IBM InfoSphere Data Replication for IMS for z/OS
Version 11 Release 3

Guide and Reference

IBM InfoSphere Data Replication for IMS for z/OS
Version 11 Release 3

Guide and Reference

Note
Before using this information and the product that it supports, read the information in "Notices" on page 235.

This edition applies to InfoSphere Data Replication for IMS for z/OS (program number 5655-IM1) and to all subsequent releases and modifications until otherwise indicated in new editions.

Chapter 1. Overview of Data Replication for IMS

You can use Data Replication for IMS™ to produce copies of your IMS databases and maintain current data in near-real time, typically on geographically-dispersed IMS subsystems.

Purpose

IBM® InfoSphere® Data Replication for IMS for z/OS® addresses your organizational requirements for reliable and available data:

- High availability and disaster recovery (HADR)
- Business intelligence
- Redundancy
- Data backup

High availability and disaster recovery (HADR)

Data center downtime is a significant interruption that affects productivity, revenue, and trust. For example, a global banking enterprise lost its secondary data centers in the aftermath of a terrorist attack because they were too close to the primary sites. The ATM network was down for days, at incalculable cost to the company and its customers. Extreme weather events, such as hurricanes or other natural disasters, can have similar consequences.

Data Replication for IMS supports your data availability strategy by helping you to ensure the availability of your data from secondary or backup instances. It offers the following advantages for your HADR solution:

- No geographic restrictions on the distance between the primary and secondary sites
- Quicker recovery from failures compared to hardware-based solutions

Business intelligence

Data Replication for IMS also supports scenarios that distribute your business intelligence workload to a secondary, read-only platform where analysts can run queries.

Redundancy

Use your synchronized replicas for active processing while you perform scheduled maintenance of your source databases.

Data backup

Create synchronized backups of mission critical data in near-real time.

How Data Replication for IMS works

The illustration shows how Data Replication for IMS maintains synchronized replicas in high-volume IMS subsystems that undergo rapid rates of change:

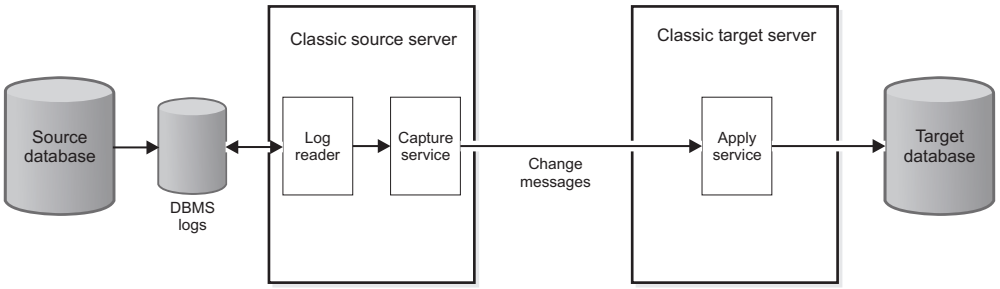


Figure 1. Classic data servers capture changes from IMS log streams and apply those changes to target databases

A Classic source server uses a log reader to process IMS log data and capture changes to a source database. Capture processing then packages these changes as change messages that describe insert, update, and delete operations on the data. The source server sends the change messages to a target server at a different site that applies the changes to a replica of the source database.

Data Replication for IMS is *unidirectional*, which means that you send changes from one site to another instead of in both directions at once. You use Data Replication for IMS with target databases that you do not update actively so that you can maintain synchronization of your data between the source and target.

Technical capabilities

Data Replication for IMS can capture changes from a single DB/DC or DBCTL subsystem. It can also capture changes from hundreds or thousands of shared databases in an IMS data-sharing environment. You can capture updates from multiple IMS DL/I batch jobs, DB/DC subsystems or DBCTL subsystems in a sysplex. Data Replication for IMS also supports high availability environments that implement FDBR (Fast Database Recovery) regions.

Data Replication for IMS has the following key components and capabilities:

- Subscriptions
- Transactional consistency
- Replicating historical changes
- Monitoring and reporting

Subscriptions

You organize the databases that support a given application by mapping them to their target databases within a *subscription*, a unique combination of source databases, memory caches, and communication paths. Because of its autonomous structure, you can start, stop, and maintain replication for a subscription independently of other subscriptions. Stopping replication for one subscription has no effect on the operations of others.

Transactional consistency

Data Replication for IMS can manage transaction processing across multiple logical partitions and databases. A subscription maintains the sequence of transactions as they occur at the source by applying changes to a given record in the correct order.

Replicating historical changes

Planned or unplanned outages can cause target databases to fall behind current processing and to go out of synchronization. Data Replication for

IMS can automatically catch up with unprocessed changes that occurred in the past, whether replication stopped due to replication or memory errors, link loss, or apply errors.

Data Replication for IMS maintains *bookmark* information that specifies where the log reader begins again in the event of an outage. The change data that the source server maintains in caches can reduce the time that it takes to catch up to current processing.

Monitoring and reporting

You can review current and accurate metrics in the Classic Data Architect, a supplied graphical user interface. Using this tool, you can measure resource consumption, latency, throughput, and memory usage, enabling you to evaluate the replication process and optimize your environment.

Latency and DL/I batch applications

Classic change data capture can capture changes made by DL/I batch jobs. However, these jobs can take hours to complete, which introduces latency.

The logs are not available for change data capture until each batch job step finishes, because Classic change data capture cannot access the IMS log until it closes. When a DL/I batch job starts and identifies itself as a subsystem that affects ordering, the IMS log reader service must suspend ordering operations until each DL/I batch job step finishes processing. The process of ordering and merging resumes when the source server can access the log.

To help manage latency, you can specify a list of IMS subsystems (both DL/I batch and online) to exclude from ordering decisions.

If your business requirements cannot accept latency, convert any existing DL/I batch workload to run as a batch message processing (BMP) program under the control of a DB/DC or DBCTL subsystem.

Release notes for IBM InfoSphere Data Replication for IMS for z/OS, Version 11.3

The Release Notes® include information about new functionality included in Version 11.3 and changes in existing functionality.

Contents

- "What's New"
- "Migration considerations" on page 5

What's New

The following items are new features in Version 11.3.

Relaxed validation of replication between databases

Data Replication for IMS allows you to configure the target server to allow replication to start when differences are detected in the source or target DBD attributes for a database. This capability allows you to make certain kinds of structural changes to a target database and once complete, continue to replicate source changes to the new target until the source database can be upgraded to match.

In a GDPS® Active-Standby environment this capability can allow you to adapt your applications to changing business requirements more rapidly with minimal replication down time. See "Synchronization of source and target IMS databases" on page 21 for more information about the requirements for using this capability.

IMS Version 13 support

Data Replication for IMS is capable of accessing IMS logs produced by IMS Version 13 and the DBD and ACB library structures used in that version of IMS. In addition, Data Replication for IMS supports the following IMS enhancements:

- Capture and replication of updates performed by applications using FLD calls. See "Apply processing" on page 121.
- Capture and replication of subset pointer updates. See "Subset pointer replication" on page 123.
- Capture and replication of positioning information for segments with an insert rule of HERE. "Insert HERE positioning" on page 123.

Support for these enhancements requires IMS APARPI07353 installed at the source site and properly augmenting the source database.

Enhanced SSA generation and positioning support

To support IMS APAR PI07353, the target server supports generation of segment search arguments (SSAs) that utilize the IMS O command code to locate non-uniquely keyed segments and uses improved key feedback analysis and target database positioning logic. You enable this capability by using a new IMSV11CMPAT configuration parameter. To use this capability, IMS Version 12 or later must be installed at the target site, regardless of the IMS release used at the source site. See "IMSV11CMPAT" on page 187.

Improved parallelism for source UORs that require serialization using completed UOR tracking

Data Replication for IMS supports a new option that allows more UORs to be applied in parallel. Previously UORs would have forced serialization because adaptive apply cannot be used to replay them during restart operations. You activate this capability by using configuration parameters. This capability also requires additional information to be stored in the bookmark database for use during restart processing for a subscription. See "Completed UOR tracking" on page 84.

Deadlock prevention enhancements

Data Replication for IMS allows you to configure dependency analysis to only track the DEDB areas or databases that were updated by a source application and use just this level of information in parallelism determination. Use of area level dependency analysis eliminates the potential for deadlocks created when Data Replication for IMS attempts to apply two UORs in parallel that randomize to the same block in the target database. See "Dependency analysis" on page 125.

Bookmark database installation customization enhancements

The installation customization process is enhanced to allow you to activate completed UOR tracking. When enabled, the default allocations for the bookmark database support up to five subscriptions that require completed UOR tracking. Completed UOR tracking is enabled by default. See "Working with the customization parameters file" on page 39.

New sample members and procedures are provided that allow you to use a DEDB bookmark database. The recommended implementation uses the

virtual storage option hosted in a data space - SHARELVL(1). See “Creating a bookmark database” on page 81.

DL/I batch log filtering

Change stream start processing is enhanced to use the DBD allocation information stored in DBRC for DL/I batch jobs to eliminate access to IMS logs that do not contain any data capture log records. IMS logs produced by a DL/I job that did not update any databases that were augmented for data capture will no longer be accessed.

The DBDLIB definition for each DBD referenced in a DBRC PRILOG entry is inspected to determine whether the DL/I log can contain data capture log records If none of the segments were augmented to generate data capture logs records, the log associated with that PRILOG entry does not contain anything of interest and will not be accessed by the data server. Likewise, if there is no database allocation information associated with a PRILOG entry, it's associated log will not be accessed.

Migration considerations

The following migration considerations apply to version 11.3 of Data Replication for IMS.

Migrating source server and target server configurations

The installation procedure to create source servers and target servers includes the step required to migrate a server configuration. For more information, see Installing Classic data servers.

Migrating subscriptions and replication mappings

The installation procedure to create source servers and target servers includes the step required to migrate subscriptions and replication mapping data sets. For more information, see Installing Classic data servers.

If you are migrating from version 10.1 to version 11.3, an automatic reformatting is performed the first time that you start the source or target server. Recommendation: Ensure that your existing subscription and replication mapping files are backed up before starting the version 11.3 servers.

Migrating the bookmark database

You can use an existing bookmark database with version 11.3 provided that the configuration does not try to activate completed UOR tracking. If you want to use completed UOR tracking and the new bookmark database structure that it requires, you need to convert an existing bookmark database to the new structure. However, no procedures are provided for performing that conversion.

If you continue to use a HDAM database to store bookmark information , you can use IMS or third party unload/reload utilities to migrate existing bookmark information to the new bookmark database. The information stored in the bookmark segment in prior releases is compatible with the new form used when completed UOR tracking is active.

