parameters passed to the Sybase Log Extract. This is a Sybase Log Extract initialization error. The text of this message can vary. (See “What you need to know” on page 169 for information.)	Contact Sybase Technical Support for assistance.
LEX0002 Unable to load initialization module: *module*.	Sybase Log Extract is unable to load one or more initialization routines into memory.	See the job log for additional system error messages that describe the exact cause of the error.
LEX0003 Unable to obtain virtual storage in module: *module*.	The named module is unable to acquire necessary virtual storage.	Increase the region size and restart the Replication Agent started task. If the error persists, set the Replication Agent max_messages configuration parameter to a lower value to reduce the amount of storage dedicated to the Log Transfer Manager's LTL queue. If the error persists, call Technical Support.

Message	Description	User action recommended
LEX0004 LOAD Failed for module.	Sybase Log Extract is unable to link to or load the named module using LOAD. Sybase Log Extract uses LOAD program from the library specified in the STEPLIB JCL statement.	Verify that the STEPLIB contains the correct load library names. See the *Replication Agent for DB2 Installation Guide for MVS* for more information. If the STEPLIB contains the correct library names, see the job log for system error messages that describe the exact cause of the error. Also see the job log for additional system error messages that may describe the exact cause of the error.
LEX0005 DB2 SSID *subsystem_id* is not defined in the Sybase configuration file.	Sybase Log Extract cannot connect to the named DB2 subsystem because the subsystem is not defined with the Log_Identifier configuration parameter. This is a DB2 connection error. The text of this message can vary. (See "What you need to know" on page 169 for information.)	The configuration file must contain an entry for the subsystem specified by *subsystem_id* in the message. To add an SSID entry to configuration file see the *Replication Agent for DB2 Installation Guide for MVS*.
LEX0006 The Sybase Log Extract requires APF.	The Sybase Log Extract modules must execute in an authorized library.	See Chapter 5, "Installing Replication Agent," in the *Replication Agent for DB2 Installation Guide for MVS*, for information about APF authorization.
LEX0007 The Sybase LTMEXT Log Extract is exiting successfully.	The Sybase Log Extract has shut down normally.	No user action is necessary.
LEX0008 DB2 subsystem *subsystem_id* is not operational.	Sybase Log Extract cannot connect to the named DB2 subsystem using CONNECT because the subsystem is not running.	Start the DB2 subsystem and then restart the Replication Agent started task.

Message	Description	User action recommended
LEX0009 Plan name was not found in this DB2 subsystem.	Sybase Log Extract is unable to open a thread to DB2 using OPEN because the named plan is not bound on the DB2 subsystem. Sybase Log Extract retrieves its plan name from the DB2Plan configuration parameter.	Verify that the correct plan name is specified in the DB2Plan parameter. To specify the DB2Plan parameter, see the *Replication Agent for DB2 Installation Guide for MVS*. If the correct plan is specified, verify that the plan is bound on this DB2 subsystem. See the section about binding the SYBASE plan in the *Replication Agent for DB2 Installation Guide for MVS* for more information.
LEX0010 User *user* is not authorized to execute plan: *plan*.	Sybase Log Extract is unable to open a thread to DB2 using OPEN because the authorization ID associated with the Replication Agent started task does not have EXECUTE authority on the named plan.	Grant the user ID EXECUTE authority on the named plan.
LEX0011 Release level mismatch between DB2 and call attachment code.	Sybase Log Extract is unable to connect to DB2 using CONNECT because the Call Attachment Facility is not at the same release level as the DB2 subsystem. Sybase Log Extract used LOAD to load the Call Attach Facility from the library specified in the STEPLIBS JCL statement.	Verify that the STEPLIB or JOBLIB statement contains the correct DB2 load library names. See the *Replication Agent for DB2 Installation Guide for MVS* for more information.
LEX0012 DB2 connection failure - reason code = *reason_code*.	Sybase Log Extract lost its connection to DB2, which may be the result of a DB2 failure. The reason code is a DB2 reason code.	See the *IBM DB2 Messages and Codes* manual for a detailed explanation of this error code.
LEX0013 Missing log data detected, log processing terminated.	Sybase Log Extract terminated because it detected a gap in the DB2 log.	If you cannot resolve the problem, contact Sybase Technical Support.
LEX0014 SQL error detected in module *module*.	Sybase Log Extract encountered a SQL error. The SQLCODE that was returned and a description of the error follows this message.	See the *IBM DB2 Messages and Codes* manual for additional information regarding the SQLCODE and any action you should take to correct the problem.

Message	Description	User action recommended
LEX0015 User lacks read authority on active log dataset. Return code = *return_code*, reason code = *reason_code*.	The user ID associated with the Replication Agent started task must have READ authority on the DB2 active log data sets. The return and reason codes shown are source address field (SAF) error codes.	Use your installation security package (such as RACF) to grant READ access over your DB2 active log data sets to the user ID associated with the Replication Agent started task.
LEX0016 Replication stopped for table *table_namelog* LRSN: *lrsn* member: *member* URID: *urid* return code=*return_code*	Sybase Log Extract encountered an error while processing a DB2 log record that describes a change to the identified table. Replication of this table stops. The log LRSN indicates the LRSN of the log record that caused the error. Possible return codes and their meanings include the following: 4--The table is not defined on this DB2 subsystem. The log LRSN, URID, and member have no meaning for this error. The name of the table as specified in the *LTMOBJECTS* table is probably incorrect. 8--Sybase Log Extract is unable to expand DB2 compressed log data. 12--The DATA CAPTURE CHANGES clause is not activated for this table. 16--An error is encountered while column data is being converted from its internal DB2 format.	Do one of the following, depending on the return code you received: 4--Use LTMCSET (the Replication Agent interface panels) to delete the invalid row from the LTMOBJECTS table and to insert a new row with the correct table creator and name (see Chapter 4, "Managing Replication Agent"). Restart the Replication Agent started task. 8--Rematerialize the replicated tables before replication of the affected table can be restarted. 12--Rematerialize the replicated tables before restarting replication of the affected table. 16--Contact Sybase Technical Support. After you correct the error or re-materialize the replicated table in error, reset the ERROR_CODE column of the *LTMOBJECTS* table to 0 to restart replication of the affected table.

Message	Description	User action recommended
LEX0017 Log sequence error detected at RBA *rba*. Recovery processing in progress.	A log data sequence error is detected while the DB2 log is read. A gap exists in the DB2 log. Some data is missing from the first copy of the DB2 log. Sybase Log Extract terminates. Each Control Interval (CI) of the DB2 log contains control information, which includes the RBA of the CI. Sybase Log Extract uses this control information to verify that log records are being processed in the correct order. The log CI returned by a READ request contained an unexpected log RBA. This message does not indicate that an error occurred. It issues under any of the following conditions: The ARCHIVE LOG command issues, which results in the truncation of the log data set that Sybase Log Extract is processing. Sybase Log Extract recovers by switching to the next active log data set. DB2 overwrites the active log that Sybase Log Extract is processing. Sybase Log Extract repositions itself within the appropriate archive log data set and continues processing.	No user action is necessary.
LEX0018 *subsystem_id*. invalid Poll_Interval, format must be HH.MM.SS.TH.	An invalid value is specified on the Poll_Interval configuration parameter. Sybase Log Extract terminates.	Edit the configuration file. See the section about editing the file in the *Replication Agent for DB2 Installation Guide for MVS* for more information.

Message	Description	User action recommended
LEX0019 *subsystem_id*. invalid Poll_Interval value must be between 00 and 60.	An invalid value is specified for the minutes or seconds portion of the Poll_Interval configuration, which must be a number between 00 and 60. Sybase Log Extract terminates. This is a configuration error. The text of this message can vary. (See "What you need to know" on page 169 for information.)	Edit the configuration file. See the section about editing the file in the *Replication Agent for DB2 Installation Guide for MVS* for more information.
LEX0020 *subsystem_id*. CI_Max must be numeric.	An invalid value is specified with the CI_Max configuration parameter. The CI_Max value must be a number from 0 through 999. Sybase Log Extract terminates. This is a configuration file error. The text of this message can vary. (See "What you need to know" on page 169 for information.)	Edit the configuration file. See the section about editing the configuration file in "Configuring Sybase Log Extract" in the *Replication Agent for DB2 Installation Guide for MVS* for more information.
LEX0021 *subsystem_id*. GenID must be an integer from 0 to 65535.	An invalid value was specified for the GenID configuration parameter. The GenID value must be a number from 0 to 65535. Sybase Log Extract terminates. This is configuration file error. The text of this message can vary. (See "What you need to know" on page 169 for information.)	See the *Replication Agent for DB2 Installation Guide for MVS* for more information.
LEX0022 *subsystem_id*. SetTruncPoint must be an integer from 0 to 99999.	An invalid value is specified for the SetTruncPoint configuration parameter. The SetTruncPoint value must be a number from 0 to 99999. Sybase Log Extract terminates. This is a configuration file error. The text of this message can vary. (See "What you need to know" on page 169 for information.)	See the *Replication Agent for DB2 Installation Guide for MVS* for more information.
LEX0023 DB2 log extract initialization in progress.	Sybase Log Extract is attached and began initialization processing. Message LEX0040 issues when initialization completes and Sybase Log Extract begins extracting data from the DB2 logs.	No user action is necessary.

Message	Description	User action recommended
LEX0024 connection established to DB2 subsystem subsystem.	Sybase Log Extract successfully connected to the named DB2 subsystem.	No user action is necessary.
LEX0025 Log processing to begin at log RBA rba.	Sybase Log Extract begins scanning the log for changes to those tables marked for replication at the indicated log RBA or position within the DB2 log.	No user action is necessary.
LEX0026 Sybase Log Extract terminating with return code *return_code*.	The Sybase Log Extract subtask is terminating. If the return code shown has a non-zero value, then Sybase Log Extract terminates due to errors. Additional messages describing the cause of the error precede this message. Return code 0 indicates that Sybase Log Extract is terminating normally at the request of the operator or Replication Agent.	No user action is necessary unless you receive a non-zero return code. If you do, see the additional messages.
LEX0027 Sybase Log Extract unable to retrieve DB2 high written RBA.	Sybase Log Extract is unable to determine the end of the current active log. This information is needed to ensure that Sybase Log Extract does not read past the end of the current active log data set.	If you cannot resolve the problem, contact Sybase Technical Support.
LEX0028 Definition of table *table* not found.	The named table is not defined on this DB2 subsystem. The name of the table specified in the *LTMOBJECTS* table is probably incorrect. If you are running Sybase Log Extract with the Replication_LogExtractError=TERM option, Sybase Log Extract terminates.	Delete the invalid row from the *LTMOBJECTS* table and to insert a new row with the correct table creator and name (see Chapter 4, “Managing Replication Agent”). Restart the Replication Agent started task.
LEX0029 Unsupported release of DB2, Sybase Log Extract terminating.	Sybase Log Extract connected to a DB2 subsystem version 5.x or earlier, which is not supported. Sybase Log Extract supports only DB2 version 6.1 or later. Sybase Log Extract terminates.	No user action is necessary.

Message	Description	User action recommended
LEX0030 Log record sequence error detected. URID: *urid* LOGRBA: *rba* LRSN: *lrsn* member: *member*	Sybase Log Extract encountered a log record that is out of sequence.	If you cannot resolve the problem, contact Sybase Technical Support.
LEX0031 Capture data changes not active, unable to format log record. Table *table* RBA: *rba* log LRSN: *lrsn* member: *member*	The named table does not have the DB2 DATA CAPTURE CHANGES attribute activated. Sybase Log Extract cannot replicate changes made to a DB2 table without DATA CAPTURE CHANGES.	Do the following: Use the DB2 ALTER TABLE statement to turn on the DATA CAPTURE CHANGES attribute. For more information about ALTER TABLE, see the *IBM DB2 SQL Reference*. Rematerialize the affected replicated tables. Update the *LTMOBJECTS* row that describes the named table. You must reset the *ERROR_CODE* column of the *LTMOBJECTS* table to 0. Restart replication of the affected table.
LEX0032 Error attempting to expand compressed data at RBA *rba*	Sybase Log Extract is unable to expand DB2 compressed data contained in the indicated log record. This error occurs if Sybase Log Extract is unable to locate a usable DB2 expansion dictionary.	Rematerialize the affected replicated tables. If you cannot resolve the problem, contact Sybase Technical Support.
LEX0033 The definition of the *LTMOBJECTS* table is invalid.	Sybase Log Extract is unable to interpret a log record that describes a change to the *LTMOBJECTS* table, which indicates that the table is not defined correctly. The *LTMOBJECTS* table must be created using the data description language (DDL) shipped with Replication Agent. Do not edit the shipped DDL to rearrange the physical order of the columns or add or delete columns.	Using the DROP and CREATE commands, drop and re-create the *LTMOBJECTS* table with the correct definition before you resume replication.

Message	Description	User action recommended
LEX0034 Replication activated for table table. Log RBA: *rba* log time *date time* Log LRSN: *lrsn* member: *member*	Replication of the named table begins. Replication is dynamically activated for a table in one of two ways while Sybase Log Extract runs: By inserting a new row into the *LTMOBJECTS* table using INSERT. By resetting the ERROR_CODE column of an existing row in the *LTMOBJECTS* table to 0.	No user action is necessary.
LEX0035 Unable to start replication for table. The specified table does not exist.	The named table is not defined on this DB2 subsystem, which means the name of the table specified in the *LTMOBJECTS* table is probably incorrect. If you run Sybase Log Extract with the Replication_Error=TERM option, Sybase Log Extract terminates.	Delete the invalid row from the *LTMOBJECTS* table and insert a new row with the correct table creator and name (see "Managing Replication Agent" on page 67). Restart the Replication Agent started task.
LEX0036 Replication stopped for table *table*. Log RBA: *rba* log time *date time* log LRSN: member: *member*	Replication of the named table terminated because this table's entry is deleted from the *LTMOBJECTS* table with DELETE. This message does not indicate an error condition.	No user action is necessary.
LEX0037 Qualify status for table *table* has been changed to qualifier_name. Log RBA: *rba* log *time date* timelog LRSN: member: *member*	The qualify status is modified because Sybase Log Extract encountered a log record describing a change to the *QUALIFY* column of this table's *LTMOBJECTS* entry.	No user action is necessary.
LEX0038 Unable to change replication status of table. This table is not currently being replicated.	A change was requested to the replication status for a table that is not being replicated.	Bring Sybase Log Extract down. Check the data and indexes for the *LTMOBJECTS* table for accuracy. Then change the replication status.
LEX0039 Qualify status for table *table* has been changed to qualifier. Replicate_name is *replication_name*. Log RBA: *rba* log *time date* time Log LRSN: member: *member*	The value in the *QUALIFY* column of the *LTMOBJECTS* table has been changed. This value indicates whether the *CREATOR* value in the *LTMOBJECTS* table will pass to Replication Server.	No user action is necessary.
LEX0040 Log extract initialization complete.	Initialization of Sybase Log Extract is complete.	No user action is necessary.