Release notes for InfoSphere Classic Data Architect, Version 11.3

Updated information for Version 11.3 of InfoSphere Classic Data Architect is provided in release notes.

Migrating the workspace from a previous version of CDA

If you are migrating from a previous version of Classic Data Architect you might have to migrate your workspace the first time you start Classic Data Architect. The new views and perspective changes are not displayed until after the migration. The procedure for migrating from an older version of Classic Data Architect varies depending on the version you are migrating from:

Migrating from CDA Version 11.1

If you are using the Data perspective you need to open it the first time you use version 11.3 of CDA by selecting **Window** > **Open Perspective** > **Data**.

Migrating from CDA Version 10.1

- The first time you start the new version of Classic Data Architect, a dialog box will appear asking you to confirm the migration of the workspace. Classic Data Architect will restart when the migration completes.
- If you are using the Data perspective you need to open it the first time you use version 11.3 of CDA by selecting **Window** > **Open Perspective** > **Data**.

Migrating from CDA Version 9.5

- Reset the Data perspective by selecting **Window** > **Reset Perspective**.
- Display the new default perspective by selecting **Window** > **Open Perspective** > **Replication**.
- If you are using the Data perspective you need to open it the first time you use version 11.3 of CDA by selecting **Window** > **Open Perspective** > **Data**.
- Recreate your data source connections in the Data Source Explorer.

When the workspace is migrated, the new views, the Replication perspective, and the Data perspective are available.

Technical overview

To understand how Data Replication for IMS works, learn about the architecture and components, transactional consistency and latency, and how replication processing catches up to near-real time after replication stops and restarts.

Architecture of Data Replication for IMS

Data Replication for IMS consists of a set of Classic data servers, subscriptions, and other components that enable you to replicate IMS data.

The illustration and the explanations that follow exemplify how Data Replication for IMS works.

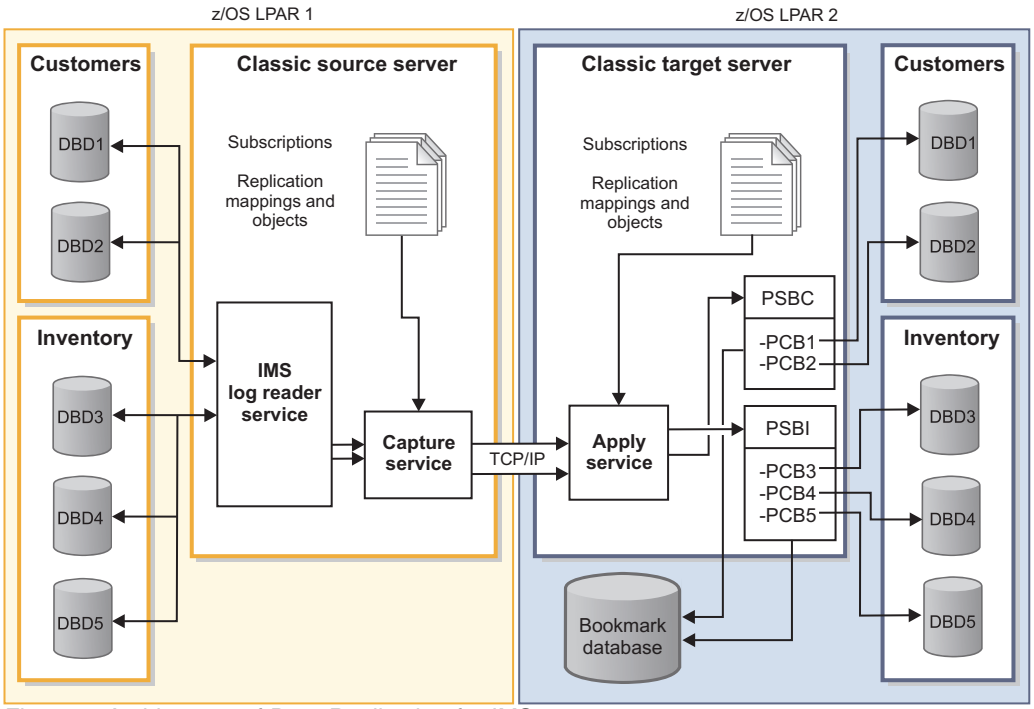


Figure 2. Architecture of Data Replication for IMS

For definitions of italicized terms in these descriptions, see the Glossary.

Geographically dispersed sites

In this example, the IMS databases on the logical partition (LPAR 1) at the source site support related business applications and data. The purpose of this solution is to back up customer and inventory data at a remote site. The target databases are on LPAR 2 at a data processing center in a separate geographical location.

Classic data servers

A *Classic data server* runs in its own address space and contains the services and other components that maintain the environment, process change data, and administer or monitor replication.

A *source server* reads log records that describe changes made to source IMS databases and sends them to a *target server* that runs on a separate sysplex. The target server applies the changes to the target IMS databases.

The target server can detect the state of connected IMS subsystems and retry its connection when IMS is unavailable in cases like these:

- IMS is not running when the Classic data server starts.
- IMS shuts down while the Classic data server is running.

IMS log reader service

The log reader service reads IMS log files and sends change messages to the capture service that describe inserts, updates, deletes, and commits.

In high-volume replication environments, you can take advantage of the support provided, starting in IMS Version 12, for the exploitation of IMS logs residing in the extended address space (EAS) of an extended address volume (EAV) and similarly configured striped IMS logs.

Subscriptions

A *subscription* manages replication for databases that relate to a given application, such as customers or inventory. A subscription replicates independently of other subscriptions, thereby providing greater reliability. When you stop replication for one subscription, this action has no affect on the operations of other subscriptions. Each subscription maintains this autonomy by using its own storage, memory, and communication paths.

When you mark a subscription as *persistent*, it restarts automatically if it encounters a recoverable problem. If you enable *parallel apply* for a subscription, you can improve performance by applying changes concurrently to different databases or to different records within the same databases.

A subscription consists of the following independent components and features:

- A uniform resource locator (URL) that identifies the target server
- A TCP/IP connection between the capture and apply services
- User-configurable memory caches that enable your deployment to manage the flow of data
- Bookmark information that provides a restart position if replication stops

Replication objects

Replication objects represent your source and target IMS databases to the Classic data server when you create your subscriptions for replication. You connect these replication objects by defining replication mappings within each subscription.

Replication mappings

A subscription contains *replication mappings* that define the databases that you want to replicate by specifying a connection between a replication object and a subscription. In Data Replication for IMS, these replication objects are database descriptions (DBDs).

A replication mapping is in one of the following states:

Parked

Indicates that the replication mapping is not replicating changes.

Active

Indicates that the replication mapping is replicating changes.

Capture service

The capture service runs in the source server and manages change data capture for replication, including interactions with the log reader service and change streams. The capture service transmits change messages to the apply service for processing.

Target IMS subsystem and databases

The target databases are in an IMS subsystem that is typically on a separate z/OS logical partition (LPAR). However, you can use Data Replication for IMS to replicate between subsystems on the same LPAR, typically in a test environment.

Consider devoting a dedicated LPAR to apply processing so that you can optimize the subsystem and improve application performance. You can

query the target databases for business intelligence, but to maintain matching data between the source and target, do not actively update the target databases.

Apply service

Data Replication for IMS requires a program specification block, or *apply PSB*, for each subscription. An apply PSB contains a program communication block (PCB) for each target database. An apply service, which runs in the target server, schedules the first apply PSB for each subscription and applies changes to the target database.

You can generate source files for apply PSBs automatically by using the Subscription wizards in the Classic Data Architect, or create them manually.

Database resource adapter service (not shown)

The target server runs a database resource adapter (DRA) service to communicate with the IMS DRA interface and manage access to IMS. Data Replication for IMS uses the DRA to apply changes to the target databases.

Bookmark database

The target IMS subsystem must contain a bookmark database per apply service to store a current restart point in the event of an outage.

For each source unit of recovery (UOR), the apply service performs the following actions in sequence:

1. Apply the changes to the target database.
2. Update the bookmark database with the new restart position for the subscription, if necessary.
3. Commit the UOR.

Event logs (not shown)

Each Classic data server writes messages about data server and subscription activity to an event log.

You can review the events in the Classic Data Architect or you can print the logs by using the log print utility (CACPRTLG).

Classic Data Architect (not shown)

Data Replication for IMS provides a tool with a graphical user interface (GUI), the Classic Data Architect, which connects to the source and target servers to simplify configuration, subscription definition, monitoring, and administration.

Change-data processing

A Data Replication for IMS environment reads IMS logs in source subsystems and preserves transactional consistency by writing change data to the target databases in the correct order.

Apply processing in the target subsystem maintains *bookmarks*, each of which describes a restart position for a subscription. They allow the log reader service to continue processing the logs from where it left off and maintain transactional consistency if errors occur or replication stops.

The source server exchanges metadata about subscriptions, replication objects, and replication mappings with the target server when you start replication for a

subscription. This enables the target server to track the components of your replication environment that manage change data processing and transactional integrity.

Transactional consistency

Data Replication for IMS preserves transactional consistency between your source and target databases by maintaining the sequence of changes to a database record.

In parallel apply processing, transactions can be applied to the target in a different order than they were applied at the source, provided that any dependent transactions are applied first.

Because low latency is also a high priority in replication, your subscription design must optimize the tradeoff between latency and maintaining precise order.

Units of recovery:

Data Replication for IMS maintains transactional consistency by applying each source transaction as a transaction to your target IMS databases.

A *unit of recovery* (UOR) is a group of operations that are either committed or backed out as a group. These operations represent a transaction, such as an online transaction, or similar changes that applications generate. You can configure the capture server to treat multiple transactions as a single transaction by specifying the UORGROUPCOUNT configuration parameter.

These programs are examples of applications that modify IMS databases:

- Batch message processing (BMP) applications
- The IMS Database Resource Adapter (DRA)
- Open Database Access (ODBA) applications

Typically, a BMP application generates multiple UORs, one for each checkpoint that the application issues. Ideally, a DL/I batch application also issues checkpoints and generates multiple UORs, but this behavior is optional.

Transactions and latency: scenarios:

Data Replication for IMS maintains a balance between transactional consistency and latency in a parallel apply environment.

Data Replication for IMS maintains consistency by applying transactions in the correct sequence. For example, a withdrawal from your bank account might succeed only if the bank first completes your transfer to that account.

Low latency is also a high priority in replication processing. The following scenarios describe how Data Replication for IMS maintains transactional consistency while minimizing latency.

You can monitor latency and throughput in the Classic Data Architect.