Message	Description	User action recommended
LEX0041 Unable to initialize LTM communications.	Sybase Log Extract is unable to establish a communications link to LTM for MVS.	See the job log for additional messages describing the cause of the error.
LEX0042 Error detected on LTMRECV call.	Sybase Log Extract received an unexpected return code from the LTMRECV function. The return code received displays in message LEX0026.	If you cannot resolve the problem, contact Sybase Technical Support.
LEX0043 Error detected on LTMOPEN call.	Sybase Log Extract received an unexpected return code from the LTMOPEN function. The return code received displays in message LEX0026.	If you cannot resolve the problem, contact Sybase Technical Support.
LEX0044 Error detected on LTMSEND call.	Sybase Log Extract received an unexpected return code from the LTMSEND function. The return code received displays in message LEX0026.	If you cannot resolve the problem, contact Sybase Technical Support.
LEX0045 *Command* is not a valid command.	An invalid command is entered at the console.	See "Entering operator commands" on page 77.
LEX0046 Value command or parameter is too long.	An invalid command or parameter is entered at the console.	See "Entering operator commands" on page 77.

Message	Description	User action recommended
LEX0047 Error_code *return_code* detected in the replication registration entry for table *table*. replication will not be restarted for this table.	A non-zero ERROR_CODE value is detected in the *LTMOBJECTS* entry for the named table. Sybase Log Extract does not replicate changes made to this table. Possible ERROR_CODE values and their meanings are: 4--The table is not defined on this DB2 subsystem. The log RBA and URID have no meaning for this error. The name of the table as specified in the *LTMOBJECTS* table is probably incorrect. 8--Sybase Log Extract is unable to expand DB2 compressed log data. 12--The DATA CAPTURE CHANGES clause is not activated for this table. 16--An error is encountered while column data is being converted from its internal DB2 format.	Do one of the following, depending on the return code you received: 4--Use LTMCSET (in the Replication Agent interface panels) to delete the invalid row from the *LTMOBJECTS* table and to insert a new row with the correct table creator and name (see Chapter 4, "Managing Replication Agent"). Restart the Replication Agent started task. 8--Rematerialize the replicated tables before restarting replication of the affected table. 12--Rematerialize the replicated tables before restarting replication of the affected table. 16--Call Technical Support. After the error is corrected or the replicated table in error is re-materialized, reset the *ERROR_CODE* column of the *LTMOBJECTS table to* 0 to restart replication of the affected table.
LEX0048 DB2 has terminated. Sybase Log Extract terminating.	The DB2 subsystem to which Sybase Log Extract connected with CONNECT terminated. Sybase Log Extract cannot run while DB2 is down. Sybase Log Extract terminates.	To restart the Replication Agent started task, restart DB2.
LEX0049 Unable to format data change log record for table *table* at RBA *rba*, LRSN: lrsn member: *member*.	The DB2 data row described by a data change log record cannot be formatted using the column definitions retrieved from the DB2 catalog. This occurs if a primary table is dropped and re-created and Replication Agent then runs to extract changes made to the table before it was dropped. It may also occur if user exit rearranges the column order in LE Command.	Rematerialize the replicated copy of the table. If you cannot resolve the problem, contact Sybase Technical Support.

Message	Description	User action recommended
LEX0050 Replicate name for *table_name* is *rep_table_name*. Log RBA rba log time *date time* LRSN: member: *member*.	The replication of the named table is performed using the value of the *REPLICATE_NAME* column in the *LTMOBJECTS* entry for this table.	No user action is necessary.
LEX0051 Replicate name for table *table_name* has been changed to *new_name*. Log RBA rba log time *date time* LRSN: lrsn member: *member*.	The value in the *REPLICATE_NAME* column of the *LTMOBJECTS* entry for this table is updated. The *REPLICATE_NAME* is modified because the Sybase Log Extract encountered a log record that describes a change to the *REPLICATION_NAME* column of the *LTMOBJECTS* entry for this table. Replication of this table continues using the updated value.	No user action is necessary.
LEX0052 Replicate name for *table_name* Has been changed to default to the actual table name. Log RBA *rba* Log time *date time* LRSN: *lrsn* member: *member*.	The value of the *REPLICATION_NAME* of the named table is updated and set to a value of blank. Sybase Log Extract continues replication of the named table using the DB2 table name.	No user action is necessary.
LEX0053 Replication bypassed for table *table_name*. Log RBA *rba* log time *date time* LRSN: *lrsn* member: *member*.	The replication of the named table is bypassed. Sybase Log Extract encountered a log record that would start replication; however, the record is ignored due to a previous error with the named table.	After the error is corrected, use *LTMCSET* (in the Replication Agent interface panels) to delete the row from the *LTMOBJECTS* table and to insert the row back into the *LTMOBJECTS* table to activate replication for the named table (see Chapter 4, “Managing Replication Agent”).
LEX0054 A row has been deleted from the *LTMOBJECTS* table. The row represented *table_name* which was not currently being replicated. Log rba *rba* log time *date time* lrsn: *lrsn* member: *member*.	A table not being replicated is deleted from the *LTMOBJECTS* table.	No action is necessary.
LEX0055 Unable to retrieve Data Sharing member information.	Sybase Log Extract is unable to obtain information about the data sharing environment.	If you cannot resolve the problem, contact Sybase Technical Support.

Message	Description	User action recommended
LEX0056 An error occurred while converting a Constant default value for Column: *column* type *type*. The system default has been used.	Sybase Log Extract is unable to convert a column value that was defined with a constant default.	Check the definition of the column.
LEX0057 Switch to log *log data set name*. Begin RBA: *rba* processing type record from member: *member* Log RBA: *rba* Log LRSN: *lrsn*.	Sybase Log Extract completed reading a DB2 log and started reading a different log.	No action is necessary.
LEX0058 IFI start call failed.	Sybase Log Extract was unable to obtain the high LRSN value because the ID running Replication Agent does not have permission to execute DB2 MONITOR2 and TRACE commands. This is a Sybase Log Extract log reader error. The text of this message can vary. (See “What you need to know” on page 169 for information.)	Grant MONITOR2 and TRACE permission on DB2 to the LTMADMIN user ID.
LEX0059 Fetch for high LRSN failed.	Sybase Log Extract was unable to obtain the high LRSN value because the ID running Replication Agent does not have permission to execute DB2 MONITOR2 and TRACE commands. This is a Sybase Log Extract log reader error. The text of this message can vary. (See “What you need to know” on page 169 for information.)	Grant MONITOR2 and TRACE permission on DB2 to the LTMADMIN user. See Chapter 5, "Installing Replication Agent," in the *Replication Agent for DB2 Installation Guide for MVS*, for instructions on granting these permissions.
LEX0060 Variable text.	Sybase Log Extract encountered a problem reading the log. This is a Sybase Log Extract log reader error. The text of this message can vary. (See “What you need to know” on page 169 for information.)	Contact Sybase Technical Support.

Message	Description	User action recommended
LEX0061 Display group IFI call failed call attach error	A DB2 Display Group command was issued, but failed due to the call attach error that is displayed.	See the *IBM DB2 Messages and Codes* manual for additional information regarding this error and any action you should take to correct the problem. After the problem is corrected, shut down Sybase Log Extract and then restart.
LEX0062 QID: qid	This message displays during startup and shutdown of Sybase Log Extract.	No user action is necessary.
LEX0063 Tablespace *table* is a DB2 V5+ large object. This version of Sybase Log Extract does not support such objects. Data changes from this object may not be replicated accurately.	Sybase Log Extract does not support DB2 version 5 large objects in this version. Data for this table may not be replicated accurately.	No user action is necessary.
LEX0064 Error in reading the log not due to encountering the end of the log.	Sybase Log Extract cannot read the log file.	Contact Sybase Technical Support.
LEX0065 A connection could not be established to the *subsystem* DB2 subsystem.	The DB2 subsystem is not operating, or the Replication Agent is not configured to connect properly to the DB2 subsystem.	Verify that the configuration and authorization are correct and that the DB2 system is operational. Contact Sybase Technical Support.
LEX0066 Incorrect number of buffers specified.	The Buffers configuration parameter must have a value from 1 to 50.	Change the Buffers configuration parameter to have a value between 1 and 50.
LEX0067 Table name %0s not found in repdefs.”	A table was specified in the LTMOBJECTS system table but was not found in the Replication definitions.	Add a replication definition and modify the LTMOBJECTS table error code to enable replication.
LEX0068 Replication %0s for table %1s.%2s.”	Replication has stopped or started for the table described in message parameters 1 and 2.	If replication on the table has stopped, check other Rep Agent messages to determine why.
LEX0069 Column %0s changed to %1s”	A character column has changed in the LTMOBJECTS table.	If the Rep Agent changed this column, check other Rep Agent messages to determine why.
LEX0070 Column %0s changed to %1h”	The ERROR_CODE column has changed in the LTMOBJECTS table.	If the Rep Agent changed this column, check other Rep Agent messages to determine the cause of the error. Then take appropriate action.

Message	Description	User action recommended
71=LEX_071,” Log RBA: %0x Log Time: %1x-%2x-%3x %4x.%5x.%6x”	Informational only.	No action required.
72=LEX_072,” Log LSRN: %0x Member: %1s”	Informational only.	No action required.
73=LEX_073,”LEX0073 DB2 Authorization Error rc=%0p reason=%1p. Check user priv	Log Extract is missing a DB2 privilege.	Check user privileges.
74=LEX_074,”LEX0074 ECSA memory allocated,(SP,231), pointer=0%0p length=%1p.”	Informational only.	No action required.

APPENDIX C

LTM for MVS Messages

This appendix contains LTM for MVS error and information messages, LTMMGR messages, and suggests remedies for the error conditions they represent. Use the information in this appendix to help understand and resolve error conditions.

If you need help from Sybase Technical Support when using this chapter, follow the instructions in “If you need help” on page vi.

What you need to know

Message format	The messages in this chapter are listed in ascending order by message number. The numeric message identifier is followed by a message level.
Message text	The message text that follows the message number is in uppercase and lowercase text. After each message is an explanation and, if applicable, a suggested action to take.

Message parameters	Messages can have up to 10 parameters. The parameters are specified using a % sign, a digit from 0-9, and a datatype character. Since the parameters are numbered, they can be put anywhere in the message text, allowing the message to be localized (translated into different languages. Datatype characteres are not case sensitive. Valid datatype characters are: c a single character h a short integer d a long integer i a long integer s a string x a hex value p a pointer value
Variables within messages	Message variables appear in lowercase italic text. Actual values replace the variables when the message appears. For example, in the following message, "*A bad switch value of switch_value*," the variable is represented by *switch_value*.
Message levels	To help you determine the severity of a problem, a one-letter code at the end of the message identifier, such as the "W" at the end of the identifier "0000W" indicates the severity. The following are the meanings of each message code: • I: Information message • W: Warning message • E: Error message
Error and information errors	This section contains most of the error and information messages written to the Replication Agent error logs (LTMLOG.
Configuration parameters	Several of the error messages in this appendix refer to values for LTM for MVS configuration parameters.

See also

- IBM DB2 documentation for DB2 error information

LTMMGR messages

Table C-1 lists LTMMGR messages.

Table C-1: LTMMGR messages

Message	Description	User action recommended
01001 Error loading modules from *loadlib*	LTMMGR encountered an error trying to load.	Verify that the STEPLIB containing the Replication Agent load modules is correct. If it is, call Technical Support.
01002 Error loading localized message file.	LTMMGR encountered an error trying to build the *LTMLOC* file.	The format of the *LTMLOC* file is invalid. Browse the file to which the DD name *LTMLOC* points, which is a fixed-block file with a record length of 255. It contains message numbers and message text. *LTMLOC* is distributed on the installation CD and is transmitted to the mainframe. If *LTMLOC* is lost or corrupt, delete and re-create it by re-running the installation CD. If the problem persists, call Technical Support.
01003 LTM now logging to *LTMLOG*.	Information message.	None.
01004 Starting Ping to RS Server.	Information message.	None.
01005 Ping successful to Rep Server=XXXX.	Information message.	None.
01006 Ping unsuccessful to Rep Server=XXXX.	Unable to ping Rep Server defined in *LTMCFG*.	Check the Rep Server host/port values in the configuration file.
01007 Starting Ping to RSSD Server.	Information message.	None.
01008 Ping successful to RSSD Server=XXXX.	Information message.	None.
01009 Ping unsuccessful to RSSD Server=XXXX.	Unable to ping RSSD Server defined in *LTMCFG*.	Check the RSSD host/port values in the configuration file. Make sure that Replication Server is running.
12010 Retrying Connection to Server=XXXX.	LTMMGR tries again to connect to the Rep Server.	LTMMGR retries connecting to the server *Maximum_connect_retries* times as defined in the *LTMCFG* file.

Message	Description	User action recommended
12011 Connection to Server=XXXXX failed.	Unable to connect to Server.	Make sure Rep Server is running and is not in hibernation mode.
12012 Error Initializing the MVS Console communications.	Replication Agent encountered an error while initializing the MVS console communications.	Call Technical Support.
12013 *LTMCFG* configuration failed.	Replication Agent encountered errors in the configuration file.	Fix errors in configuration file.
12014 Successful connection to Rep Server %0s.	Information message.	None.

LTMOC messages

Table C-2 lists LTMOC messages.