Subscriptions that perform parallel apply

When you enable parallel apply for a subscription, you can improve performance by controlling how you apply changes concurrently at the target site. Serial subscriptions always apply transactions in the same order in which they occurred

at the data source. To minimize latency, parallel subscriptions apply transactions in strict order only to a *single record*. Parallel subscriptions place a lower priority on maintaining a strict sequence for transactions that change *separate records* or *separate databases*.

Parallel apply processing is likely to write transactions out of order when any of these conditions are true:

- The subscription is processing concurrent transactions at the source.
- The source transactions modify different records in the same database.
- The source transactions modify different databases or partitions.

This less stringent approach to maintaining order enables the target server to perform more parallel processing.

Transactions that modify databases in different subscriptions

Transactions that change databases in more than one subscription can result in splitting a UOR into two or more apply UORs that arrive at the target databases in an unpredictable order.

For this reason, Data Replication for IMS cannot maintain consistency for transactions that modify databases in multiple subscriptions. Design your subscriptions to include all databases that are related to a single business application, such as customers or inventory, to keep the number of such transactions to a minimum.

Situations in which apply processing can fall behind

To keep latency low, Data Replication for IMS applies changes to a target database almost immediately after an application commits a unit of recovery (UOR) in the source database. In some cases, however, the contents of a target database can lag behind the source:

- Planned or unplanned replication outages
- Large UORs that take longer to process
- High volumes of updates
- Heavy load or burst conditions
- Long running DL/I batch jobs that suspend source log ordering

In situations like these, applying updates to the target data sets might lag behind the source updates until the condition causing the slowdown is resolved. A very large UOR can lag behind because the source server only sends committed UORs to the target. This means that a UOR cannot start apply processing until it is complete at the source. Both the source and target servers use caching mechanisms to ensure that maximum throughput is achieved in these situations.

Processing historical changes if replication stops

Planned or unplanned interruptions of replication can cause a subscription to lag behind current processing. Data Replication for IMS manages this situation for you in most cases.

Your replication environment can go back in time and process units of recovery (UORs) that were not processed while replication was inactive for a subscription.

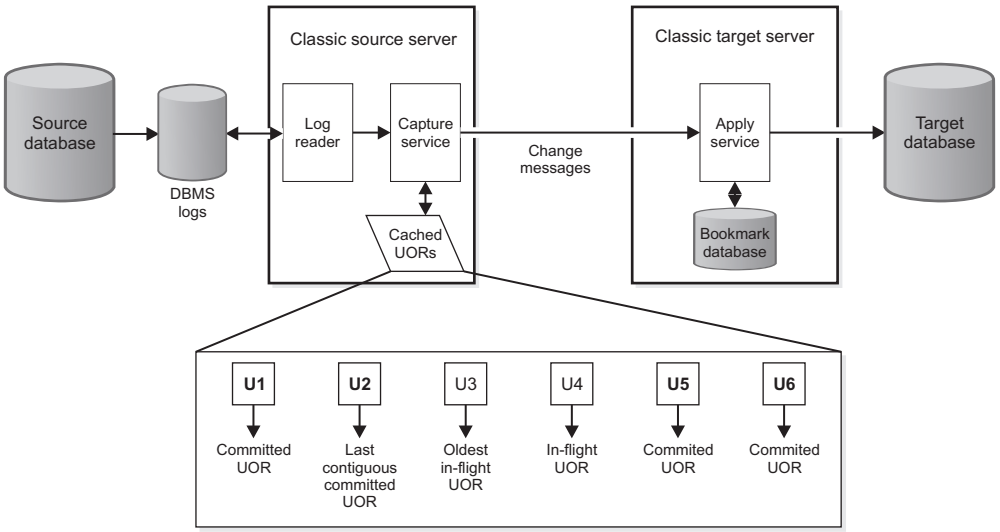


Figure 3. Historical changes available for replication processing after an outage

The target server maintains information in a bookmark database that the source server and log reader service can use to restart change data capture at the correct log position. The exact position depends upon the last *contiguous committed* UOR and the oldest *in-flight* UOR (U2 and U3 in the illustration). Change data capture begins reading the logs with the first change for any uncommitted UORs. If changes for the log position are unavailable for immediate retrieval in the cache, the log reader retrieves them from the logs.

Replication can stop under these conditions:

A subscription stops automatically

- Your replication environment encounters a serious inconsistency between the source and target, such as the following inconsistencies:
  - Apply errors

    For example, when the replication mapping is in standard apply mode:
    - Attempting to update a record that does not exist
    - Inserting a record and finding a duplicate key record that already exists
  - Validation processing that detects mismatches between database descriptions (DBDs)
- An error or system outage occurs:
  - Link loss

    A link loss can be a lost TCP/IP connection between source and target servers.
  - Other internal errors

You stop and then subsequently restart a subscription

- The source or target database is offline for maintenance.
- You perform administrative operations on a replication mapping or a subscription. such as changing the state of a replication mapping from Parked to Active.

While replication is inactive for a subscription, you cannot replicate any of the databases in the replication mappings. Processing for other subscriptions continues. Use the following approach if you want to replicate most of the databases in the subscription, or all the databases but one:

1. Stop the subscription.
2. Park the replication mappings that you do not want to replicate.
3. Restart replication for the subscription.

The illustration shows how the source server allocates any required resources, such as change streams, to catch up to current processing when you restart replication:

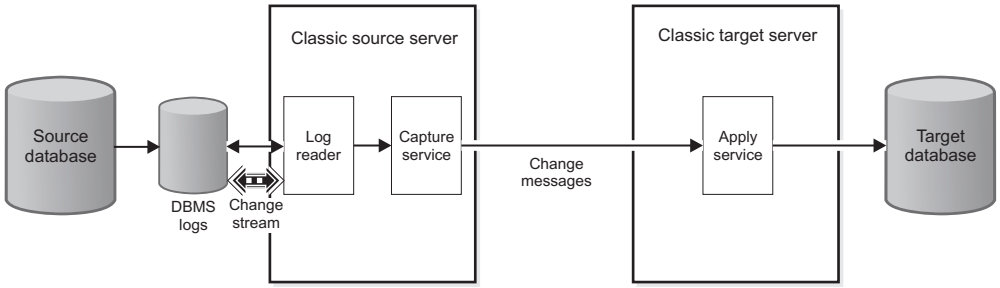


Figure 4. The source server creates a change stream for specific subscriptions that are catching up

When replication starts again, subscriptions that are behind receive higher priority until all subscriptions are replicating current changes.

Resource consumption

Data Replication for IMS increases resource consumption on the source and target logical partitions (LPARs). Examples include logging and archiving activity, CPU consumption, network bandwidth, and memory.

Log records for data capture

Data Replication for IMS relies upon a special type of IMS log record that provides additional information for change data capture. The log record is typically much larger than a standard undo or redo log record. The amount of log data is likely to increase by a factor of 1.5 of your current online logging activity. The online logs cycle faster because of the increased log activity, but you do not necessarily have to increase their size or number unless actual logging activity requires it.

Classic change data capture enables you to reduce the number of log records by augmenting for additional logging only the subset of database segments that you are replicating. You can limit the size of the log records even more if your source data is compliant with the design principles of third normal form.

Archive logs

- For IMS data sources

  IMS creates more archive logs, which increases the amount of direct access storage devices (DASD), virtual tape, or physical tape that your LPAR requires. If your LPAR runs DL/I batch workload that logs to DASD, you might have to increase the size of these logs by a factor of 2 or 3 to accommodate the additional volume of data.
- For VSAM data sources

VSAM logging causes the z/OS System Logger to create offload log data sets. Depending on the RETPD setting and the amount of logging, there might be increased numbers of log data sets with replication logging.

Processing time

CPU time increases measurably, though not necessarily significantly, compared with processing the same workload without change capture. Batch processing produces the most measurable increase.

Target LPAR

Resource consumption is heavier on the target LPAR, because apply processing produces the same DL/I operations that occurred at the source LPAR:

- Increased logging
- Increased MIPS consumption (millions of instructions per second)
- More DASD or tape to store archived log information

Consider dedicating an IMS subsystem (perhaps the entire LPAR) to apply processing. Large numbers of replicated databases schedule multiple apply PSBs (program specification blocks). Assess your need to add larger work pools to accommodate additional PSBs and DMBs (data management blocks).

The target server uses up to 1000 MB of additional above-the-bar memory for dependency analysis related to transaction processing.

Tune the subsystem for the exclusive use of apply processing to avoid negative effects on the performance of applications that use the target IMS subsystem.

Network bandwidth

Data Replication for IMS uses significant network bandwidth and requires TCP/IP links between the source and target LPARs to operate effectively at high speeds. All replicated data and the control messages necessary to maintain replication flow over the network connection.

Service classes

Data Replication for IMS must run in high-priority service classes. Use a workload manager (WLM) to give the Classic data servers the resources that they require to synchronize source and target databases in near-real time. Define service classes and assign the Classic data servers to these classes. Give the Classic data servers a higher priority to allocate sufficient cycles for the workload.

Typically, you give the target engine a higher relative dispatching priority than the source server so that it can offload work as fast as (or faster than) the source server sends it.

Memory

Both the source and target data servers consume system memory.

Source servers have a capture cache per subscription that can require up to 2 GB of memory, especially as the number of subscriptions increase.

Subscription processing

Subscriptions use two caches that you configure independently in the Subscription wizard to accommodate differences in the speed of the source

and target servers. These subscription-level caches can improve performance when replication errors require your Classic data servers to catch up to current processing.

Tasks overview

To set up and maintain Data Replication for IMS, you must perform installation, configuration, and administration tasks.

Installation tasks

Identify the appropriate user roles at your site for installing Data Replication for IMS, which consists of installing mainframe software and a graphical user interface (GUI), preparing the installation environment, and customizing the installation.

Typical user roles for installation tasks

Installer

Some mainframe sites have a person in a dedicated installer role to install the mainframe and distributed software. The installer usually performs the SMP/E installation steps described in the Program Directory. Other duties might include preparing the installation environment and customizing the installation.

System programmer

The system programmer performs installations and configurations and secures permissions to allocate and use resources.

In addition to these duties, the system programmer might also perform installation tasks that you typically associate with installers. In particular, the system programmer resolves system-level problems.

Configuration tasks

Identify the appropriate user roles at your site for configuring Data Replication for IMS. Configuration consists of preparing IMS subsystems, deploying and securing data servers, designing and creating subscriptions, and performing an initial load of the target IMS databases.

Typical user roles for configuration tasks

System programmer

The system programmer performs installations and configurations and secures permissions to allocate and use resources.

The responsibilities of the system programmer might overlap with the installer role, including preparing and customizing the installation. The system programmer typically configures services and troubleshoots problems with the Classic data server.

System administrator

The system administrator manages hardware and physical resources, such as port numbers, TCP/IP connections, and storage, which can include memory or direct access storage devices (DASD).

The system administrator role often overlaps with that of the system programmer. At some mainframe sites, the system administrator also performs the duties of the security administrator.

Security administrator

The security administrator manages security for z/OS systems and applications.

The security administrator role in configuring Data Replication for IMS includes creating security classes and profiles in your external security manager (ESM) and configuring service-level parameters in the Classic data server.

Database administrator (DBA)

The DBA understands the data, data structures, and metadata. This person does day-to-day maintenance of the database, including administration and reorganization, such as restructuring and cleanup operations.

The DBA role in configuring Data Replication for IMS is to set up the data source as required, such as augmenting database descriptions (DBDs) or deploying user exits. The DBA maps relational tables in the Classic data server to the non-relational data in the source database, and maintains matching data structures and data between the source and target databases.

Typically, the DBA loads target databases before replication begins or continues. If the structure of a source database changes in a way that affects replication, the DBA reinstalls the new structures at the target database.

Application programmer

The application designer and application programmer (or *application developer*) design, build, test, and deliver mainframe applications for end users and customers. Based on requirements gathered from business analysts and end users, the designer creates a design specification from which the programmer constructs an application. The process includes several iterations of code changes and compilation, application builds, and unit testing.

The application programmer has an interest in all tasks that affect applications, such as user exits, loading target databases, workload issues, or subscription design.

Administration tasks

Identify the appropriate user roles at your site for administering Data Replication for IMS. Administration includes ongoing tasks such as starting and stopping replication for subscriptions, loading or installing target IMS databases, and monitoring statistics, throughput, and latency.

Typical user roles for administration tasks

System operator

The system operator monitors and controls hardware and software operations. This person starts and stops tasks, monitors the console for unusual conditions, and looks after the health and normal operation of systems. The system operator uses run books associated with applications to identify console messages that require operator intervention, identify specific operator responses, and modify job flows to comply with business requirements.

The system operator typically monitors replication. This person can do this by using a monitoring tool with a graphical user interface (GUI), such as the Classic Data Architect. The system operator might also monitor the job, diagnostic, and event logs for the Classic data server to review and interpret messages that describe server events, activities, and statuses.

System programmer

The system programmer performs installations and configurations and secures permissions to allocate and use resources.

The system programmer typically configures services and troubleshoots problems with the Classic data server. This person might work with other experts, such as DBAs or system programmers, to create and modify subscriptions.

Database administrator (DBA)

The DBA understands the data, data structures, and metadata. This person does day-to-day maintenance of the databases, including administration and reorganization, such as starting and stopping replication and restructuring and cleanup operations.

In addition to the ongoing administrative tasks related to the source and target databases, the DBA might release enqueues or other holds on log files at the data source for maintenance, or load target data or reinstall target data structures.

Classic data servers

A Classic data server runs in its own address space. The Classic data server performs all data access.

A source server is a Classic data server that reads replication records generated by source IMS databases and sends them to a target server that runs on a separate z/OS image. The target server is a Classic data server that applies the changes to the target IMS databases.

Classic data servers perform the following functions:

- Determine the type of data to access
- Maintain the environment
- Process changed data
- Administer and monitor replication

The architecture of a Classic data server is service-based. The Classic data server consists of several components, or *services*.

Services and their functions

When a Classic data server is created during the installation customization process, the services required for the Classic data server are pre-configured.

Configuration definitions include the following server-wide and individual service categories:

- Server-wide, or global, definitions that affect all services within the Classic data server
- Service definitions specific to Data Replication for IMS that consist of unique configuration information that affects each service individually.

Each service is a member of a service class. Services also have a service name and a task name.

service class

The type of service, such as the capture service (CAP) or the administration service (PAA).

service name

The unique name that references a specific instance of a service in a Classic data server.

task name

The load module that is associated with services of a service class.

A service class contains a specific set of configuration parameters. The values of the configuration parameters define a service instance and the behavior of that service.

Critical services

A *critical service* is a service that is critical to the operation of the Classic data server. The Classic data server cannot continue running when one or more services that are critical to the operation of the server are stopped or stop abnormally.

The following services are critical services:

Administration service

The administration service manages activities such as creating and updating replication metadata and processing client requests.

Apply service

The apply service runs in the target server. This service schedules the first apply PSB for each subscription and applies changes to the target database.

Capture service

The capture service runs in the source server. This service manages the source side of replication, including the log reader service and change streams.

IMS log reader service

The IMS log reader service reads IMS log files and sends change messages to the capture service.

Logger service

The logger service receives messages from all services in the Classic data server and coordinates writing the messages to common logs.

You cannot stop a critical service. If you attempt to stop a critical service by issuing a STOP,SERVICE command or a STOP,TASKID command, a warning message is issued.

Detailed information describes each service that runs in the Classic data server, each service configuration parameter, and configuration methods for administrating the configurations for the Classic data server.

Optimizing memory consumption for a Classic data server (guidelines)

To optimize memory consumption, estimate initial memory settings in the job control language (JCL) for your Classic data server and then evaluate them in a test environment.

Procedure

1. Estimate initial values for **REGION** in the JCL for the Classic data server and for the **MESSAGEPOOLSIZE** configuration parameter.
   - For smaller environments, try the default values for the Classic data server:
     – MEMLIMIT=NOLIMIT (target server only)
     – REGION=96MB
     – MESSAGEPOOLSIZE=64MB
   - Consider larger values for **REGION** and **MESSAGEPOOLSIZE** for larger deployments that require more resources. For example, you can begin with values that are larger than needed at first. Then you can define your environment and work toward reducing these values by monitoring the environment and running reports such as the output from the DISPLAY,MEMORY command.

     Consider factors that contribute to resource consumption:
     – Fixed overhead per Classic data server outside message pool storage:
       - C runtime library functions (LE)
       - Database connections
       - Added threads
       - Above-the-bar memory for dependency analysis (target server only)
     – The number of subscriptions, and related metadata:
       - Subscriptions
       - Replication mappings
       - Replication objects
       - For IMS data sources, the number of segments in the database description (DBD)
2. Experiment with different configurations in a test environment to verify that your Classic data server has sufficient resources for the size of your environment.
   a. Specify a region size that is at least 8 MB lower than the site limit, and use the greater of these values:
      - 8 MB higher than the message pool
      - 20% higher than the message pool

      If the 8 MB gap between the region and the message pool is still not sufficient, increase this difference in increments of 8 MB.
   b. Set the **MESSAGEPOOLSIZE** parameter to the greater of these values:
      - 20% less than the region size
      - 8 MB below the **REGION** value or 8 MB below any site limit imposed by exits.

      If you increase the value of the **MESSAGEPOOLSIZE** parameter, set the region size higher to maintain the 8 MB gap.
3. If possible, set the maximum cache size of 3 GB per subscription for the source and target server.

Caches on the source and target server use storage areas outside of the message pool storage, but those storage areas can also be a factor in estimating region size.

The size of the capture cache that you set with the CAPTURECACHE parameter represents the maximum size of one of the storage areas associated with the capture cache. A capture cache consists of multiple storage areas. As a result, the maximum storage that a capture cache uses is approximately 1.5 times the specified value. For example, if you specify 2048M as the capture cache size of a subscription, in a situation where the capture cache fills, the capture cache will occupy approximately 3072 megabytes of storage.

Memory consumption on a Classic data server

If default settings on the Classic data server are insufficient for the size of your environment, you might have to evaluate different configurations in a test environment.

Above-the-bar (64-bit) memory

The target server uses above-the-bar memory for dependency analysis, and the source server does not use above-the-bar memory at all. The **MEMLIMIT** parameter in the job control language (JCL) for the target server specifies the maximum amount of above-the-bar memory that the server can allocate. When you specify this parameter on the JOB statement in the JCL, the setting applies to all steps of the job and overrides any **MEMLIMIT** value in an EXEC statement in JCL.

These examples show how the **MEMLIMIT** parameter works:

MEMLIMIT=NOLIMIT

If `MEMLIMIT = NOLIMIT` (the default), and site-imposed restrictions are in effect, the Classic data server cannot allocate more resources than the site limit allows.

Even when `MEMLIMIT=NOLIMIT` or you specify `REGION=0M`, site-imposed restrictions such as those that the IEFUSI installation exit imposes can limit the virtual storage resources that the data server can allocate.

No MEMLIMIT value

If you do not specify a **MEMLIMIT** value, System Management Facilities (SMF) provides a default value. The IEFUSI installation exit can override any value that the JCL or SMF specifies.

REGION=0 with no MEMLIMIT value

If you specify `REGION=0M` without a **MEMLIMIT** value, it is the same as specifying `MEMLIMIT=NOLIMIT`.

MEMLIMIT=0M

Unlike the **REGION** parameter, a specification of `MEMLIMIT=0M` means that the step cannot use virtual storage above the bar.

For more information, see the z/OS documentation for **MEMLIMIT** and the z/OS installation exits IEFUSI and IEALIMIT.

REGION and MESSAGEPOOLSIZE

You specify a **REGION** size in the JCL for the Classic data server to define the maximum amount of memory that the server can allocate for 24-bit

(below-the-line) and 31-bit (above-the-line) addresses. **MESSAGEPOOLSIZE** is a global configuration parameter for the Classic data server that defines the amount of storage that the Classic data server reserves and manages for its application requirements.

Site limits can restrict region size. If your site specifies a limit, the Classic data server cannot allocate more, regardless of the **REGION** setting in the JCL. When a Classic data server initializes, it allocates storage for **MESSAGEPOOLSIZE** memory first. The Classic data server cannot access more message pool storage than the limit that you set for **REGION**, regardless of the value of the **MESSAGEPOOLSIZE** parameter.

Synchronization of source and target IMS databases

Because Data Replication for IMS sends only change data to the target database, you must begin replication with matching copies of your source and target databases.

Data Replication for IMS does not provide support for initially populating your target databases. You can use various IBM or third party utilities that allow you to unload the source database and reload its contents at the target site, once you transfer the data from the source site to the target site. Additional setup steps might also be required to define the databases at the target site depending upon whether you already have a mechanism in place to support a secondary site or you are starting from scratch.

Data Replication for IMS sends only changed data to a target server to be applied to your target databases by reading and merging data capture log records produced by IMS at your source site. These data capture log records identify:

- The database and segment that was updated by one of your source applications.
- The operation the application performed (insert, update, or delete).
- Information about the segments that were updated by the application.

For replication to work properly, a general requirement is that data updated at the source must exist at the target. While you can enable adaptive apply for one or more of your databases to allow data inconsistencies between your source and target databases, updates to data missing or out-of-sync at the target will not be applied and therefore will remain out-of-sync until steps are taken to resynchronize this data.