Table C-2: LTMOC messages

Message	Description	User action recommended
02001 Starting LTM Open Client LTL connection.	Beginning connection process to Rep Server.	Informational message.
02002 Failed to allocate CS_CONTEXT in *cs_ctx_alloc()*.	LTMOC encountered an error calling *cs_ctx_alloc*.	Check region memory parameter and Open Client License key.
02003 *ct_init()* call failed.	LTMOC encountered an error calling *cs_init*.	Check region memory parameter and Open Client License key.
02004 Failed to allocate CS_CONNECTION in *ct_con_alloc()*.	LTMOC encountered an error calling *ct_con_alloc*.	Check region memory parameter and Open Client License key.
02005 Failed to set a property in the CS_CONNECTION structure. Property = '%0s',value = '%1d'.	LTMOC encountered an error calling *ct_con_props*.	Check region memory parameter and Open Client License key.
02006 *ct_diag* call failed.	LTMOC encountered an error calling *ct_diag*.	Check region memory parameter and Open Client License key.
02007 Connection to %0s '%1s' failed.	LTMOC encountered an error calling *ct_connect*.	Check region memory parameter and Open Client License key. Check *RS=*, *RS_user*, and *RS_pw* parameters and message 02008.
02008 severity: '%0d, message: '%1s'.	Message texre-runt from *ct_diag* call.	Message contains text of client and server messages from *ct_diag* calls.

Message	Description	User action recommended
02009 Failed to allocate a CS_COMMAND structure.	LTMOC encountered an error calling *ct_command_alloc*.	Check region memory parameter and Open Client License key.
02010 *ct_command* failed cmd='%0s'.	LTMOC encountered an error calling *ct_command*.	Check region memory parameter and Open Client License key.
02011 *ct_send* failed.	LTMOC encountered an error calling *ct_send*.	Check region memory parameter and Open Client License key.
02012 *ct_results* failed.	LTMOC encountered an error calling *ct_results*.	Check region memory parameter and Open Client License key.
02020 LTMOC SNDBUFR call failed.	LTMOC encountered an error sending data to the Rep Server.	Turn on tracing, re-run, and call Technical Support.
02021 LTMOC SNDASYNC call failed.	LTMOC encountered an error sending data to the Rep Server.	Turn on tracing, re-run, and call Technical Support.
02022 LTMOC Connect Source call failed, check *rs_source_ds*/ *rs_source_db*	LTMOC encountered an error sending a connect source command to Rep Server.	Check the *rs_source_ds* and *rs_source_db* parameters in the configuration file.
02023 LTMOC MAINTUSR call failed.	LTMOC encountered an error trying to get the Maintenance User from Rep Server.	Check to make sure Rep Server is running.
02024 LTMOC GETTRUNC call failed.	LTMOC encountered an error trying to get the truncpoint from Rep Server.	Check to make sure Rep Server is running.

LTMREPDF messages

Table C-3 lists LTMREPDF messages.

Table C-3: LTMREPDF messages

Message	Description	User action recommended
03001 LTM Open Client Replication Definition retrieval started.	Beginning Retrieval of Replication Definitions.	Informational message.
03002 Failed to allocate CS_CONTEXT in *cs_cxt_alloc()*.	LTMREPDF encountered an error calling *cs_ctx_alloc*.	Check region memory parameter and Open Client License key.
03003 *ct_init()* call failed.	LTMREPDF encountered an error calling *ct_init*.	Check region memory parameter and Open Client License key.
03004 Failed to allocate CS_CONNECTION in *ct_con_alloc()*.	LTMREPDF encountered an error calling *ct_con_alloc*.	Check region memory parameter and Open Client License key.

Message	Description	User action recommended
03005 Failed to set a property in the CS_CONNECTION structure. Property = '%0s', value = '%1d'.	LTMREPDF encountered an error calling *ct_con_props*.	Check region memory parameter and Open Client License key.
03006 *ct_diag* call failed.	LTMREPDF encountered an error calling *ct_diag*.	Check region memory parameter and Open Client License key.
03007 Connection to %0s '%1s' failed.	LTMREPDF encountered an error calling *ct_connect*.	Check region memory parameter and Open Client License key. Check *RS=*, *RS_user*, and *RS_pw* parameters and message 03008.
03008 severity: '%0d', message: '%1s'.	Message text from *ct_diag* call.	Message contains text of client and server messages from *ct_diag* calls.
03009 Failed to allocate a CS_COMMAND structure.	LTMREPDF encountered an error calling *ct_cmd_alloc*.	Check region memory parameter and Open Client License key.
03010 *ct_command* failed cmd='%0s'.	LTMREPDF encountered an error calling *ct_command*.	Check region memory parameter and Open Client License key.
03011 *ct_send* failed.	LTMREPDF encountered an error calling *ct_send*.	Check region memory parameter and Open Client License key.
03012 *ct_results* failed.	LTMREPDF encountered an error calling *ct_results*.	Check region memory parameter and Open Client License key.
03020 USEDB for RSSD database failed.	Database named in *RSSD_database* config parm not found.	Check *RSSD_database* config value.
03021 *use_repdef=y* and no repdefs in RSSD.	No repdefs were found in the RSSD database.	Either add repdefs to the RSSD or set *use_repdef=N*.
03022 LTMREPDF rep def query failed.	Unable to retrieve any repdefs.	Check Repdef definitions in RSSD database.
03023 LTMREPDF USEDB query failed, statement.	Unable to access RSSD database.	Check repdef database name in RSSD.
03024 Table Name >128 not supported.	Table name in Repdef is too large for DB2.	Change table name to be <128.
03025 Table Name = %0s.	Name of table in previous message.	None.
03026 Column Name >128 not supported.	Column name in Repdef is too large for DB2.	Change column name to be <128.
03027 Column Name = %0s	Name of column in previous message.	None.

LTMCFG messages

Table C-4 lists LTMCFG messages.

Table C-4: LTMCFG messages

Message	Description	User action recommended
04001 LTM Configuration started.	Information message.	None.
04002 Logging messages in file '%0s'.	Information message.	None.
04003 Reading '%0s' for configuration information.	Information message.	None.
04004 Configuration %0s was specified as %1s.	Information message.	None.
04005 LTM Configuration failed.	Information message.	Correct configuration errors.
04006 LTM Configuration successful.	Information message.	None.
04008 Parm '%0s' must be length 8 or less.	Information message.	Correct parm in error.
04009 Parm '%0s' must be specified.	Information message.	Specify parm listed in message.
04010 Syntax error for parm '%0s'.	Information message.	Fix syntax error on parm.
04011 Not numeric error on parm '%0s'.	Information message.	Fix non-numeric error on parm.
04012 Value must be Y or N for parm '%0s'.	Information message.	Set value of parm to Y or N.
04013 Invalid IP address for parm '%0s'.	Information message.	Change parm to valid IP address.
04014 *RSSD_server* not specified. Required if Use_repdef=Y.	Information message.	Specify *RSSD_server*.
04015 *RSSD_user* not specified. Required if Use_repdef=Y.	Information message.	Specify *RSSD_user*.
04016 *RSSD_pw* not specified. Required if Use_repdef=Y.	Information message.	Specify *RSSD_pw*.
04017 *RSSD_database* not specified. Required if Use_repdef=Y.	Information message.	Specify *RSSD_database*.
04018 When using Use_repdef=Y, *Minimal_cols* must be set to Y.	Information message.	Set *Minimal_cols* to Y or set *Use_repdef* to N.
04019 Invalid value specified for *Use_repdef*. Specify Y or N.	Information message.	Set *Use_repdef* to Y or N.
04020 RSSD IP Address invalid or missing, cannot set host properties.	Information message.	Enter a valid IP address for RSSD_IP _address.

Message	Description	User action recommended
04021 RSSD Port invalid or missing, cannot set host properties.	Information message.	Enter a valid port number for RSSD_port.
04022 Communications Protocol '%0s' is invalid. Use IBMTCP.	Information message.	Specify parm as IBMTCP.
04023 Invalid value specified for Suppress_col_names. Specify Y or N.	Information message.	Set Suppress_col_names to Y or N.
04024 Invalid value specified for *Support_DB2_comp_rec*. Specify Y or N.	Information message.	Set *Support_DB2_comp_rec* to Y or N.
04025 *Maximum_connect_retries* is invalid. Valid values are 0-9999.	Information message.	Enter a value for *Maximum_connect_retries* between 0 and 9999.
04026 Invalid value specified for *Minimum_year*. Valid values are four digits, 0000 - 9999.	Information message.	Enter a value for *Minimum_year* between 0000 and 9999.
04027 *Suppress_col_names*=Y is not valid with *Use_repdef*=N.	Information message.	Either change *Suppress_col_names* to N or change *Use_repdef* to Y.
04028 RS Port invalid or missing, cannot set host properties.	Information message.	Enter a valid number for *RS_port*.
04029 RS IP Address is invalid or missing, cannot set host properties.	Information message.	Enter a valid IP address for *RS_IPaddress*.
04030 TCPIP Address Name invalid or missing, cannot set host properties.	Information message.	Enter a valid TCPIP address space name.
04031 The data sharing option is MULTI, so the data sharing member will be ignored.	Information message.	*Datasharingmember* parm ignored.
04032 An RS Host and an IP address were specified, host will be used.	Information message.	*RS_host* will be used to connect to Rep Server.
04033 An RSSD Host and an IP address were specified, host will be used.	Information message.	*RSSD_host* will be used to connect to Rep Server.
04034 CCSID must be less than 65533.	CCSID value is out of range.	Correct CCSID value.
04035 *Datasharingoption* must be None, Single, or Multi.	Information message.	Change *Datasharingoption* to None, Single, or Multi.
04036 *Graphictype* must be char or bin.	Information message.	Enter char or bin for *Graphictype*.

Message	Description	User action recommended
04037 Invalid Trace value.	Information message.	Re-enter or modify the trace command to specify the correct value.

LTMAPI messages

Table C-5 lists LTMAPI messages.

Table C-5: LTMAPI messages

Message	Description	User action recommended
05001 '%0s' is being extracted by '%1s%2s'.		None.
05002 User exit module '%0s' detected and successfully loaded.	User exit loaded.	None.
05003 User exit module '%0s' not found in module library.	User exit not found.	Check Steplib DDs to make sure module is in one of the loadlibs.
05004 The API verb '%0s' received an invalid Globals pointer.	Internal error.	Call Technical Support.
05005 Unable to open the LTM Truncation point file: LTMTRUNC. Please correct the JCL and restart the LTM.	LTMTRUNC DD is incorrect.	Check DD LTMTRUNC for the correct DSN name. Then, correct the JCL and restart the LTM.
05006 Retrying connection to Server = %0s.	LTM is trying to reconnect to Rep Server.	None.
05007 Connection to Server=%0s failed.	Unable to access Rep Server.	None.
05008 LTLSIZE=%0d too big for buffer=%1d.	LTL too big for packet size.	Increase packet size parameter.
05009 Load failed for Module =%OS.	A module is missing from the install libraries.	Call Technical Support.
05014 Successful connection to Rep Server %0s.	Information message.	None.
05021 The control cmd is invalid.	The control console command was invalid.	Enter the correct syntax.

LTMINFO messages

Table C-6 lists LTMINFO messages.

Table C-6: LTMINFO messages

Message	Description	User action recommended
INF0001 TABLE LOADING: %0s.	Informational message.	No action required.
INF0002 Processing table: %0s.	Informational message.	No action required.
INF0003 Replication table %0s has Error Code %1d.	Informational message.	No action required.
INF0004 Repdef column not in DB2 table.	Repdef column does not correspond to a DB2 column.	Check Repdef and DB2 tables.
INF0005 DB2 table has no repdefs: %0s.	DB2 table does not have repdefs.	Add repdefs to Adaptive Server to replicate from table.
INF0006 DB2 Table has repdefs: %0s.	Informational message.	No action required.
INF0007 DB2 column has repdef: %0s.	Informational message.	No action required.
INF0008 This column will not be replicated: %0s.	Informational message.	No action required.
INF0009 OBJUPDT FOR: %0s.	Informational message.	No action required.
INF0010 OBJUPDT SET LEERRCD: %0d.	Informational message.	No action required.
INF0011 OBJUPDT PREPARE: %0d.	Informational message.	No action required.
INF0012 OBJUPDT UPDATE for: %0s. Errcode: %1d.	Informational message.	No action required.
INF0013 Replication stopped for %0s ErrorCode=%1d.	Informational message.	No action required.
INF0014 Replication started for %0s.	Informational message.	No action required.
INF0015 Number of rows retrieved from LTMOBJECTS %0d.	Informational message.	No action required.
INF0016 LTM Object not found: %0s.	Informational message.	No action required.
INF0017 TABLE LOAD started.	Informational message.	No action required.
INF0018 TABLE LOAD ended.	Informational message.	No action required.
INF0019 Unable to open LTMOBJECTS.	LTMOBJECTS table not found.	Check Creator and DB2 parameters.

APPENDIX D

MVS Console Messages

This appendix lists the Replication Agent error and information messages that display on the MVS console. It also contains suggested remedies for the error conditions these represent. Use these messages to help resolve error conditions.

If you need help from Sybase Technical Support when using this chapter, follow the instructions in "If you need help" on page xv.

What you need to know

Message format	The messages in this chapter are listed in ascending order by message number. The message number format usually consists of the prefix "LTM," followed by a numeric message identifier and a message level.
Message text	The message text that follows the message number is in uppercase and lowercase text. After each message is an explanation and, if appropriate, a suggested action to take.
Variables within messages	Message variables appear in lowercase italic text. Actual values replace the variables when the message appears. For example, in the message, "A bad switch value of *switch_value*," *switch_value* is the variable.

Message levels

To help you determine the severity of a problem, a one-letter code at the end of the message identifier, such as the "W" at the end of the identifier "LTM0000W" indicates the severity.

The following are the meanings of each message code:

- I: Information message
- W: Warning message
- E: Error message

See MVS documentation for further information on MVS console messages.

LTM for MVS messages displayed on the MVS console

Table D-1 lists LTM for MVS messages that appear on the MVS console.

Table D-1: LTM for MVS messages on the MVS console

Message	Description	User action recommended
CSV003: REQUESTED MODULE *module_name* NOT FOUND.	The requested module is not loaded.	No user action is necessary.
LTMRAM0002: Replication Extract subtask started.	The subtask that reads the primary database file started.	No user action is necessary.
LTM0003: Log Transfer Interface subtask started.	The subtask that communicates with Replication Server started.	No user action is necessary.
LTM0004: Valid configuration parameter accepted.	A valid configuration command was entered on the console.	No user action is necessary.
LTM0006: Replication Extract task ended unexpectedly.	The subtask that reads the log file ended prematurely.	Review the LTM for MVS error logs (LTMLOG and LTMLOG2) and the MVS console log to find the reason for this termination (based on messages reported by Replication Extract). Correct the error and restart Replication Agent.