In addition to using data capture log records, Data Replication for IMS also captures and analyzes other kinds of log records produced by IMS to determine whether a unit of recovery (UOR) was committed or rolled back. Using this information Data Replication for IMS can keep your source and target databases synchronized and mirrors the actions performed by the source IMSs. For example, if your source BMP or online transaction abnormally terminates, the source IMS automatically backs out those changes and produces additional IMS log records indicating that the UOR failed. Data Replication for IMS intercepts those log records and does not forward the associated changes to the target since those updates were never hardened to the source databases.

A possible exception to this rule is when a DL/I batch job abnormally terminates. In these cases, Data Replication for IMS determines that the last (and possibly only) UOR produced by the failing batch job should not be applied at the target but some or all of the changes were applied to the source databases. To ensure that source and target databases match after a DL/I batch job abends, you must run the

IMS Batch Backout Utility (DFSBBO00) in default mode before making any further changes to the source database. If a batch backout is not performed, you must reload the target database before restarting replication.

To ensure that replication is successful, Data Replication for IMS requires, by default, that the source and target DBD structures are identical based on key attributes. These attributes are identified in the topic "DBD validation" on page 120. However, you can relax this requirement and replicate data provided that the source and target database are compatible. This capability is provided to give flexibility while upgrading databases and applications to support new business requirements and can actually be used to assist in this process.

Regardless of whether the source and target database structures and contents are identical, certain database structures and options are problematic from a replication perspective. The topic "Discrepancies between source and target databases: scenarios" on page 24 provides detailed information.

Modifying source and target databases

You can follow the guidelines described in this topic to modify source and target databases.

Installing or reinstalling the data structures

To install or reinstall the structure, you copy the source member for your database descriptions (DBDs) to the target subsystem and perform the online change process.

Conditions under which you install or reinstall a target database include the following ones:

- You set up Data Replication for IMS for the first time.
- You add a new database to a subscription.
- You change the source database that prevents replication from continuing. See the list below for the types of changes that require a database to be reinstalled.

Loading or reloading the data

You load the data by taking an image copy of each source database, transferring the image copies to the target subsystem, and loading the target databases.

Conditions under which you load or reload a target database include the following ones:

- You encounter errors or inconsistencies in the target database, such as missing parent segments
- You restore a source database to a prior version.
- A DL/I batch job abends and you do not run the IMS Batch Backout Utility (DFSBBO00) in default mode.
- Mass updates occurred at the source that change most or all records in the database
- Replication has been inactive for a subscription, and reloading the data will take less time than replicating the historical changes, or source logs necessary to apply historical changes are no longer available.

Upgrading database structures

Assuming you have an existing replication environment setup, you can

make any of the following changes to a target database definition and still expect replication to continue operating successfully:

- Increase the maximum size of a variable length segment
- Add new subset pointers
- Add new child segments as leaf segments in the hierarchy

For replication to function properly, the following kinds of changes cannot be made to the target database:

- Deletion of a segment that is actually populated by source applications
- Changes to existing parent-child relationships
- Changing the size of a fixed length segment
- Changing sequence field information
- Changing insert rules
- Deletion of subset pointers

By default replication is not allowed to start for a subscription if any of the key DBD attributes are different between the source and target definitions. Setting the apply service STRICTVALIDATION configuration parameter to FALSE allows replication to start for a subscription when the DBD attributes are different. You should only enable this configuration setting when the source and target databases are compatible and only during the period of time when the source and target database definitions are out of synchronization.

Disabling STRICTVALIDATION assumes that while database definitions have differences, these differences are not expected to cause problems during replication. The specific differences are not analyzed by the replication software in this mode, so it is the responsibility of the user to ensure that differences will not cause replication problems.

The general procedure for making compatible structural changes to a target database without upgrading the source database at the same time is as follows:

1. Perform a controlled end of replication for the subscription that contains the database being changed.
2. When replication stops for the subscription, take the database offline and make the required changes.
3. Start the database after it is reloaded with the new structure and the online change is completed.
4. Set STRICTVALIDATION to FALSE.
5. Restart replication for the subscription.

Changes that were made at the source while the target was offline will now be replicated using the new target definition.

If you are operating an Active-Standby environment, the general procedure for upgrading the source database structure after a target changes is as follows:

1. Allow replication to run long enough for the changed target database to synchronize with the source.
2. Quiesce update activity to the source database.
3. Capture any trailing changes to the source database.
4. Perform a controlled end of replication.

5. Follow fail over procedures to switch production applications to the target site.
6. Upgrade the source database structures to be completely compatible with target databases.
7. Set STRICTVALIDATION to TRUE for future source to target replication.
8. Follow fail back procedures to synchronize target databases with source databases and restart replication between the source and target.

Data Replication for IMS uses a hashing method to determine whether the source and target DBD attributes match. When the hash values between the source and target databases do not match, the software is not aware of the specific differences between the databases and whether the differences will cause problems for replication. Using the default STRICTVALIDATION behavior immediately notifies you when differences exist and prevents replication from starting. This behavior reduces the risk of target database corruption or replication failures when differences in DBD attributes do exist. Therefore, you should always set STRICTVALIDATION to TRUE unless a specific database modification is needed at the target.

Discrepancies between source and target databases: scenarios

In Data Replication for IMS, the source and target databases might not match exactly under certain conditions.

For example, IMS databases manage many types of information, some of which your application might not require, such as audit information or temporary records. Data Replication for IMS does not examine, verify, or validate database partitioning schemes, DASD allocations, secondary indexes, or the number of areas in the database.

Some types of IMS databases have structures that can affect the match between the source and target database, such as the following structures:

Sequential dependent segments in DEDB databases

You cannot capture delete operations that the DEDB Sequential Dependent Delete Utility performs.

Non-keyed or non-unique keyed segments

The capture service sends before-image information for a segment when that segment does not have a unique concatenated key. If the before image contains unique qualifying information, the apply service can write the changes to the correct location in the target database. Otherwise, the target server applies the change to the first instance of a matching segment that it encounters.

When Data Replication for IMS replicates databases that use an insert rule of HERE, whether or not you are running IMS version 13 determines if apply processing can insert the segments in the same location where they occurred in the source database.

- If you have databases that use an insert rule of HERE and your source site is running IMS version 13, you can apply maintenance and augment your source database to activate INPOS processing. This processing

includes additional information in the data capture log records that can assist in reliably replicating inserts for these kinds of segments.

These enhanced data capture log records include information about whether the source application used a first or last command code to perform the insert. Otherwise the log records include segment data for the existing twin that the new instance was inserted before. With this data, and assuming the positioning segment data is unique, the target server is able to insert the new segment data at the same location at the target.

- If the IMS version 13 maintenance is not installed, the source database has not been augmented to capture INPOS data, or the positioning data is not unique, the source and target database will not match for segments that use an insert rule of HERE.

Logical relationships

The subscription wizards in the Classic Data Architect manage these restrictions for you automatically by including all logically-related databases in the subscription. Administrative actions on one of the logically-related databases affect all of them.

Replication monitoring

You can monitor replication metrics for your subscriptions, including latency, cache consumption, and throughput. An Event Log view enables you to monitor replication events.

You can monitor replication metrics and events by using the Classic Data Architect, which provides both numeric and visual data. The following views help you to manage your replication environment in real time:

- Show Latency
- Show Cache Meters
- Show Throughput

Latency

Latency statistics report the time interval that elapsed between the completion of the last unit of recovery (UOR) at the source and the completion of the corresponding UOR at the target.

Caches

Cache meters provide a visual display of the amount of storage that your Classic data servers are using to store change messages and UORs.

Throughput

Throughput statistics report the volume of work that your replication environment is processing.

The Classic Data Architect also provides an Event Log view to monitor events in real time or review historical events:

- All events in a single view
- Events for a selected subscription
- Events for all connected servers

You can also monitor subscription information by using MTO commands.

Chapter 2. Installing Data Replication for IMS

Installing Data Replication for IMS consists of installing mainframe components and the Classic Data Architect, preparing your installation environment, and customizing the installation to create a functional runtime environment.

The following table lists the major tasks required for Data Replication for IMS installation with a link to where to find information about each task. Perform the tasks in the following recommended order.

Task		Reference
1	Perform the SMP/E tasks to install the components required for Data Replication for IMS on the mainframe.	"Installing Data Replication for IMS on the mainframe" on page 31
2	Prepare for the installation customization process by completing the tasks in the checklist for setting up the installation environment. Obtaining items such as required authorizations and port numbers will prepare you for the customization procedure.	"Setting up the installation environment" on page 31
3	Customize the installation environment by completing the tasks in the installation customization process for the type of Classic data server that you want to customize.	"Installation customization process" on page 32
4	Install the Classic Data Architect client application to manage server connections, tables, views and subscriptions, monitor metrics, and perform configuration tasks.	"Installing the Classic Data Architect" on page 58

System requirements (Data Replication for IMS)

Before you install IBM InfoSphere Data Replication for IMS for z/OS, ensure that your system meets all of the system prerequisites and requirements.

Prerequisites

Replication from IMS data sources

- IBM z/OS Version 1.13 or later
- IBM IMS Version 12.1 or later
- The source and target sites must support the same level of Enterprise Systems Architecture (ESA) compression services
- APAR OA39035, PTF UA65302 on z/OS version 1.12
- APAR OA39035, PTF UA65303 on z/OS version 1.13

Supported IMS databases

Data Replication for IMS supports most full-function IMS databases, including these database types:

- Direct entry (DEDB)
- High availability large (HALDB)
- Hierarchical direct access method (HDAM)
- Hierarchical indexed direct access method (HIDAM)

DL/I batch jobs

You can capture changes that DL/I batch jobs make to IMS databases only under the following conditions:

- The batch job must run with `DBRC=Y`.
- The DFSFLGX0 exit must be in the STEPLIB of the batch job.
- IEFRDER must point to a permanent dataset, therefore cannot be `DUMMY`.

Other requirements

- The source and target sites must use the same z/OS code page.
- You must register subsystems whose changes you want to capture in a shared set of RECON data sets.
- You must catalog all IMS logs that you want to capture in a shared integrated catalog facility (ICF).
- DASD-based logs must reside on shared DASD that any image in the source sysplex can access.

Restrictions

Before you install IBM InfoSphere Data Replication for IMS for z/OS, ensure that your data sources qualify for replication.

Restrictions for IMS data sources

Unsupported database types

Data Replication for IMS does not support the following IMS database types:

- GSAM (generalized sequential access method)
- HSAM (hierarchical sequential access method)
- MSDB (main storage databases)

Certain logically-related databases

Your environment replicates databases that participate in logical relationships as a single entity. Logically-related databases with multiple access paths can create conflicting conditions that make these databases ineligible for replication. The following criteria for EXIT parameters, logical relationships, and delete rules, when you consider them together, create the conflict:

- All segments must have EXIT parameters.
- Logical child segments with EXIT coding *must* have a virtual delete rule.
- Logical parent segments with EXIT coding *cannot* have a virtual delete rule.