Message	Description	User action recommended
LTM0009: Log Transfer task ended. Retrying connection.	The connection to Replication Server terminated. Replication Agent attempts to re-establish the connection. **Note** The LTM for MVS does not re-establish the connection until there is data replication or enough DB2 activity to cause the Replication Extract to request a truncation point.	Check the LTM for MVS error logs (LTMLOG and LTMLOG2), the Net-Gateway error log, and the Replication Server log to find the reason the connection terminated. • If the communications network is down temporarily, Replication Agent restarts automatically when communications are re-established. • If the problem is not in network communications, correct the error. Depending on the type of error, shut down Replication Agent and restart it when the error is corrected.
LTM0010: Configuration parameter cannot be modified.	The configuration parameter entered at the console cannot be modified while Replication Agent runs.	To change a configuration parameter value, shut down Replication Agent, change the configuration parameter value, and restart Replication Agent. See Appendix A, “LTM for MVS Configuration Parameters.”
LTM0011: That command not valid until connection established.	Replication Agent cannot process the command entered on the console until the connection to Replication Server is established. Replication Agent tries to establish communications with Replication Server until it succeeds. Replication Agent only accepts the shutdown command while it tries to establish communications with Replication Server.	No user action is necessary.

Message	Description	User action recommended
LTM00012: LTM now logging to *LTM_error_log*.	*LTM_error_log* refers to one of the LTM for MVS error logs, LTMLOG or LTMLOG2. This message appears at these times: • When LTM for MVS starts. • When the current LTM for MVS error log (LTMLOG or LTMLOG2) fills, which causes Replication Agent to start sending log entries to the other log file.	No user action is necessary. **Note** Sybase recommends that you back up each log file to ensure that any errors or warnings written to the log are not lost or overwritten when the log switches.
LTM00013: Log Transfer Interface subtask restarted.	The connection to Replication Server is re-established and Replication Agent tries to replicate.	No user action is necessary.
LTM0014: RECEIVED MESSAGE FROM CONSOLE: *Operator_Message*.	*Operator_Message* is the first 30 characters of the command entered at the console.	No user action is necessary.
LTM0015: LTM IS PROCESSING SHUTDOWN	Replication Agent received a command to shut down, ended the subtasks, and shuts down. This message does not appear when Replication Agent shuts down unexpectedly.	No user action is necessary.
LTM0016: LTM alarm thread subtask started.	The subtask that flushes the buffer during periods of low activity started.	No user action is necessary.
LTM0017: STORAGE SUB-POOL *xx* ALLOCATED	*xx* represents the storage sub-pool being allocated.	No user action is necessary.
LTM0018: USE_REPDEF=Y. PROCESSING REPDEFS.	This message reports that the value of the Use_repdef configuration parameter is Y.	No user action is necessary. Change the value of this parameter if desired.
LTM04038: Invalid license key	The license key in the configuration file is in valid.	Correct the license key in the configuration file or obtain a new license key.
LTM04039: Missing license key	The license key is missing in the configuration file.	Obtain and enter a valid license key in the configuration file.

APPENDIX E

Running Multiple Replication Agents

This appendix describes how to run multiple Replication Agents in your replication system.

Running multiple Log Transfer Managers

You can run multiple LTM for MVS instances in your replication system to obtain the following:

- Improved performance of your replication system if you are replicating large volumes of transaction information
- Access to multiple DB2 subsystems
- Access to multiple DB2 data-sharing groups

For each LTM for MVS instance in your replication system, you must have uniquely named versions of some, but not all, data sets. The following tables shows which data sets must be unique, and which can be unique or shared:

Table E-1: Unique data sets required for each LTM for MVS

Data set description	Data set name
LTM for MVS configuration data set	*hlq.JCL(LTMCNFG)*
Truncation point data set	*hlq.ssid.TRUNCPT*
RA message data set	*hlq.ssid.LTMLOG*

Table E-2: Data sets required for each LTM for MVS, shared or unique

Data set description	Data set name
RA DB2 load library	*hlq.RA.LINKLIB*
DB2 table specifying which tables to replicate	*creator.LTMOBJECTS*

JCLLIB (LTMCNFG)

The following parameters in the *LTMCNFG* data set should be reviewed:

- *RS*
- *RS_user*
- *RS_pw*
- *RSSD_database*
- *RSSD_user*
- *RSSD_pw*
- *RS_source_ds*
- *RS_source_db*
- *Log_identifier*
- *creator*
- *LTMPlan*
- *GenID*

The following parameters are optional:

- *RSAddress*
- *RSPort*

- *TCPName*
- *RSSDIPAddress*
- *RSSDPort*
- *SharingName*
- *SharingType*

RA.LINKLIB

This Replication Agent load library contains all the modules needed for the Replication Agent to execute.

Each Replication Agent should have its own *LTMCNFG* file. The *RSIPAddress*, *RSPort*, *RSSDIPAddress*, *RSSDPort*, and *TCPName* parameters in the *LTMconfiguration* file must point to the appropriate Replication Servers and SQL servers for each Rep Agent.

creator.LTMOBJECTS

In DB2, the table *LTMOBJECTS* contains the specifications for all the tables to be replicated by a RA DB2. You can use the same *creator.LTMOBJECTS* for multiple RA DB2s if the replication specification is the same. If the specification differs, create another *LTMOBJECTS* with a different DB2 owner, and specify the new owner in the *creator* parameter in the *hlq.JCL (LTMCNFG)*. The DB2 table must be named *LTMOBJECTS*.

APPENDIX F

Creating User Exits

User exits are transaction programs that are written to take control at a determined point in a program.

Warning! Running user exits can compromise the integrity of Replication Agent. Sybase Technical Support does not assist with designing, coding, or testing your user exit application.

Understanding user exits

Replication Agent allows you to write custom user exits, which are programs that can alter transaction operations before LTM for z/OS forwards the operations to Replication Server. Replication Agent supports user exits written in z/OS/370 Assembler language.

You can create user exit applications to alter transaction operations in the following ways:

- Filter transaction operations
- Change replicate table or column names
- Add columns to transaction operations
- Drop columns from transaction operations

- Change column datatypes
- Write messages to the LTM for MVS log
- Validate or customize date range values, data values, or formats
- Shut down LTM for MVS

You can find a sample user exit in the *RADAEX1* member of the *hlq.DOCS* data set. You can adapt this sample application to create your own exit routine that performs transaction operation manipulations.

See "Using the sample user exit" on page 221 for information about the contents of the sample user exit.

Understanding applications for user exits

Table F-1 describes situations for which you can write a user exit, methods of programming for those situations, alternatives to programming a user exit, and considerations for programming a user exit.

Table F-1: User exit uses, programming methods, and alternatives

In this situation...	What you can do	For user exit details, see
The replicate table name differs from the source table name.	*With a user exit*: • Change the physical table name by supplying a new replicate table name in the value of LEPHYSTB (44 characters maximum). If the first byte of LEPHYSTB contains binary zeroes (x'00'), then the next 4 bytes can contain a pointer to a larger table name (45 - 128 bytes). The table name data at that pointer must have a two byte length followed by the name string. **Note** Always check the maximum length of tables names in the Replication Server and target data base documentation to ensure that those servers will accept large table names. *Alternate methods:* • Change the value of the REPLICATE_NAME column in the LTMOBJECTS table, if appropriate, to optimize performance. • Change the replicate table name in the replication definition (if you are using Replication Server 11.5 or later) .	*Table F-5: LE Command structure contents*

In this situation...	What you can do	For user exit details, see
The replicate column names differ from primary record or segment names.	*With a user exit*: • Change column names by setting values for the @COL_NAME and @COL_NAME_LENGTH attributes. If the first three bytes of @COL_NAME contain binary zeroes (x'000000'), then the next 4 bytes can contain a pointer to a larger table name (19 - 128 bytes). The column name at that pointer must have a two byte length, followed by the name string. **Note** Always check the maximum length of table names in the Replication Server and target database documentation to ensure that those servers will accept large table names. *Alternate method:* • Create Replication Server function strings. • Change the replicate column name in the replication definition (if you are using Replication Server 11.5 or later).	*Table F-7: COL structure contents*
The replicate definition does not contain all of the columns in the primary record or segment.	*With a user exit*: Specify to not replicate a specific column by setting the value of the @COL_TYPE attribute to @COL_DONOT_REPL. *Alternate method:* Set the value of the Use_repdef LTM for z/OS configuration parameter to Y to reduce network traffic and increase throughput.	*Table F-7: COL structure contents*

In this situation...	What you can do	For user exit details, see
The datatype of your primary data is incompatible with the replicate data.	*With a user exit*: • Change the value of the @COL_TYPE attribute of the COL message structure to convert datatypes as needed. For example, you can preserve all digits in timestamp columns by doing the following: 1 Define the columns as char(26) in the replication definition and in the replicate table. 2 Set the value of the @COL_TYPE attribute to @COL_TYP_CHAR. 3 Convert timestamp data to char representation. *Alternate method:* • Use the date and time conversion configuration parameters to convert Replication Agent-supported date, time, and timestamp data to char representations. • Create Replication Server function strings.	*Table F-7: COL structure contents*
The replicate column values differ from primary record or segment values.	*With a user exit*: • Change the area pointed to by the @RPTCOL_REDO and @RPTCOL_UNDO attributes of the @RPTCOL message structure • Change the @COL_LENGTH value as appropriate. *Alternate method:* • Create Replication Server function strings.	*Table F-6: @RPTCOL structure contents*
Columns must be added to the primary transaction operation to match the replicate table.	*With a user exit*: • Add @RPTCOL nodes to the end of the @RPTCOL list in the LE Command structure. *Alternate method:* • Create Replication Server function strings.	*Table F-6: @RPTCOL structure contents*
Custom message(s) must be logged by Replication Agent.	*With a user exit*: • Place a pointer to a list of message nodes in the @APLOGMSG attribute of the user exit communications area.	*Table F-3: User exit communications area structure*

In this situation...	What you can do	For user exit details, see
LTM for MVS *must be shut down.*	*With a user exit*: • Either set Register 15 to return a code other than 0, or set the value of the *APFATAL* attribute of the user exit communications area to Y.	• *Table F-2: User exit register contents* • *Table F-3: User exit communications area structure*
The transaction operation must not be replicated.	*With a user exit:* • Set the value of the *APREPSW* attribute of the user exit communications area to N to reduce network traffic (especially when where clauses in your subscriptions filter a large number of transaction operations). *Alternate method:* • Create Replication Server function strings.	*Table F-3: User exit communications area structure*
DB2 GRAPHIC and VARGRAPHIC datatypes must be replicated.	*With a user exit:* • Define the columns as binary to ASE.	

What you need to know

User exit memory management	The exit routine must not free any space passed through the parameter list. The user exit manages all memory it allocates. Sybase recommends that you use the provided work area (the APWKAREA attribute of the user exit communications area), instead of allocating additional memory.
What to avoid when using a user exit	Avoid performing physical input and output operations. These operations significantly impair Replication Agent performance. Do not allocate storage that you do not maintain.
User exits and Replication Server	Avoid circumventing Replication Server by using a user exit to replicate into another database on z/OS. This task involves issues beyond the scope of Replication Agent.

Considerations for working with the LE Command structure (LE_CMD)	Do not remove nodes from the LE Command structure. Do not rearrange the sequence of columns defined by the sequence of @RPTCOL nodes in the LE Command structure. The user exit does not change the value of the @LECOL pointer in the LE_CMD node. The user exit does not change the value of the @RPTCOL_NEXT pointer in the @RPTCOL nodes, *except* when you use a user exit to add columns. In this case, the exit changes the value of the @RPTCOL_NEXT pointer from null to a valid address of the added column in the last @RPTCOL node in the @RPTCOL list.

See also

- "LE_CMD structure" on page 212
- "@RPTCOL structure" on page 213
- "COL structure" on page 214

Understanding user exit communications

LTM for MVS communicates with a user exit application through the user exit communications area, which is described in this section, and by using the LE Command structure, described in detail on 211.

General purpose register contents

Table F-2 shows the contents of the general purpose registers at the time the user exit invokes.

Table F-2: User exit register contents

User Exit Register	Contents
1	A pointer to a list of parameter addresses: • The first address points to the LE Command structure (see "LE_CMD structure" on page 212). The LE Command structure defines the transaction operations that you want to send to Replication Server. • The second address points to the user exit communications area (see Table F-3 on page 210).
13	Address of the caller's save area.
14	Return address.
15	User exit entry point address. At exit, register 15 contains your user exit application's return code. You can specify that your user exit return any value. If it returns any value other than 0, LTM for MVS issues return code 20 to Replication Extract and shuts down with appropriate messages.

User exit communications area structure

The user exit communications area contains two nodes:

- The user exit communications area node
- The message node

Table F-3 shows the structure of the user exit communications area.

Table F-3: User exit communications area structure

Attribute	Length (bytes)	Valid values	Description
APREPSW	1	• Y (default) Allows replication of the transaction operation. • N Prevents replication of the transaction operation.	Indicates whether the transaction operation should be replicated.

Attribute	Length (bytes)	Valid values	Description
APFATAL	1	• Y Shuts down LTM for MVS (sends a return code of 20 to Replication Extract). • N (default) LTM for MVS continues operating.	Indicates whether LTM for MVS should issue a return code of 20 to Replication Extract.
APFUTURE	8	Not applicable	Reserved for future use.
APLOGMSG	4	• x'00' (null pointer) • Valid address of a message node	Points to a linked list of messages to be logged (see *Table F-4: Message node structure*).
APWKAREA	4000	Not applicable	This is a static work area for your user exit application. When the user exit places data in this area, the data is retained across multiple calls to the user exit.

Table F-4 shows the structure of the message nodes to which APLOGMSG can point.

Table F-4: Message node structure

Attribute	Length (bytes)	Description
APMSGNXT	4	A pointer to the next message structure or null.
APMSGTXT	4	A pointer to a null-terminated message string. (The null terminator is x'00'.)

Understanding the LE Command structure

When you install a user exit, the user exit intercepts the LE Command (LE_CMD) structure and changes its content. The LE Command structure defines the transaction operations that you want to send to Replication Server.

The LE Command structure points to the *@RPTCOL* structure; the *@RPTCOL* structure then points to the COL structure.

The *LECMDEXT* member of the *hlq.MACLIB* data set contains sample LE Command, *@RPTCOL*, and COL structures.

The following sections show the contents of these structures.

LE_CMD structure

Table F-5 shows the contents of the LE Command structure.

Table F-5: LE Command structure contents

Attribute	Length (Bytes)	Contents
LEORID	4	The database server site ID. This information is not sent to Replication Server, so no restriction exists for its contents.
LEORDB	31	The database name.
LEORQID	36	The Replication Server origin queue ID, consisting of: • LEORGYR: The origin year (4 bytes). • LEORGMON: The origin month (2 bytes). • LEORGDAY: The origin day (2 bytes). • LEORHOUR: The origin hour (2 bytes). • LEORMIN: The origin minute (2 bytes). • LEORSEC: The origin seconds (2 bytes).
LESTATUS	4	A full word, which is a binary value indicating the transaction operation type. Valid values are: • 8 = LEINSERT • 16 = LEDELETE • 36 = Log error message
LECMD	4	A full word, which is a binary value indicating the operation type. Valid values are: • 111 = LECOMMIT • 112 = LEROLLBACK • 120 = LEAPPLYD **Note** The LEAPPLYD subcommand implies that the value of the redo pointer is set; the undo pointer value is also set for an update. The other LE Command subcommands set the values of these pointers to x'00' (null).
LECOL	4	The address for the first @RPTCOL structure.
LEPHYSTB	45	The name associated with the object of the transaction operation (not null-terminated). The first space marks the end of the value.

Attribute	Length (Bytes)	Contents
LESBCS	8	Number indicating a single-byte code page.
LEDBCS	8	Number indicating a double-byte code page.
LEMIXED	8	Number indicating mixed code page.
LEUSERID	8	The user ID associated with this transaction operation.
LEXACTLE	8	Length of the transaction ID.
LEXACTID	8	Address of the transaction ID.
LECREATR	8	DB2 CREATOR ID.
LEQUAL	1	Indicates whether tablename is qualified with CREATOR ID.
LEPLANNM	8	DB2 plan name.
LECORRID	12	DB2 correlation ID.
LEUTILITY	8	DB2 connection ID.

@RPTCOL structure

Table F-6 shows the contents of the @RPTCOL structure.

Table F-6: @RPTCOL structure contents

Attribute	Length (Bytes)	Contents
@RPTCOL_NEXT	4	A pointer to the next @RPTCOL structure. Set the value of this node to null if it is the last column.
@RPTCOL_COL	4	A pointer to the column description structure (COL).
@RPTCOL_REDO	4	A pointer to the after image of the data. Set the value of this node to -1 (*0xFFFFFFF*) if the column has a null value.
@RPTCOL_UNDO	4	A pointer to the before image of the data. Set the value of this node to -1 (*0xFFFFFFFF*) if the column has a null value, or to 0 (null) for inserts and deletes.
@RPTCOL_REDO_LENGTH	2	The binary length of the redo data.
@RPTCOL_UNDO_LENGTH	2	The binary length of the undo data.
LECRSV4	2	Reserved.

COL structure

Table F-7 shows the contents of the COL structure. The first five full words are reserved for use by the Replication Extract.

Table F-7: COL structure contents

Attribute	Length (Bytes)	Contents
@COL	20	Reserved.
@COL_LENGTH	2	The length of the column as defined in the table in binary.
@COL_COLNO	2	The column sequence within the table in binary. The first column number must be 1.
@COL_SCALE	2	The number of decimal places in a numeric field containing a decimal in binary.
@COL_KEYSEQ	2	The sequence of this column in the table key represented in binary. The value is null (*X'00'*) if it is not part of the key (reserved).
@COL_NAME_LENGTH	4	The binary length of the column name.
@COL_NAME	18	The name of the column, which is not null-terminated.

Attribute	Length (Bytes)	Contents
@COL_TYPE	1	The type of the column. Valid values are: • COL_TYPE_CHAR: Character data. • COL_TYPE_VARCHAR: Character data. • COL_TYPE_DATE: This date is in the *YYYY-MM-DD* format (where *YYYY* is the year, *MM* is the month, and *DD* is the day of the month). • COL_TYPE_DECIMAL: Packed decimal field. • COL_TYPE_DONOT_REPL: The name of the column that you do not want to replicate. This value does not appear in the LTL. • COL_TYPE_FLOAT: 4-byte or 8-byte float. • COL_TYPE_GRAPHIC: Treated as binary (non-quoted). • COL_TYPE_INTEGER: Signed binary full word. • COL_TYPE_SMALLINT: Signed Binary half-word. • COL_TYPE_TIME: This time is in the *HH.MM.SS.TTTTTT* format (where *HH* are the hours, *MM* are the minutes, and *SS* are the seconds). • COL_TYPE_TIMESTMP: This timestamp is in the *YYYY-MM-DD-HH.MM.SS.xxxxxx* format (where *YYYY* is the year, *MM* is the month, and *DD* is the day of the month, *HH* are the hours, *MM* are the minutes, *SS* are the seconds, and *xxxxxx* are fractions of seconds). • COL_TYPE_VARG: Treated as binary (non-quoted). • COL_TYPE_DISTINCT: User-defined datatype. • COL_TYPE_TINYINT: 1-byte integer (unsigned). • COL_TYPE_DATETIME: The time is in the *MM-DD-YYYY HH*:*MM*:*SS*:*TTT* format. This encompasses the full range of Sybase datetime values. See documentation for Adaptive Server Enterprise for further details. • COL_TYPE_SMALLDATETIME: The time is in the *MM-DD-YYYY HH*:*MM*:*SS*:*TTT* format. This encompasses a restricted range of Sybase datetime values. See documentation for Adaptive Server Enterprise for further details. • COL_TYPE_BINARY: Variable-length binary field.
@COL_FLAG	1	Valid values for column flags are: • @COL_DEFAULT: x'0000 0010', NOT NULL WITH DEFAULT. • @COL_GRAPHIC: x'0100 0000', the column contains graphic data. • @COL_NULLS: x'1000 0000', the column can contain nulls. • @COL_VAR: x'0000 0001', the column length is variable.
@COL_MIXED		Indicates column contains mixed data.
@COL_SINGLE		Indicates column contains single-byte data.

Attribute	Length (Bytes)	Contents
@COL_FORBIT		Indicates column contains binary data.

LE Command structure illustrations

The following four figures show the LE Command structure for updates, inserts, deletes, and messages; and for a begin, commit, or rollback.

Figure F-1: LE Command structure for an update

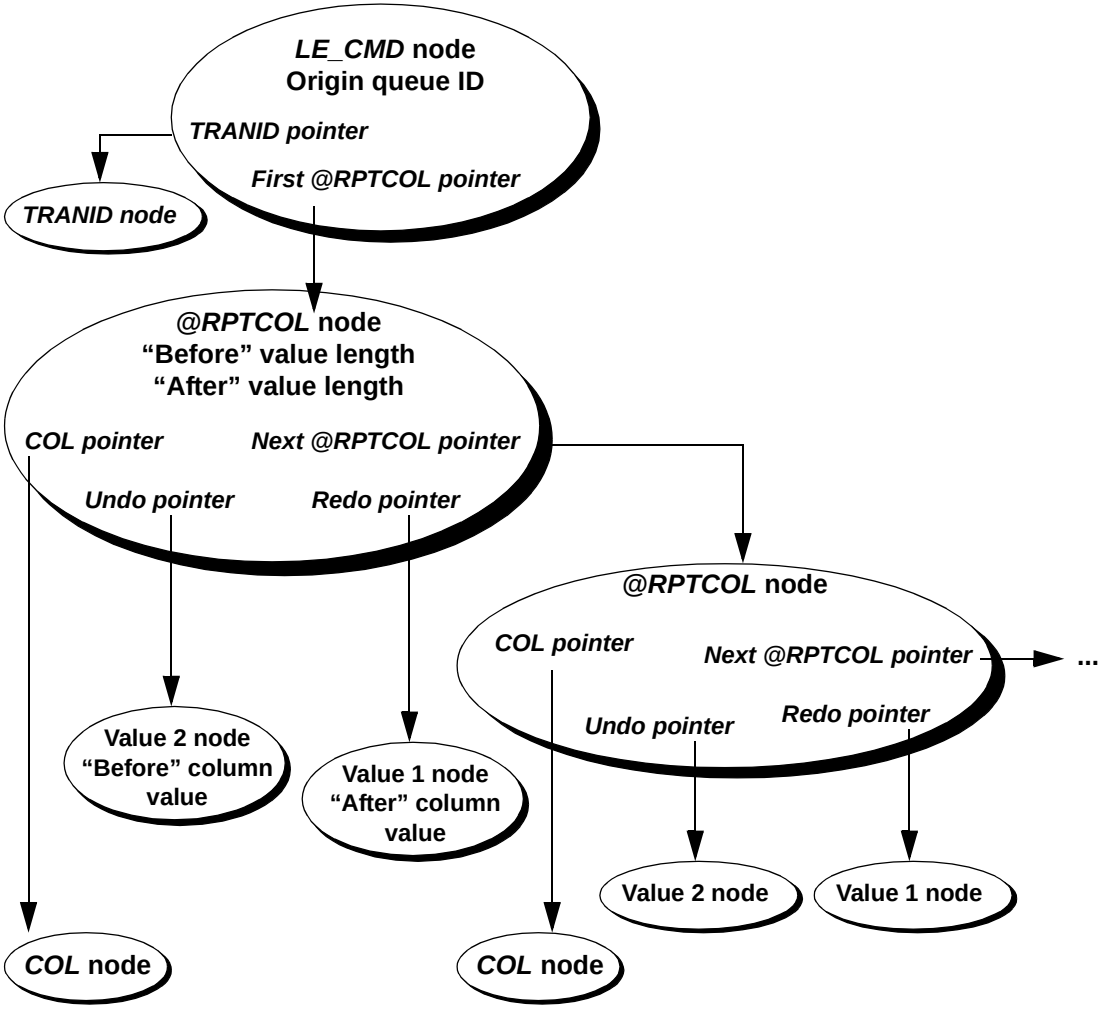


Figure F-2: LE Command structure for an insert or delete

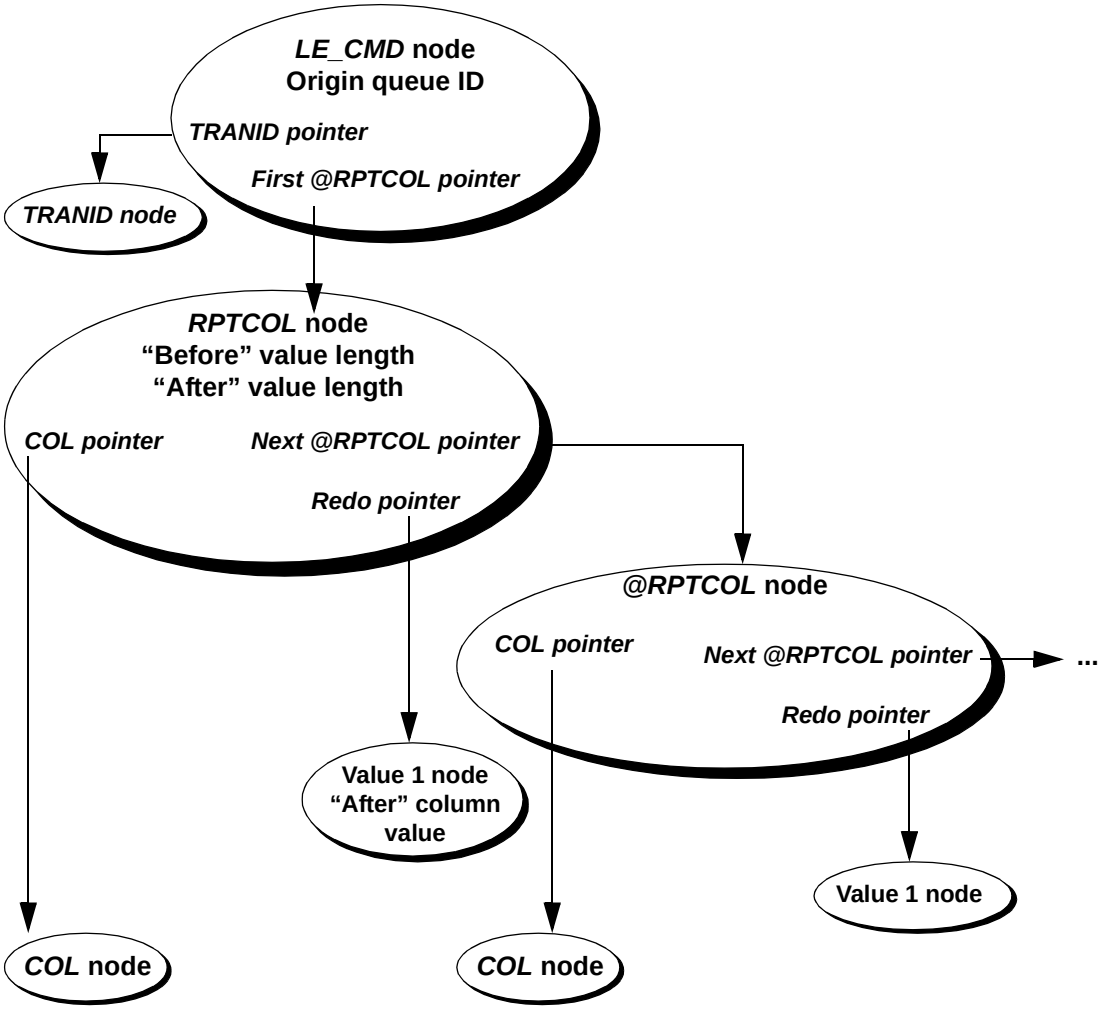
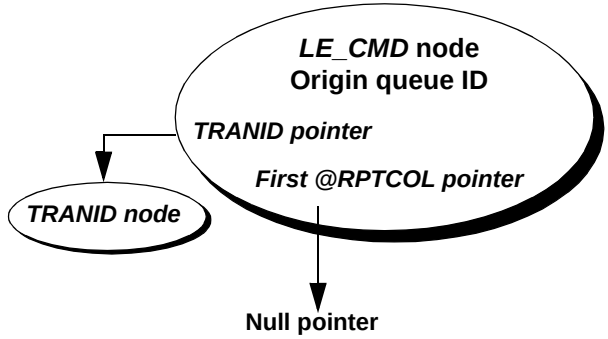
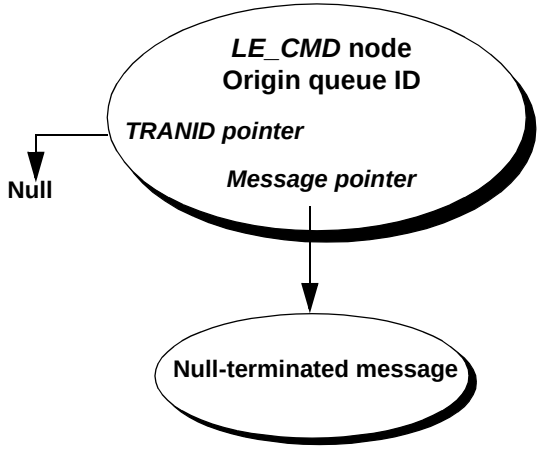


Figure F-3: LE Command structure for a begin, commit, or rollback



The LE Command structure for a message is shown in the following figure:

Figure F-4: LE Command structure for a message



This is the structure of a message to be sent to the log.