If a segment in a logically-related database is both a logical parent and a logical child at the same time, that database and any other databases that are logically related to it are ineligible for replication.

Unsupported subsystem and application types

You cannot capture changes from IMS subsystems that operate in an Extended Recovery Facility (XRF) complex.

Space allocation for formatted online log data sets

On the source IMS site, before using any newly allocated online log data set (OLDS), you must ensure that the space occupied by the OLDS is filled to capacity by legitimate IMS log records at least once. This requirement is more stringent than the IMS condition which strongly suggests that space used by an OLDS allocated on newly initialized or re-initialized volumes should first be formatted for performance reasons.

Low-performing database types

The following database types can make replication slow or unreliable:

Logically-related databases

Databases that participate in logical relationships must perform serial apply, which inhibits performance.

Databases with segments that have no key, or lack a unique key

Data Replication for IMS cannot accurately replicate databases with segments that do not contain a unique concatenated key and data contents. In such cases, the source and target databases might not match.

The capture service sends before-image information for a segment without a unique sequence field. If the before image contains unique qualifying information, the apply service can locate the correct instance in the target database. Otherwise, the target server applies the change to the first instance of a matching segment that it encounters.

Databases with an insert rule of HERE

If you have databases that use an insert rule of HERE and your source site is running IMS version 13, you can apply APAR PI07353 and augment your source database to activate INPOS processing. This processing includes additional information in the data capture log records that can assist in reliably replicating inserts for these kinds of segments.

With additional positioning information available, the same rule stated above applies. If the positioning data identifies a unique instance, the data is inserted at the proper location at the target. Otherwise the data is inserted before the first instance of the positioning data.

If a prior version of IMS is installed at the source, or if INPOS processing is not enabled, it is probable that the source and target databases will not match for segments that use an insert rule of HERE.

Databases with sequential dependent segments

You cannot capture delete operations that the Sequential Dependent Delete Utility performs on a direct entry database (DEDB). If the majority of the databases at your site have sequential dependent segments (SDEPs) and most applications perform inserts to an SDEP, these databases must also perform serial apply.

Additional restrictions:

- Subscriptions that contain databases that participate in logical relationships only support use of a single apply PSB. Therefore these kinds of subscriptions do not support parallel apply operations.
- The source server analyzes the databases and segments that are referenced in each source UOR and identifies whether a UOR can be applied in parallel. A source UOR that only contains references to segments with unique sequence fields, and does not contain any updates to subset pointers, is classified as being eligible to be applied in parallel. Any source UOR that contains operations that update a subset pointer, or reference a segment without a unique sequence field, are identified as requiring serialization.
- If you enable completed UOR tracking at the target server, a UOR that is classified as requiring serial processing is subject to completed UOR tracking. This allows that UOR to be applied in parallel. However, before the changes are committed to the target, the bookmark database is updated for the subscription to record the fact that a UOR requiring serialization was successfully applied at the target.

Consider these additional restrictions when you implement Data Replication for IMS:

- Data Replication for IMS does not replicate data that you restore from image copy.
- Data Replication for IMS does not replicate data that you load into the database.
- Data Replication for IMS does not support Automated Recovery Management (ARM) features that help you to recover from address space failures, image failures, or site outages.
- Replication of updates made by applications using FLD calls requires:
  - IMS version 13 installed at the source site with APAR PI07355 applied.
  - The EXIT statement on the source database augmented to include the FLD option.

  Sites running prior versions of IMS, or that do not have the required IMS maintenance installed and the database properly augmented, will not capture changes made by applications that use FLD calls to update an IMS database.
- Replication of subset pointer updates requires that IMS version 13 is installed at the source site with APAR PI07353 applied. Capture of subset pointer changes also requires that you augment the EXIT statements, for the source databases that contain subset pointers, to include the SSPCMD option.
- Data Replication for IMS does not capture changes to primary and secondary indexes. However, the target IMS subsystem maintains the primary and secondary indexes automatically as the target server applies changes to target databases.

Installing Data Replication for IMS on the mainframe

The IBM InfoSphere IMS Replication for z/OS product is included on tape and the installation instructions are provided in the product program directory.

The program directory details the system requirements and provides installation instructions that supplement the installation information in the product library. See InfoSphere Data Replication for IMS for z/OS Program Directory V11.3.

Setting up the installation environment

After you complete the mainframe SMP/E installation, the next step in the installation process is to set up the installation environment. Setting up the installation environment is a prerequisite to the installation customization process.

The following table provides a checklist of tasks needed to set up the installation environment for source servers and target servers.

Table 1. Checklist of installation environment setup for source and target servers

Task	Server type	Reference
Obtain APF library authorizations for the installation load library SCACLOAD	Both	Obtaining library authorizations for the authorized program facility (APF)
Assign port numbers for communication for Classic data servers and log reader notification.	Source	Obtaining ports for replication
Assign port numbers for communication for Classic data servers.	Target	
Set up resources profiles and security classes for security for Classic data servers.	Both	Securing a Classic data server
Ensure that you have the authorization required to run the Administrative Data Utility (IXCMIAPU). You need this authorization before you run the utility to define the Classic event log and the diagnostic z/OS log streams.	Both	Administrative Data Utility
Augment DBDs for change data capture.	Source	Augmenting DBDs for change data capture
Create an IMS DRA startup table (DFSPZPxx load module), which is required for the DRA service.	Target	Setting up the database resource adapter (DRA) for Data Replication for IMS
Define the bookmark database in the target IMS. This includes identifying the new resource to IMS, registering the database, and database initialization.	Target	Creating a bookmark database
Define the apply PSB in the target IMS, once per subscription.	Target	Defining an apply PSB for a subscription

Customizing the installation environment

The goal of the installation customization process is to simplify the setup of your runtime environment by providing a central place for you to specify the site-specific information that is needed to configure your environment.

The information that you provide is then used as input for generating all JCL and configuration data needed to build the runtime environment.

Installation customization process

Installation customization is a process that allows you to provide setup and configuration information to create a customized installation environment for a source server, a target server, or both.

The installation customization process involves a set of steps that you perform after you complete the mainframe SMP/E installation. You provide setup and configuration information that is used to generate all of the sample JCL and configuration members in the *USERHLQ*.USERSAMP data set that require edits. You then run installation customization jobs that are generated based on the parameters that you specify to create a customized installation environment.

The installation customization process is based on the role of the Classic data server. The possible roles for a Classic data server for Data Replication for IMS are the source server or the target server. You can customize an installation environment for a source server, a target server, or both a source and a target server. You specify the role of a Classic data server with the SERVERROLE parameter for the installation customization utilities. This parameter controls the installation components that you customize. You create installation data sets (*USERHLQ*.USERSAMP and *USERHLQ*.USERCONF) that contain the required components for the type of installation that you choose, and you customize only the parameters needed for that environment.

You can set up one or more Classic data servers on the same LPAR. When you complete the installation customization process, an operational environment is established that you can build upon as needed. The environment includes a functional Classic data server and all of the services required for the specified role. All services are pre-configured during the customization process.

Overview of installation customization procedure

The installation customization process consists of the following basic steps:

1. The user samples allocation utility creates a working set of the SCACSAMP and SCACCONF data sets that contain all customized JCL and configuration members. This working set is referred to as the *USERHLQ*.USERSAMP and *USERHLQ*.USERCONF data sets.
2. You gather site-specific configuration information needed to customize the environment and enter that information in the customization parameters file.
3. The installation customization utility generates customized JCL and configuration members and stores them in the *USERHLQ*.USERSAMP and *USERHLQ*.USERCONF data sets that were created in the first step.
4. You allocate and initialize the following components by using generated customization jobs:
   - z/OS log streams
   - Configuration files

- Data sets to store your subscriptions and replication mappings

5. You use the generated JCL and configuration members to start the runtime environment.

Installation customization components

The following table lists the components and sample JCL members that you use during the installation customization process.

Table 2. Summary of installation customization components distributed in the SCACSAMP data set.

Component name	Description
User samples allocation utility	Allocates the *USERHLQ*.USERSAMP and *USERHLQ*.USERCONF data sets. Populates the *USERHLQ*.USERSAMP data set with a copy of the customization parameters file (CECCUSPI) and the installation customization utility JCL (CECCUSI2). The SCACSAMP(CECCUSI1) JCL runs this utility. SCACSAMP(CECCUSI1) is the JCL that runs the user samples allocation utility. CECCUSI1 is the only member in the distributed SCACSAMP data set that you edit. The JCL comments provide editing instructions.
Installation customization utility	Reads the customization parameters file *USERHLQ*.USERSAMP(CECCUSPI) and generates the necessary JCL and configuration members in the *USERHLQ*.USERSAMP and *USERHLQ*.USERCONF partitioned data sets. The *USERHLQ*.USERSAMP(CECCUSI2) JCL runs this utility. *USERHLQ*.USERSAMP(CECCUSI2) is the generated JCL that submits the installation customization utility and generates all necessary JCL and configuration members.
Customization parameters file	Contains the installation and customization information that you specify in the form of parameter and value pairs to complete an installation and establish an initial functioning environment. This file is located in *USERHLQ*.USERSAMP(CECCUSPI).
The following components are specific to the server role for the source server or the target server.	
USERHLQ.USERSAMP(CECISSLS)	Generated JCL that runs the Administrative Data Utility (IXCMIAPU) to define the z/OS event log stream and a log stream for the diagnostic log for the source server. For VSAM data sources, this member also creates a simple replication log for the IVP VSAM file when CDCRLGST is specified.
USERHLQ.USERSAMP(CECITSLS)	Generated JCL that runs the Administrative Data Utility (IXCMIAPU) to define the z/OS event log stream and a log stream for the diagnostic log for the target server.
USERHLQ.USERSAMP(CECISCFG)	Generated JCL that allocates and initializes the configuration files for the source server.
USERHLQ.USERSAMP(CECITCFG)	Generated JCL that allocates and initializes the configuration files for the target server.
USERHLQ.USERSAMP(CECISSUB)	Generated JCL that allocates and initializes the VSAM data sets for the source server that store metadata for subscriptions and replication mappings.
USERHLQ.USERSAMP(CECITSUB)	Generated JCL that allocates and initializes the VSAM data sets for the target server that store metadata for subscriptions and replication mappings.
USERHLQ.USERSAMP(CECIMSSC)	Generated JCL to start the source server.
USERHLQ.USERSAMP(CECIMSTG)	Generated JCL to start the target server.