Running and removing user exits

To alter primary transaction operations marked for replication, you can create, install, and run a custom user exit procedure, or you can run the sample user exit. Remove your user exit when you no longer want to run it.

Installing user exits

This section contains instructions for installing and running a user exit.

To install and run a user exit application:

1 Place your application load module in the data set where the Replication Agent *hlq..LINKLIB* resides.

2 Edit the LTM for z/OS configuration file to specify the name of the user exit as the value of the User_exit configuration parameter.

The sample configuration file is located in the *LTMCNFG* member of the *hlq.JCL* data set.

3 Stop LTM for z/OS using the instructions in Chapter 4, “Managing Replication Agent.”

4 Start LTM for z/OS using the instructions in Chapter 4, “Managing Replication Agent.”

5 Verify that the user exit installed successfully by checking the LTM for z/OS log file for messages about the user exit.

Removing user exits

This section contains instructions for removing a user exit.

To remove the user exit application:

1 Edit the LTM for z/OS configuration file to comment out the User_exit configuration parameter. (The sample configuration file is located in the *LTMCNFG* member of the *hlq.JCL* data set.)

2 Stop LTM for z/OS using the instructions in Chapter 4, “Managing Replication Agent.”

3 Start LTM for z/OS using the instructions in Chapter 4, “Managing Replication Agent.”

4 Verify that the user exit did not load by checking the LTM for z/OS log file for messages about the user exit.

Using the sample user exit

Replication Agent supplies a sample user exit, located in the *RADAEX1* member of the *hlq.DOCS* data set.

The sample user exit contains routines that provide examples for the following operations on all tables and columns in transaction operations specified for replication:

- Excluding transaction operations from replication
- Excluding columns from replication
- Formatting columns to include dashes (- characters)
- Changing column datatypes
- Changing table names
- Converting column names to lower case
- Increasing column name lengths to up to 30 characters

Warning! If you run the sample user exit, *RADAEX1*, without alterations, the exit changes all column names in specified replicate tables to lowercase. This can result in a mismatch between the source column names and the column names specified in the replication definitions.

Migration considerations affecting user exits

The user exit should be recompiled with each release of Replication Agent for DB2 UDB for OS/390 to ensure compatibility with the current structure.

Testing your user exit

This section explains methods of testing your user exit applications, including using LTM for MVS configuration parameters, creating a sample table for testing, and generating sample LTL. These sections also help you evaluate your user exit by showing sample contents of the JES JOB log and the Replication Agent log with and without a user exit installed.

Configuring LTM for MVS for testing user exits

You can set the values of two LTM for MVS configuration parameters to help you evaluate and test the user exit application you create.

To test your user exit using the LTL_test_only configuration parameter:

1 Set the value of the LTL_test_only configuration parameter to Y.

LTM for MVS then sends all LTL to the LTLOUT DD statement for transactions that it would otherwise pass to Replication Server.

2 View the contents of the LTLOUT file to verify whether your user exit is altering transaction operations as desired.

To test your user exit using the API_com_test configuration parameter:

1 Set the value of the API_com_test configuration parameter to Y or O so that you can receive output to troubleshoot your user exit, if necessary.

2 View the contents of the *APICOM* and *UELECMD* files to examine the LE Command structure before and after invoking the user exit.

See Appendix A, “LTM for MVS Configuration Parameters,” for more information about the LTL_test_only and API_com_test configuration parameters.

Creating a user exit testing table

The following example syntax shows sample data definition language (DDL) you can use to create a table for testing a user exit.

Example

```
CREATE TABLE DWMA08.AUTHOR
                (AU_SSN      CHAR(09)              NOT NULL,
                 AU_LNAME    VARCHAR(40)           NOT NULL,
                 AU_FNAME    VARCHAR(20)           NOT NULL,
                 PHONE       CHAR(12)             NOT NULL,
                 ADDRESS     VARCHAR(12),
                 CITY        VARCHAR(20),
                 STATE       CHAR(02),
                 COUNTRY     VARCHAR(12),
                 POSTALCODE  CHAR(14),
                 BIRTHDATE   DATE,
                 NUM_UPDATES SMALLINT,
                 ORIG_INSERT TIMESTAMP,
                 LAST_UPDATE TIMESTAMP
  IN DATABASE DWMDBAS;
  COMMIT;
```

Generating sample log transfer language

The following example shows data manipulation language (DML) you can use to generate sample LTL for testing a user exit.

Example

```
-- TRANSACTION 1
INSERT INTO DWMA08.AUTHOR
VALUES('123456789', 'FUDD', 'ELMER', '5043333333', 'FIRST STREET',
       'BAKER', 'CO', 'USA', '70714', '1943-04-25',
       0, CURRENT TIMESTAMP, CURRENT TIMESTAMP);

INSERT INTO DWMA08.AUTHOR
VALUES('987654321', 'PIG', 'PORKEY', '3037777777', 'SOME STREET',
       'DENVER', 'CO', 'USA', '80013', '1932-06-12',
       0, CURRENT TIMESTAMP, CURRENT TIMESTAMP);

COMMIT; -- TRANSACTION 2
INSERT INTO DWMA08.AUTHOR
VALUES('564738291', 'BUNNY', 'BUGS', '3035551212', 'OTHER STREET',
       'PARIS', 'FR', 'FRN', '8293879283', '1932-07-04',
       0, CURRENT TIMESTAMP, CURRENT TIMESTAMP);
```

```
COMMIT;

-- TRANSACTION 3
INSERT INTO DWMA08.AUTHOR
VALUES('192837465', 'DUCK', 'DAFFY', '3031215555', 'SOME STREET',
       'LONDON', 'UK', 'UKN', '3473476545', '1932-12-03',
       0, CURRENT TIMESTAMP, CURRENT TIMESTAMP);

UPDATE DWMA08.AUTHOR
SET STATE = 'LA',
    NUM_UPDATES = NUM_UPDATES + 1,
    LAST_UPDATE = CURRENT TIMESTAMP
WHERE AU_SSN = '123456789';

UPDATE DWMA08.AUTHOR
SET COUNTRY = 'USA',
    NUM_UPDATES = NUM_UPDATES + 1,
    LAST_UPDATE = CURRENT TIMESTAMP
WHERE AU_SSN = '564738291';

COMMIT;

-- TRANSACTION 4
UPDATE DWMA08.AUTHOR
SET COUNTRY = 'FRN',
    NUM_UPDATES = NUM_UPDATES + 1,

LAST_UPDATE = CURRENT TIMESTAMP
WHERE AU_SSN = '192837465';

COMMIT;

-- TRANSACTION 5
UPDATE DWMA08.AUTHOR
SET COUNTRY = 'FRN',
    NUM_UPDATES = NUM_UPDATES + 1,
    LAST_UPDATE = CURRENT TIMESTAMP
WHERE AU_SSN = '987654321';

DELETE
FROM DWMA08.AUTHOR
WHERE AU_SSN IN ('123456789', '564738291');

COMMIT;
```

Understanding LTLOUT file contents

The contents of the *LTLOUT* file show the Log Transfer Language records that would normally be sent to Replication Server.

LTLOUT file contents with the user exit installed

The following example shows sample contents of the *LTLOUT* file when the sample user exit is installed and the sample DML (from "Generating sample log transfer language" on page 223) executes. The LTL shows the first transaction from the sample DML.

Example

```
_ds 1 ~";Oct 12 1998 04:34:16:000pm,4 0x0038B1316C7CF6D100000001B1316C7CF6C20
0010000236D22B0000000000000,6 0xC4E6D4F50000236D22B0B1316C7CF6C200010000B1316
C7CF6BF0A01 _bg tran _ds 1 ~";Oct 12 1998 04:34:16:000pm,4 0x0038B1316C7CF730
00000001B1316C7CF6C200010000236D22B0010000000000,6 0xC4E6D4F50000236D22B0B131
6C7CF6C200010000B1316C7CF6BF0A01_ap~"(AUTHORS.~|*rs_delete_yd_bf~$&au_id=~",2
3456789,~$) au_lname=~"$FUD,~$)au_fname=~"&ELMER,~$&phone=~"+5043333333,~$(ad
dress=~"FIRST STREET,~$%city=~"&BAKER,~$&state=~"#LA,~$(country=~"$USA,~$+pos
talcode=~"&70714,~$*birthdate=~"404-25-
194307:47:32,~$,num_updates=1,~$,orig_insert=~";1998-10-12
15.57.01.241423,~$)sourcedb=~"%DWM5 _ds 1 ~";Oct 12 1998 04:34:16:000pm,4
0x0038B1316C7CF74D00000001B1316C7CF6C200010000236D22B0010000000000,6
0xC4E6D4F50000236D22B0B1316C7CF6C200010000B1316C7CF6BF0A01 _ap
~"(AUTHORS.~|*rs_delete _yd _bf ~$&au_id=~",564-73-
8291,~$)au_lname=~"&BUNNY,~$)au_fname=~"%BUGS,~$&phone=~"+3035551212,~$(addre
ss=~"-OTHER
STREET,~$%city=~"&PARIS,~$&state=~"#FR,~$(country=~"$USA,~$+postalcode=~"+829
3879283,~$*birthdate=~"407-04-1932
07:47:32,~$,num_updates=1,~$,orig_insert=~";1998-10-12-
15.57.01.306722,~$)sourcedb=~"%DWM5 _ds 1 ~";Oct 12 1998 04:34:16:000pm,4
0x0038B1316C7CFCE300000001B1316C7CF6C200010000236D22B0000000000000,6
0xC4E6D4F50000236D22B0B1316C7CF6C200010000B1316C7CF6BF0A01 _cm tran
```

LTLOUT file contents without the user exit installed

The following example shows sample contents of the *LTLOUT* file when the sample user exit is not installed and the sample DML (from "Generating sample log transfer language" on page 223) executes. The LTL shows the first transaction from the sample DML.

Example

```
_ds 1 ~";Nov 05 1999 07:02:18:000pm,4
0x0002B31AB050CBE900000001B31AB050A4C900010000981A1722000000000000,6
0xC4E6D4F50000981A1722B31AB050A4C900010000B31AB050A4C63802 _cm tran _ds 1
~";Nov 05 1999 07:0 6:43:000pm,4
0x0002B31AB14D60E700000001B31AB14D60C700010000981A567F000000000000,6
0xC4E6D4F50000981A 567FB31AB14D60C700010000B31AB14D60C47600 _bg tran _ds 1
~";Nov 05 1999 07:06:43:000pm,4 0x0002B31AB1
4D62AC00000001B31AB14D60C700010000981A567F000000000000,6
0xC4E6D4F50000981A567FB31AB14D60C700010000B 31AB14D60C47600 rollback
transaction
```

Understanding the JES JOB log contents

The following shows a sample log message for the status of the user exit module:

```
11.22.32 JOB18936  CSV003I REQUESTED MODULE RADAEX1  NOT
FOUND
```

This message does not appear when the user exit installs successfully.

Understanding Replication Agent log contents

This section shows sample contents of the Replication Agent log.

Log contents with the user exit installed

The following example shows sample contents of the Replication Agent log when a user exit installs successfully, including log messages about the status of the user exit module:

Example

```
I. Mon Jul 29 11:45:26 1996. User exit module detected and successfully loaded.
```

The following example shows sample contents of the Replication Agent log when the sample DML shown in “Generating sample log transfer language” on page 223 executes.