User samples allocation utility

The user samples allocation utility allocates the *USERHLQ*.USERSAMP and *USERHLQ*.USERCONF data sets and populates the *USERHLQ*.USERSAMP data set with a copy of the customization parameters file and the installation customization utility.

The user samples allocation utility performs these functions:

- Allocates the *USERHLQ*.USERSAMP and *USERHLQ*.USERCONF data sets. These data sets are created with the same characteristics as the distributed SCACSAMP and SCACCONF data sets.

  If you run the utility again, the *USERHLQ*.USERSAMP or *USERHLQ*.USERCONF data sets that already exist are reused. The utility replaces the customization parameters file and customization utility JCL members. All other members remain the same.
- Generates the customization parameters file *USERHLQ*.USERSAMP(CECCUSPI) and the installation customization utility JCL *USERHLQ*.USERSAMP(CECCUSI2). All input parameters specified for the samples allocation utility are populated in the generated CECCUSPI and CECCUSI2 members.

You use the SCACSAMP(CECCUSI1) job to run the allocation utility. The JCL contains comments with editing instructions. You specify the following input as parameters:

CACINHLQ=CEC.V11R3M00
: The value specified for the CACINHLQ keyword must be the high-level qualifier of the installation data sets that the SMP/E installation produces.

CACUSHLQ=USER.V11R3M00
: The value specified for the CACUSHLQ keyword must be the high-level qualifier for the *USERHLQ*.USERSAMP and *USERHLQ*.USERCONF data sets that the samples allocation utility creates or updates.

CACDUNIT=SYSALLDA
: The value specified for the CACDUNIT keyword identifies the disk unit that is used when allocating the *USERHLQ*.USERSAMP and *USERHLQ*.USERCONF data sets. This is an optional parameter.

CACDVOLM=
: The value specified for the CACDVOLM keyword identifies the disk volume that is used when allocating the *USERHLQ*.USERSAMP and *USERHLQ*.USERCONF data sets. This is an optional parameter.

CACSTGCL=
: The value specified for the CACSTGCL keyword identifies the SMS storage class that is used when allocating the *USERHLQ*.USERSAMP and *USERHLQ*.USERCONF data sets. This is an optional parameter.

CACMGTCL=
: The value specified for the CACMGTCL keyword identifies the SMS management class that is used when allocating the *USERHLQ*.USERSAMP and *USERHLQ*.USERCONF data sets. This is an optional parameter.

ISPFHLQ=ISP
: The value specified for the ISPFHLQ keyword identifies the high-level qualifier for ISPF installation. The samples allocation utility runs a TSO batch application and uses TSO functions.

ISPFLANG=ENU
The value specified for the ISPFLANG keyword identifies the language prefix for the ISPF installation.

SERVERROLE=(*role-name*, ...)
The value of the SERVERROLE keyword specifies that the server environment being installed and customized contains the components required for a Data Replication for IMS source server or target server environment. You can specify one or more roles for your Classic data server. If you specify multiple role names, you must separate the names with commas and enclose the names in parentheses.

IMSR_SOURCE
Specify this value to install the components required for an IMS source server.

IMSR_TARGET
Specify this value to install the components required for an IMS target server.

The samples allocation utility produces a summary report that is written to the SYSTSPRT DD specified in the JCL. The report lists the status for allocating the *USERHLQ*.USERSAMP and *USERHLQ*.USERCONF data sets and lists the members updated in the *USERHLQ*.USERSAMP data set.

The following figure shows sample output written to SYSTSPRT.

```
                    ********** Samples Allocation **********
                                  Summary Report


CACCUSX1 compiled on 2012-09-13 15:31:02 by REXX370 3.48
Execution timestamp: 2012-09-13 15:31:02  MVS Product ID: z/OS 01.10.00  SMF ID: ABC System ID: ABC
-------------------------------------------------------------------------------------------------


 Effective Parameters:

   CACDUNIT:    SYSALLDA
   CACDVOLM:
   CACINHLQ:    CEC.V11R3M00
   CACMGTCL:
   CACSTGCL:
   CACUSHLQ:    USER.V11R3M00.IMSSRC
   ISPFHLQ:     ISP
   ISPFLANG:    ENU
   SERVERROLE:  IMSR_SOURCE

Member     Status
--------   --------
CECCUSI2   Processed successfully in DDN(USERSAMP)
CECCUSPI   Processed successfully in DDN(USERSAMP)


Summary:
   Members Successful:        2
   Members in Error:          0
   Members Not Replaced:      0
   Members Processed:         2


Return Status: 0
```

Figure 5. Sample output for the samples allocation summary report

Installation customization utility

The installation customization utility generates the JCL and configuration members needed in the *USERHLQ*.USERSAMP and *USERHLQ*.USERCONF data sets based on the values that you provide in the customization parameters file.

The installation customization utility performs these functions:

- Captures the customization settings that you provide in the customization parameters file *USERHLQ*.USERSAMP(CECCUSPI).
- Applies the customization parameters to all JCL members associated with the specified SERVERROLE parameter and places the customized members in the *USERHLQ*.USERSAMP data set.
- Applies the customization parameters to all configuration members associated with the specified SERVERROLE parameter and places the customized members in the *USERHLQ*.USERCONF data set.

You use the *USERHLQ*.USERSAMP(CECCUSI2) job to run the installation customization utility. You specify the following input as parameters:

CACINHLQ=CEC.V11R3M00
: The value specified for the CACINHLQ keyword must be the high-level qualifier for Classic distribution data sets produced by the SMP/E

installation. This value is automatically populated with the value previously specified as input to the user samples allocation utility.

CACUSHLQ=USER.V11R3M00
The value specified for the CACUSHLQ keyword must be the high-level qualifier for the *USERHLQ*.USERSAMP and *USERHLQ*.USERCONF data sets that were created or updated by the user samples allocation utility. This value is automatically populated with the value previously specified as input to the user samples allocation utility.

MEMBER=(*member-name***, ...)**
This is an optional parameter. The value specified for the MEMBER keyword identifies a list of one or more member names to process. Only a subset of the members associated with the specified SERVERROLE parameter is processed. If you specify multiple member names, you must separate the names with commas and enclose the names in parentheses.

All members are processed when this parameter is not specified.

OVERWRITE=YES | NO
The value specified for the OVERWRITE keyword indicates how to process existing members of target data sets, for example the *USERHLQ*.USERSAMP and *USERHLQ*.USERCONF data sets.

- When you specify OVERWRITE=NO, existing members of the target data sets are not replaced. OVERWRITE=NO is the default.
- When you specify OVERWRITE=YES, existing members of the target data sets are replaced.

Example: OVERWRITE=NO is in effect. Members CACCFGDS and CACCFGUT already exist in the target data set. Member CACSX04 does not exist in the target data set.

```
Member   Status
-------- --------
CACCFGDS CACCFGDS not replaced in DDN(USERSAMP)
CACCFGUT CACCFGUT not replaced in DDN(USERSAMP)
CACSX04  Processed successfully in DDN(USERSAMP)
```

Example: OVERWRITE=YES is in effect. The processing status for the same members shown in the previous example appear as follows (whether or not any of these members previously existed in the target data set):

```
Member   Status
-------- --------
CACCFGDS Processed successfully in DDN(USERSAMP)
CACCFGUT Processed successfully in DDN(USERSAMP)
CACSX04  Processed successfully in DDN(USERSAMP)
```

The processing summary information produced at the bottom of the report identifies the number of members that were stored successfully and the number of members that were not replaced. For example:

```
Summary:
   Members Successful:       90
   Members in Error:          0
   Members Not Replaced:      0
   Members Processed:        90
```

SERVERROLE=(*role-name***, ...)**
The value of the SERVERROLE keyword specifies that the server environment being installed and customized contains the components required for a Data Replication for IMS source server or target server environment. You can specify one or more roles for your Classic data

server. If you specify multiple role names, you must separate the names with commas and enclose the names in parentheses.

IMSR_SOURCE
Specify this value to install the components required for an IMS source server.

IMSR_TARGET
Specify this value to install the components required for an IMS target server.

The utility produces a summary report that is written to the SYSTSPRT DD that is specified in the JCL. The report lists the partitioned data sets and the data set members that were processed. The final summary lists the total number of members processed, the number that were successful, and the number with errors.

The following figure shows sample output written to SYSTSPRT.

```
********************     Installation Customization    **********************
                             Summary Report

CACCUSX2 compiled on 2012-08-15 08:46:51 by REXXC370 3.48
Execution timestamp: 2012-08-15 08:49:39 MVS Product ID: z/OS 01.10.00 SMF ID: SYE9 System ID:
-------------------------------------------------------------------------------------------

Effective Parameters:

  CACINHLQ:    CEC.V11R3M00
  CACUSHLQ:    USER.V11R3M00.IMSSRC
  OVERWRITE:   No
  SERVERROLE:  IMSR_SOURCE


Processing parameters file: USER.V11R3M00.IMSSRC.USERSAMP  Member: CECCUSPI

Processing Members for Product: All  Role: Common
Member     Status
--------   --------
CACCFGDS   Processed successfully in DDN(USERSAMP)
CACCFGUT   Processed successfully in DDN(USERSAMP)
CACPRTLS   Processed successfully in DDN(USERSAMP)
CACLGFLT   Processed successfully in DDN(USERSAMP)
CACSX04    Processed successfully in DDN(USERSAMP)

Processing Members for Product: Data Replication for IMS for z/OS Role: Common
Member     Status
--------   --------

Processing Members for Product: Data Replication for IMS for z/OS  Role: IMSR_SOURCE
Member     Status
--------   --------
CECISCFG   Processed successfully in DDN(USERSAMP)
CECISSLS   Processed successfully in DDN(USERSAMP)
CECISSUB   Processed successfully in DDN(USERSAMP)
CECIMSSC   Processed successfully in DDN(USERSAMP)
CECISXFG   Processed successfully in DDN(USERCONF)


Summary:
   Members Successful:       10
   Members in Error:          0
   Members Not Replaced:      0
   Members Processed:        10

Return Status: 0
```

Figure 6. Sample output for the installation customization summary report.

Working with the customization parameters file

These guidelines describe how to enter values in the customization parameters file.

The customization parameters file, *USERHLQ*.USERSAMP(CACCUSPI), contains pairs of keyword and value settings used to customize JCL and configuration files in the *USERHLQ*.USERSAMP and *USERHLQ*.USERCONF data sets.

The following sections provide guidelines for entering input into the customization parameters file, describe how the file is organized, and list the keyword and value settings that the customization parameters file contains. Other considerations include the use of job cards, pre-defined variables, and STEPLIB concatenations.