Example

```
I. Mon Jul 29 11:46:02 1996. Message from the user exit: This transaction
operation owned by ID DWMA08
I. Mon Jul 29 11:46:02 1996. Message from the user exit: TranOp summary: 1, 0, 0.
I. Mon Jul 29 11:46:02 1996. Message from the user exit: This transaction
operation owned by ID DWMA08
I. Mon Jul 29 11:46:02 1996. Message from the user exit: TranOp summary: 2, 0, 0.
I. Mon Jul 29 11:46:02 1996. Message from the user exit: This transaction
operation owned by ID DWMA08
I. Mon Jul 29 11:46:02 1996. Message from the user exit: TranOp summary: 3, 0, 0.
I. Mon Jul 29 11:46:02 1996. Message from the user exit: AUTHOR table: Non-USA
transaction operations dropped:         1
I. Mon Jul 29 11:46:02 1996. Message from the user exit: This transaction
operation owned by ID DWMA08
I. Mon Jul 29 11:46:02 1996. Message from the user exit: TranOp summary: 4, 0 0.
I. Mon Jul 29 11:46:02 1996. Message from the user exit: AUTHOR table: Non-USA
transaction operations dropped:         2
I. Mon Jul 29 11:46:02 1996. Message from the user exit: This transaction
operation owned by ID DWMA08
I. Mon Jul 29 11:46:02 1996. Message from the user exit: TranOp summary: 4, 1, 0.
I. Mon Jul 29 11:46:02 1996. Message from the user exit: This transaction
operation owned by ID DWMA08
I. Mon Jul 29 11:46:02 1996. Message from the user exit: TranOp summary: 4, 2, 0.
I. Mon Jul 29 11:46:02 1996. Message from the user exit: This transaction
operation owned by ID DWMA08
I. Mon Jul 29 11:46:02 1996. Message from the user exit: TranOp summary: 4, 3, 0.
I. Mon Jul 29 11:46:02 1996. Message from the user exit: AUTHOR table: Non-USA
transaction operations dropped:         3
I. Mon Jul 29 11:46:02 1996. Message from the user exit: This transaction
operation owned by ID DWMA08
I. Mon Jul 29 11:46:02 1996. Message from the user exit: TranOp summary:
4, 4, 0.
I. Mon Jul 29 11:46:02 1996. Message from the user exit: This
transaction operation owned by ID DWMA08
I. Mon Jul 29 11:46:02 1996. Message from the user exit: TranOp summary: 4, 4, 1.
I. Mon Jul 29 11:46:02 1996. Message from the user exit: This transaction
operation owned by ID DWMA08
I. Mon Jul 29 11:46:02 1996. Message from the user exit: TranOp summary: 4, 4, 2.
```

Log contents without the user exit installed

The following shows a sample message that appears in the Replication Agent log when a user exit you specify is not installed:

LTM05003: User exit module *user_exit_name* **not found in**

```
module library.
```

If you intended a user exit to be loaded, verify that the name of your user exit module is identical to the value of the User_exit configuration parameter. Correct any discrepancies and restart Replication Agent.

Note When you do not specify a user exit in the User_exit configuration parameter, Replication Agent prints message 11050 to the LTMLOG file: "No LE exit module detected."

Glossary

This glossary includes terms and definitions either used or paraphrased from the following sources:

- The *IBM Dictionary of Computing*
- The *American National Dictionary for Information Systems*
- The *Information Technology Vocabulary*, developed by Subcommittee 1, Joint Technical Committee 1, of the International Organization for Standardization and the International Electrotechnical Commission (ISO/IEC JTC1/SC1). Sybase also used definitions from draft international standards, committee drafts, and working papers being developed by ISO/IEC JTC1/SC1.

This glossary uses the following references:

- *Contrast with* refers to a term that has an opposite or different meaning.
- *Compare with* refers to a term that has a similar meaning.
- *See also* refers to terms that have a related meaning.

Words containing these references are highlighted.

active log

The part of the DB2 log where log records are written as they are generated. The active log always contains the most recent log records. The archive log holds older records that no longer fit on the active log. See also **Database 2**. Contrast with **archive log.**

Adaptive Server Enterprise

The server in the Sybase Client-Server architecture. It manages multiple databases and multiple users, tracks the actual location of data on disks, maintains mapping of logical data description to physical data storage, and maintains data and procedure caches in memory.

address

An identifying number (often hexadecimal or binary) that describes a location in computer memory where information is stored.

after image

The complete contents of a row after an update or insert. See also **image**. Contrast with **before image**.

APF See **authorized program facility**.

API See **application program interface**.

application program interface (API) A predefined functional interface, supplied by an operating system or other licensed program, that allows an application program written in a high-level language to use specific data or functions of the operating system or the licensed program.

Replication Agent for DB2 includes the Replication API, an open API to which application developers can write Replication Extract programs. See also **Log Transfer Manager for z/OS**, Replication API, and Replication Extract.

archive log The part of the DB2 log that contains log records that no longer fit in the active log. See also **Database 2**. Contrast with **active log**.

ASE See **Adaptive Server Enterprise**.

atomic materialization The process of populating a replicate database using Replication Server commands that prevents client applications from executing transactions against the primary database while the subscription data is unloaded.

authorization The process of granting a user either complete or restricted permission to access an object, resource, or function.

authorized program facility (APF) A facility that recognizes the programs authorized to use restricted functions.

automatic materialization The process of populating a replicate database using Replication Server commands. After the replicate database is populated with a copy of the primary data, this command automatically activates your transaction replication subscriptions. Replication Server supports automatic materialization only from primary databases to which it has a connection (through a gateway, if necessary) and that can process SQL.

BCP See **bulk copy transfer**.

before image The complete contents of a row before an update or delete. See also **image.** Contrast with **after image**.

bootstrap data set (BSDS) A VSAM data set that contains name and status information for DB2 and RBA range specifications for all active and archive log data sets. See also **relative byte address** and **Virtual Storage Access Method**.

BSAM Basic Sequence Access Method.

BSDS See **bootstrap data set**.

bulk copy transfer (BCP) A transfer method wherein multiple rows of data are inserted into a table in the target database.

bulk materialization A method of initializing subscription data outside the replication system. This is also a process of populating replicate transactions into a remote database management system before receiving distribution of changes from the Replication Server. See also **materialization** and Replication Server.

CCSID Coded Character Set Identifier.

CICS See **Customer Information Control System**.

Client-Library A library of routines that is part of Open ClientConnect. Open ClientConnect's Client-Library comprises a subset of the Open ClientConnect Client-Library routines.

commit An instruction to DB2 to make permanent the changes requested in a transaction. See also **transaction**. Contrast with **rollback**.

compensatory operation A transaction operation used to negate the effects of another operation with a lower origin queue ID in the same transaction. Replication Server reconciles compensatory operations and their counterparts so that neither are applied in the replicate database. See also **origin queue ID**.

configure To define to a system the devices, optional features, and programs installed on the system.

connectivity The capability to attach a variety of functional units without modifying them.

Customer Information Control System (CICS) An IBM product that provides an environment for building, using, processing, and maintaining databases in the z/OS and VSE environments for online transaction processing (OLTP). See also **online transaction processing**.

database A set of related data tables and other database objects that are organized and presented to serve a specific purpose.

database management system (DBMS) A computer-based system for defining, creating, manipulating, controlling, managing, and using databases. The software for using a database can be part of the database management system, or it can be a stand-alone database system. Contrast with **relational database management system**.

Database 2 (DB2) A relational database management system for the IBM z/OS operating system. See also **Multiple Virtual System**.

data definition name (DD name) The name of a JCL DD statement (also called "file name") that identifies one or more input/output data sets. See also **data definition statement**.

data definition statement (DD statement) A statement required in an z/OS job to describe an input or output file and associated data sets. See also **job control language**.

data distribution The process of partitioning data to more than one location. Data distribution is distinct from data replication. Distributed data is partitioned among different sites in the distributed system as a single image and is not necessarily replicated data. See also **data replication**.

Data Facility Product (DFP) A program that isolates applications from storage devices, storage management, and storage device hierarchy management.

data replication The process of copying data to remote locations. The copied (replicated) data is then kept synchronized with the primary data. Data replication is distinct from data distribution. Replicated data is stored copies of data in particular sites throughout a system and is not necessarily distributed data. See also **data distribution** and **transaction replication**.

data server A database management system program that responds to client requests. See also **local area network**.

data server interface (DSI) A thread from Replication Server into the replicate database used to transfer messages from Replication Server to the replicate data server.

datatype A keyword that identifies the characteristics of stored information on a computer. Some common datatypes are char, int, smallint, date, time, numeric, and float. Different databases support different datatypes.

Data Warehousing The process of turning the data contained in operational systems into data that can be used by analysts and management to make business decisions. This typically involves:

- Moving the data to new servers, both to avoid overloading operational systems with analysis queries and to take advantage of database engines more efficient for decision support
- Transforming, summarizing, and reconciling the data gathered from disparate operational systems into formats more suitable to decision support

DB2 See **Database 2**.

DBMS See **database management system**.

DD name See **data definition name**.

DD statement See **data definition statement**.

decision support application — A database application characterized by ad hoc queries, reports, calculations, and few data-update transactions.

Decision Support System (DSS) — In contrast to online transaction processing (OLTP), processing that is characterized by fewer, more complex, longer-running transactions from fewer users. Transactions are typically queries only and may reference data from many tables. Database systems designed and tuned for DSS are commonly used to hold Data Warehouse data gathered from operational systems. Contrast with **online transaction processing**.

default — The value, option, or behavior used when a property is not explicitly specified.

DFP — See **Data Facility Product**.

DirectConnect — A Sybase Open Server application that provides access management for non-Sybase databases, copy management (transfer), and remote systems management. Each DirectConnect consists of a server and one or more service libraries to provide access to a specific data source. DirectConnect replaces the products "MDI Database Gateway" and "OmniSQL Access Module." Contrast with **EnterpriseConnect**.

DLL — See **dynamic link library**.

DSI — See **data server interface**.

DSI thread — An outbound queue connection to the target or replicate database.

DSNTIAD — A DB2 sample program written in Assembler language that uses dynamic SQL. DSNTIAD is used to issue all SQL statements except SELECT in batch mode.

DSNTEP2 — A DB2 sample program written in PL/I. It runs SQL statements in batch mode.

DSS — See Decision Support System.

dynamic link library (DLL) — A file containing executable code and data bound to a program at load time or runtime, rather than during linking. The code and data in a dynamic link library can be shared by several applications simultaneously.

exception channel program (EXCP) — A program that allows direct reading of disk and tape files.

file transfer protocol (FTP) — A TCP/IP utility that moves files efficiently between machines.

FTP — See **file transfer protocol**.

function A unit of functional capability expressed by an individual command in an application program interface. See also **application program interface**.

gateway Connectivity software that allows communication between two or more computer systems with different network architectures.

group attach name The name of the DB2 data-sharing group.

guaranteed transaction delivery A LTM for MVS subsystem that ensures the transactions entered into the replication system are properly distributed, even following an interruption caused by the failure of one or more replication system components. See also Replication Agent for DB2.

IEBCOPY An IBM z/OS system utility that performs copy operations including backup copying, unloading, reloading, and compressing partitioned data sets; and merging partitioned data sets to create new partitioned data sets. See also **Multiple Virtual Storage**.

image The complete contents of a row at a particular time. See also **after image** and **before image**.

index A set of pointers that are logically ordered by the values of a key. Indexes are used to provide rapid access to data and can enforce uniqueness on the rows in a table.

initial program load (IPL) A process for loading system programs and preparing a system to run jobs.

Interactive System Productivity Facility (ISPF) An IBM licensed program that serves as a full-screen editor and dialog manager. Used for writing application programs, ISPF provides a means of generating standard screen panels and interactive dialogues between the application programmer and terminal user.

interfaces file A file containing information that Sybase Open Server/Open ClientConnect applications (including Replication Server and Adaptive Server) use to establish TCP/IP connections to other Open Server/Open ClientConnect applications. Sybase interfaces file names differ across platforms: *interfaces* for UNIX platforms, *sql.ini* for Windows NT, and *win.ini* for Windows. See also **SYGWHOST macro** and **TCP/IP**.

IP address The internet protocol address at which TCP/IP expects to find the defined device. See also **TCP/IP**.

IPL See **initial program load**.

ISPF See **Interactive System Productivity Facility**.

isql An interactive SQL client application to Sybase Adaptive Server.

JCL See **job control language**.

job A collection of related programs, identified by appropriate job control statements.

job control language (JCL) In z/OS, a control language used to identify a job and describe its requirements to an operating system. See also **Multiple Virtual Storage**.

K (kilobyte) A measurement of storage equal to 1024 bytes (2^{10}).

local area network (LAN) A computer network located on the user's premises and covering a limited geographical area. Communication within a local area network is not subject to external regulations; however, communication across the LAN boundary can be subject to some form of regulation.

LE Command structure The definition of memory allocation and usage for areas containing the commands and data passed from the Replication Extract to the Replication API. See also Replication API and Replication Extract.

log A written record of operations performed, normally used for recovery. See also **active log**, **archive log**, or **transaction log**.

log extract A subset of Replication Agent for DB2 that interacts with data sources to replicate logged transactions marked for replication. Replication Extract is the log extract supplied with Replication Agent for DB2. Replication Extract uses the Replication Application Program Interface (API) to convey the transactions to the Log Transfer Interface (LTI), which transforms them into Log Transfer Language (LTL) and sends them to a Replication Server. Replication Server then replicates the data changes to replicate databases. See also **Log Transfer Interface**, **Log Transfer Language**, **Log Transfer Manager for z/OS**, Replication API, and Replication Server.

log record sequence number (LRSN) A 6-byte value that DB2 generates and associates with each log record. The LRSN is contained in the DB2 log record header and tablespace header page. The LRSN replaces the RBA in a data-sharing environment, although each member continues to use the RBA for its own logs and BSDS. A number that DB2 generates and associates with each log record. DB2 also uses the LRSN for page versioning. The LRSNs generated by a given DB2 data-sharing group form a strictly increasing sequence for each DB2 log and a strictly increasing sequence for each page across the DB2 group. See also **relative byte address**.

Log Transfer Interface (LTI)

A component of the Log Transfer Manager that provides the session between any z/OS-based Log Transfer Manager for z/OS and a Replication Server. The LTI receives z/OS data changes in Sybase Adaptive Server datatype values and transforms the changes into Log Transfer Language transactions, which are then sent to the Replication Server. See also **Log Transfer Language**, **Log Transfer Manager for z/OS**, and Replication Server.

Log Transfer Language (LTL)

The subset of the Replication Command Language that Log Transfer Manager for z/OS uses for submitting the information retrieved from the primary database transaction logs to Replication Server. See also **primary database**, Replication Agent for DB2, Replication Server, and **transaction log**.

Log Transfer Manager (LTM)

An application that communicates updates made at a primary data server to Replication Server. Examples include the Log Transfer Manager for Sybase Adaptive Server, and Replication Agent for DB2, which consists of Sybase's LTM for z/OS and Replication Extract. See also **Database 2**, **LTM**, Replication Agent for DB2, Replication Extract, and Replication Server.

Log Transfer Manager for z/OS (LTM for MVS)

Log Transfer Manager for z/OS is an application that communicates transactions identified by a Replication Extract to Replication Server. See also Replication Agent for DB2, Replication Extract, and Replication Server.

LTI

See **Log Transfer Interface**.

LTL

See **Log Transfer Language**.

LTM

See **Log Transfer Manager**.

LTM for MVS

See **Log Transfer Manager for z/OS**.

LTM for MVS configuration file

A file that contains the configuration parameters needed to operate LTM for MVS. A sample file is shipped in the *hlq.JCL* data set as the LTMCNFG member. See also **Log Transfer Manager for z/OS**.

LTM Locator

An origin queue ID that identifies the last transaction operation successfully saved in Replication Server's inbound queue. Replication Extract uses the LTM Locator to identify the location in the log to begin scanning during start-up and restart. See also **origin queue ID** and Replication Extract.

LTMOBJECTS table

See **replication registration table**.

mainframe

An IBM or IBM-compatible computer that conforms to the System/370 or System/390 architecture and supports z/OS. Sometimes referred to as a host. See also **Multiple Virtual Storage**.

maintenance user ID An identifier defined in the replicate database that is used by Replication Server to identify the Replication Server maintenance user. See also Replication Server.

materialization The method or process by which data is placed into a replicate table in preparation for replication. See also **replicate table**.

module A program unit that is discrete and identifiable with respect to compiling, combining with other units, and loading; for example, the input to or output from an assembler, compiler, linkage editor, or executive routine.

z/OS See **Multiple Virtual Storage**.

NCP See **Network Control Program**.

network A configuration of data processing devices and software connected for information exchange.

nonatomic materialization The process of populating a replicate database using Replication Server commands that allows client applications to execute transactions against the primary database while the subscription data is unloaded.

OLTP See **online transaction processing**.

online transaction processing (OLTP) A system that is characterized by many small, simple, short-duration transactions from many users. These transactions tend to be oriented toward data manipulation (insert, update, delete), and to only reference one or a few tables.

Open ClientConnect application An application written using Sybase Open ClientConnect libraries.

Open ClientConnect A Sybase product that provides customer applications, third-party products, and other Sybase products with the interfaces required to communicate with Open ClientConnect and Open ServerConnect applications.
Open ClientConnect allows Open Client-style communication between DB2 and Replication Server.

Open Client/Server Architecture The combination of Open ClientConnect and Open ServerConnect, which provide the connectivity backbone for Sybase's distributed client-server architecture.

Open Server A Sybase product that provides the tools and interfaces required to create a custom server. Clients can route requests to DirectConnect through an Open Server that they configured to meet specific needs, such as the preprocessing of SQL statements or decision making about routing RPCs to Transaction Router Service for DB2 or to other servers.

Open Server application A custom server built with Sybase Open Server.

operating system A group of programs that translates commands to the computer, helping to perform such tasks as creating files, running programs, and printing documents.

origin queue ID An identifier built by Replication Extract that uniquely identifies each transaction operation transmitted to Log Transfer Manager (LTM) for z/OS. Replication Server maintains the origin queue ID of the last update that was successfully stored in the Replication Server stable queue during a given connection. LTM for MVS requests this identifier from Replication Server upon start-up and restart. See also **Log Transfer Manager for z/OS**, **LTM Locator**, Replication Agent for DB2, Replication Extract, and Replication Server.

packer routine A routine that converts a primary database log record into a LTL record, which is then sent to Replication Server. See also **Log Transfer Language** and Replication Server.

parallel sessions Multiple sessions running simultaneously between two LUs across a single LU 6.2 conversation. See also **session**.

permission The level of access to an object, resource, or function.

ping A method used to create a connection to determine the availability of a process or data source.

ping interval The period of time between pings. See also **ping**.

primary data The version of a set of data in a replicated data system from which replication occurs. With Replication Agent for DB2, primary data is stored in DB2. See also **Database 2**, Replication Agent for DB2, and Replication Server.

primary database The database that contains the transactions to be replicated or processed. See also **primary data**.

primary table A DB2 table used as a source for replication or processing. See also **Database 2**, **primary data**, and **primary database**.

program temporary fix (PTF) A temporary solution or by-pass of a problem to resolve a defect in a current unaltered release of a program.

protocol A set of standards that govern the behavior of computers communicating on a network.

PTF See **program temporary fix**.

QID See **queue ID**.

queue A list constructed and maintained so that the next data element to be retrieved is the one stored first.

queue ID (QID) The DB2 relative byte address of the log record associated with the insert, update, or delete that Replication Agent for DB2 sends to Replication Server. Replication Server maintains the QID of the last update that was successfully stored in the Replication Server stable queue for this connection. Requesting this QID is part of the initial conversation that occurs between Replication Agent for DB2 and Replication Server (analogous to the get truncation point in Log Transfer Manager for Sybase Adaptive Server). See **relative byte address**.

RACF See **Resource Access Control Facility**.

RBA See **relative byte address**.

RCB See **receive event control block**.

RCL See **Replication Command Language**.

RDBMS See **relational database management system**.

receive event control block (receive ECB) The event control block passed to the Replication Extract at start-up that, when posted, indicates that there are messages for the Replication Extract. The Replication Extract can obtain the messages using the Replication API's LTMSEND function call. See also Replication API and Replication Extract.

record A set of one or more related data items grouped for processing.

recovery The process of rebuilding one or more databases from database dumps or log dumps.

relational database A collection of data in which relationships between data items are explicitly specified as equally accessible attributes. The data is viewed as being stored in tables consisting of columns (data items) and rows (units of information). Relational databases can be accessed by SQL requests. See also **Structured Query Language**.

relational database management system (RDBMS) The application that controls relational databases. See also **relational database**. Contrast with **database management system**.

relative byte address (RBA) The address of a byte in the DB2 log. The address is the byte's offset from the beginning of the log.

replicate database — The database to which Replication Server is replicating transactions. There can be more than one replicate database for each Replication Server. See also **primary database** and Replication Server.

replicate table — A database management system table maintained by Replication Server. See also Replication Server.

replicate transactions — Information copied from a primary database using Replication Server and stored in a replicate table. See also **replicate table** and Replication Server.

replicated data system — A data-processing system in which data is replicated in multiple databases to provide remote users with the benefit of local data access.

replication — See **transaction replication**.

Replication Agent — See Replication Agent for DB2.

Replication Agent for DB2 — The Sybase application consisting of LTM for MVS and Replication Extract. Replication Agent for DB2 allows the replication of transactions from a primary database to replicate databases through Sybase's Replication Server. See also **Log Transfer Manager for z/OS**, Replication Extract, and Replication Server.

Replication API — An interface for LTM for MVS used by Replication Extracts to communicate transactions to Sybase's replication system. See also **application program interface**, **Log Transfer Manager for z/OS**, and Replication Extract.

Replication Command Language (RCL) — The command language used with Replication Server.

replication definition — A description of a table for which subscriptions can be created. The definition, maintained by Replication Server, includes information about the columns in the table, the location of the primary version of the table, and some options that specify how the table can be used. See also Replication Server and **subscription**.

Replication Extract — A log extract that is a component of Replication Agent for DB2. Replication Extract interacts with primary data servers to capture updates of transactions marked for replication. See also **log extract** and Replication Agent for DB2.

replication registration table — A DB2 table named LTMOBJECTS that controls the replication of DB2 data and each row of which identifies a primary DB2 table that has been registered by the user. See also **Database 2**.

Replication Server — A Sybase Open Server application that maintains replicate transactions received from a data source, which can be either the source data server or another Replication Server. See also **Adaptive Server Enterprise**.

Replication Server Interface (RSI) — A thread from a primary Replication Server into a replicate Replication Server used to transfer commands from the RSI outbound stable queue to the replicate Replication Server. There is one RSI thread for each replicate Replication Server that is a recipient of commands from the primary Replication Server. See also Replication Server and **stable queue**.

Replication Server System Database (RSSD) — The system database in which Replication Server stores its system information. The tables reside in a Sybase Adaptive Server database. See also **Database 2**, Replication Server, and **Adaptive Server Enterprise**.

Replication Toolkit — A programming guide for the Replication API and sample log extract programs to allow a customer or third-party vendor to develop a Log Extract program that interfaces with the Log Transfer Manager.

request — One or more database operations the application sends as one unit to the database. During a request, the application gives up control to the DBMS and waits for its response. Depending on the response, the application commits or rolls back the request. One or more requests can be grouped into a single unit of work.

resource — Any facility of a computing system or operating system required by a job or task, including main storage, input/output devices, processing unit, data sets, and control or processing programs.

Resource Access Control Facility (RACF) — A security package for IBM mainframes.

rollback — A transaction operation instructing the DBMS to ignore the changes requested in a transaction. See also **transaction**. Contrast with **commit**.

route — A connection from a primary Replication Server to a replicate Replication Server. See also Replication Server.

RPL — Request parameter list.

RSI — See Replication Server **interface**.

RSSD — See Replication Server **System Database**.

source — The primary database or the database that contains the data to be replicated or processed.

SPUFI See **SQL Processor Using File Input**.

SQL See **Structured Query Language**.

sqledit A utility for creating and editing *sql.ini* files and file entries.

SQL Processor Using File Input (SPUFI) A DB2 facility that enables users to execute SQL statements without embedding them in an application program.

SQL Server See **Adaptive Server Enterprise**.

SSID See **subsystem identifier**.

stable queue A store-and-forward queue in which Replication Server messages destined for a route or database connection are stored. Messages written into the stable queue remain there until they can be delivered to the replicate Replication Server or replicate database. See also Replication Server.

statement In programming languages, a language construct that represents a step in a sequence of actions or a set of declarations.

storage group A named set of direct access storage device (DASD) volumes. DB2 data can be stored in storage groups. See also **Database 2**.

Structured Query Language (SQL) A language developed by IBM to process data in a relational database. SQL is an industry standard.

subscription A request for Replication Server to maintain either a replicated copy of a table or a set of rows from a table in a database at a specified site. See also Replication Server.

subsystem In IBM z/OS/CICS, a single instance of a relational database.

subsystem identifier (SSID) The name of a DB2 subsystem.

SYGWHOST macro The SYGWHOST TYPE=ENTRY macro, distributed in the *XCPHPING* member of the *hlq.JCL* data set, serves as a directory containing network addresses and other information that controls how Replication Agent for DB2 connects to other servers, and configures the Open ClientConnect component of LTM for MVS to use IBM TCP/IP for a TCP/IP connection between Replication Agent for DB2 and Replication Server. The SYGWHOST TYPE=ENTRY macro can be a member of a Partitioned Data Set (PDS) or a sequential file.

	The Replication Agent for DB2 TCP/IP installation requires a SYGWHOST TYPE=ENTRY macro entry for each primary Replication Server in your replication system. You can define any number of Replication Server entries in the SYGWHOST macro. The SYGWHOST TYPE=ENTRY macro replaces the interfaces file required in earlier versions of the Replication Agent for DB2 software. See also **interfaces file** and **TCP/IP**.
synchronization	The method that ensures primary and replicate tables are equivalent. For example, if transaction number 100 is successful in the primary database, the 100th transaction in the replicate database should also be successful.
systems management	The process of initiating, configuring, monitoring, and adjusting applications on a system.
sysplex	A set of z/OS systems communicating and cooperating with each other through certain multisystem hardware components and software services that protect customer workloads.
system administrator	The person at a computer installation who designs, controls, and manages the use of the computer system.
table	In a relational database management system, a two-dimensional array of data or a named data object that contains a specific number of unordered rows composed of a grouping of columns specific for the table.
tablespace	In z/OS, a page set that stores the records in one or more tables. See also **z/OS/CICS**.
target	The database receiving the replicate data in a replication environment. Also known as a destination.
task control block (TCB)	An z/OS control block used to communicate information about tasks within an address space. There is one TCB per task in an address space. An address space can support one or more tasks, but any task may reside in only one address space. See also **Database 2** and **Multiple Virtual Storage**.
TCB	See **task control block**.
TCP/IP	See **Transmission Control Protocol/Internet Protocol**.
trace	The process of recording the sequence in which the statements in a program are executed and, optionally, the values of the program variables used in the statements.

transaction A group that can include zero, one, or many database transaction operations (including inserts, updates, and deletes) that are applied or rejected as a whole.

transactional consistency A condition in which all transactions in the primary database are applied in the replicate database in the same order that they were applied in the primary database.

transaction log The log of transactions kept by a database server. Replication Extract reads these logs to identify the changes to primary databases. See also **log extract**, Replication Agent for DB2, and Replication Extract.

transaction program A program that processes transactions. There are two kinds of transaction programs: application transaction programs and service transaction programs.

transaction replication The process of copying transaction operations (updates, inserts, and deletes) to remote locations. The replicated transactions are then kept consistent with the primary transaction. See also **data duplication** and **data replication**.

Transact-SQL A Sybase Adaptive Server SQL extension that allows procedural programming within a DBMS to access and manipulate data. See also **Adaptive Server Enterprise** and **Structured Query Language**.

Transmission Control Protocol/Internet Protocol (TCP/IP) A computer network communications protocol designed by the Department of Defense Advanced Research Projects Agency. Contrast with **advanced program-to-program communications.**

truncation point The location in a DBMS transaction log that separates the part of the log verifiably received by Replication Server from the part that has not yet been verifiably received. In Adaptive Server, the transaction log entries that precede the truncation point have already entered the replication system, so that portion of the log can be safely truncated. See also **database management system**, **LTM Locator**, **origin queue ID**, Replication Server, **Adaptive Server Enterprise**, and **transaction log**.

UNIX An operating system developed by Bell Laboratories that allows multiple programs to run and multiple users to work concurrently.

URID See **UR identifier**.

UR identifier The log RBA of the beginning of a recovery unit. The UR identifier is the earliest relative byte address required to process the UR during restart. See also **relative byte address**.

user-defined function A user-defined, custom-built, Replication Server function whose name and parameters exactly match the name and parameters of a replicated stored procedure.

user exit

A user-written transaction program that can be given control at a determined point in a program.

In Replication Agent for DB2, a program called by the Replication API's LTMSEND function call to alter transaction operations before sending them to Replication Server. See also **Replication API**.

view

An alternative representation of data from one or more tables. A view can include all or some of the columns in a table. Refer to Sybase Adaptive Server or IBM DB2 documentation for more information about Adaptive Server views. See also **Database 2** and **Adaptive Server Enterprise**.

Virtual Storage Access Method (VSAM)

An IBM-licensed program that controls reading and writing data to and from a disk.

VSAM

See **Virtual Storage Access Method**.

WAN

See **wide area network**.

wide area network (WAN)

A network that provides communication services to a geographic area larger than that served by a LAN. Contrast with **local area network**.