Input guidelines

The following guidelines describe how to enter values in the customization parameters file:

- Keyword and value pairs:
  - You cannot change the keyword component.
  - You must delimit the value component with double quotes ("").
  - Spaces are allowed before and after the keyword and value.
  - Values cannot span multiple lines.
- The minimum required parameters that you must change for a successful installation are denoted by an asterisk within parentheses at the end of the comment for that parameter. For example: CACINHLQ="&CACINHLQ" HLQ of Classic product(*)
- Comments:
  - An asterisk (*) in column 1 defines the line as a comment line.
  - Any input that you include after the first space after the value component is treated as comments.

File organization

The following table describes the organization of the customization parameters file.

Table 3. Organization of the customization parameters file.

Section name	Section content
Common installation section	Parameters that apply to all installations, such as the high-level qualifier for the Classic product installation
Source server parameters section: The customization parameters file contains the following sections for source servers when you specify SERVERROLE=IMSR_SOURCE.	
Source server parameters	Parameters that apply to the source server
Source security parameters	Security parameters that control user connections for the source server
Source log stream parameters	Log stream parameters for the source server
Target server parameters section: The customization parameters file contains the following sections for target servers when you specify SERVERROLE=IMSR_TARGET.	
Target server parameters	Parameters that apply to the target server
Target bookmark parameters	Parameters that define and customize the bookmark database
Target security parameters	Security parameters that control user connections for the target server
Target log stream parameters	Log stream parameters for the target server

Use of job cards

Job card information is defined in the common installation section of the customization parameters file. The following two-line job card information is used as a template when generating JCL members:

```
CACDJOB1="JOB (CLASSIC),'CLASSIC JOB',CLASS=A,"
CACDJOB2="MSGCLASS=X,NOTIFY=&SYSUID"
```

The CACDJOB1 value is placed after the job name in each generated JCL member. The CACDJOB2 value is provided on the second line of the job card in each JCL member.

The initial value for the job card keywords is populated from the job card that is specified on the CECCUSI1 JCL member.

Use of pre-defined variables

Many of the data set values in the customization parameters file contain pre-defined variables such as &CEC to reference previously defined high-level qualifiers. Most of the generated JCL members make use of inline PROC definitions. These variables reference the actual PROC variables. The following table describes what each variable defines:

Table 4. Pre-defined variables.

Variable	Description
&CEC	Classic product installation high-level qualifier
&USERHLQ	User SCACSAMP high-level qualifier
&IMS	IMS installation high-level qualifier

Library concatenations

For Classic data server parameters that require specific DD data set concatenation customization such as STEPLIB, parameters are provided for concatenation. You can specify the same parameter keyword multiple times. The order specified for the parameter keywords is the order in which the data sets will be included in the data set concatenation.

The following example shows how the STEPLIB concatenations are customized for IMS access:

IMS STEPLIB

For IMS, you must define the IMS RESLIB and EXITLIB in the STEPLIB. The following example shows how the IMSSTEPL keyword is used twice to define the STEPLIB concatenation to include the SDFRESL followed by the EXITLIB. You can add additional libraries by specifying more IMSSTEPL settings. Use the &IMS.. prefix to use the IMS installation high-level qualifier. Otherwise, you can specify a full data set name:

```
IMSSTEPL="&IMS..SDFRESL"
IMSSTEPL="&IMS..EXITLIB"
*IMSSTEPL="&IMS..SDFSRES2"
*IMSSTEPL="&IMS..SDFSRES3"
```

Customization parameters file settings:

The parameter keyword and value pairs in the customization parameters file are set to default values. You can modify these values to customize your installation.

Each section of the customization parameters file provides the parameter default values, and a description of each parameter.

Customization parameters file: Common section:

The common section of the customization parameters file contains parameters that are common to all installations.

The following table lists the parameters in the common section of the customization parameters file, the parameter default values, and a description of each parameter.

Table 5. Common parameter and default settings for USERSAMP(CECCUSPI)

Parameters	Default value	Description
Common section		
CACINHLQ	CEC.V11R3M00	High-level qualifier of the installation data sets for the Classic product. This value is populated with the value specified for the CACINHLQ input parameter of the CECCUSI1 job.
CACUSHLQ	USER.V11R3M00	High-level qualifier for the *USERHLQ*.USERSAMP and *USERHLQ*.USERCONF data sets. This value is populated with the value specified for the CACUSHLQ input parameter of the CECCUSI1 job.
CACDUNIT	SYSALLDA	Disk unit that is used for the generated jobs that create data sets such as the configuration, subscription, and replication mapping files. This value is populated with the value specified for the CACDUNIT input parameter of the CECCUSI1 job. If the value is "", it is assumed that the site SMS rules will determine the data set allocation.
CACDVOLM	""	Disk volume that is used for the generated jobs that create data sets such as the configuration, subscription, and replication mapping files. This value is populated with the value specified for the CACDVOLM input parameter of the CECCUSI1 job. If the value is "", it is assumed that the site SMS rules will determine the data set allocation.
CACSTGCL	""	SMS storage class that is used for the generated jobs that create data sets such as the configuration, subscription, and replication mapping files. This value is populated with the value specified for the CACSTGCL input parameter of the CECCUSI1 job. If the value is "", it is assumed that the site SMS rules will determine the data set allocation.
CACMGTCL	""	SMS management class that is used for the generated jobs that create data sets such as the configuration, subscription, and replication mapping files. This value is populated with the value specified for the CACMGTCL input parameter of the CECCUSI1 job. If the value is "", it is assumed that the site SMS rules will determine the data set allocation.

Customization parameters file: Source server section:

The source server section of the customization parameters file contains parameters that are specific to a source server.

The following table lists the parameters in the source server section of the customization parameters file, the parameter default values, and a description of each parameter.

Table 6. Source server parameters and default settings for USERSAMP(CECCUSPI)

Parameters	Default value	Description
Common server parameters for change data capture		

Table 6. Source server parameters and default settings for USERSAMP(CECCUSPI) (continued)

Parameters	Default value	Description
IMCPHLQD	USER.V11R3M00.IMSSRC	High-level qualifier (HLQ) of data sets created for the source server: • Binary configuration data sets • Subscription data set • Replication mapping data set This value replaces the references to &CPHLQ.. in other keyword values such as IMCPCFGD, IMCPCFGX, and IMCPSUBS.
IMCPCFGD	&CPHLQ..CACCFGD	Name of the configuration data file for the source server. This value is prefixed with the IMCPHLQD high-level qualifier.
IMCPCFGX	&CPHLQ..CACCFGX	Name of the configuration index file for the source server. This value is prefixed with the IMCPHLQD high-level qualifier.
IMCPHOST	0.0.0.0	Host name or IP address where the source server will run. This value is used in the definition of the COMMSTRING configuration parameter of the connection handler for the source server.
IMCPPORT	9087	Port number that the source server connection handler service listens on. This listen port communicates with the Classic Data Architect to monitor and manage the source server. This value is used in the definition of the COMMSTRING configuration parameter of the connection handler for the source server.
IMCPSUBS	&CPHLQ..SUB	Data file name for source server subscriptions. This value is prefixed with the IMCPHLQD high-level qualifier.
IMCPRMDS	&CPHLQ..RM	Data file name for source server replication mapping. This value is prefixed with the IMCPHLQD high-level qualifier.
IMCPCDPG	""	Host code page of the system for the source server. This value defines the global HOSTCODEPAGE configuration parameter for the source server.
IMCNMICS	""	Network management interface (NMI) communication string for the source server. Sets the NMICOMMSTRING configuration parameter for the monitoring service.
IMCPIHLQ	IMSCP	High-level qualifier (HLQ) assigned to the IMS libraries that the source server uses to access IMS. This HLQ is applied to the IMS data sets in the STEPLIB sections of the generated JCL. This value replaces the references to &IMS.. in other keyword values such as IMCPSTPL.
IMCPSTPL	&IMS..SDFSRESL	Library concatenation, for example STEPLIB concatenation. You can specify this keyword multiple times. The order of the multiple IMCPSTPL keywords defines the order in which the files are included in the concatenation for the generated JCL members.
IMCPNPRT	5001	Port that the log reader service listens on to receive notifications.
IMCPEXCL	""	Log reader exclusion list. If specified, this list includes the IMS SSIDs separated by commas. For example: "SSID1, SSID2". You can also exclude DL/I batch jobs by specifying batch job names. For more information, see SSIDEXCLUDELIST.

Table 6. Source server parameters and default settings for USERSAMP(CECCUSPI) (continued)

Parameters	Default value	Description
IMCPMDOL	MODSTAT	IMS DBD validation. This value specifies how the Classic data server identifies the active ACB library for the IMS subsystem. Valid values are MODSTAT or "". Specifying "" indicates that the server will load the module OLCSTAT from the STEPLIB DD statement and dynamically allocate the OLCSTAT data set defined in the module. The OLCSTAT module is created during IMS system definition by creating a DFSMDA macro instruction with TYPE=OLCSTAT. In most cases, this module is placed in the *IMSVS.EXITLIB*. If you manage your IMS subsystems by using global online change, specify "" to dynamically allocate the OLCSTAT data set.
IMCPMODS	&IMS..MODSTAT	Use the IMCPMODS parameter to identify the name of the MODSTAT data set for the subsystem. If your IMSPlex uses global online change, you should set the IMCPMDOL parameter to "" for dynamic allocation of the OLCSTAT data set and this parameter will be ignored. If your IMSPlex does not use global online change, specify the data set name for MODSTAT updated in a data sharing environment. Because each IMS subsystem in an IMSPlex must perform online changes separately, the contents of the MODSTAT data sets for individual subsystems can be unsynchronized. The subsystem that you update first is considered the master MODSTAT.
IMCPACB	&&IMS..ACBLIB	IMSACB DD library concatenation. You can specify this keyword multiple times. The order of the multiple IMCPACB keywords defines the order in which the files are included in the concatenation for the generated JCL members.
IMCPACBA	&&IMS..ACBLIBA	IMSACBA DD library concatenation. You can specify this keyword multiple times. The order of the multiple IMCPACBA keywords defines the order in which the files are included in the concatenation for the generated JCL members.
IMCPACBB	&&IMS..ACBLIBB	IMSPACBB DD library concatenation. You can specify this keyword multiple times. The order of the multiple IMSPACBB keywords defines the order in which the files are included in the concatenation for the generated JCL members.
IMCPDBD	&&IMS..DBDLIB	DBDLIB DD library concatenation. This parameter validates that source databases are correctly augmented for data capture when DEDB databases are replicated. Full function EXIT validation is preformed by using the contents of the ACB libraries. The use of DBDLIB is for EXIT validation purposes only. You can specify this keyword multiple times. The order of the multiple IMCPDBD keywords defines the order in which the files are included in the concatenation for the generated JCL members.
Security parameters for the source server		
IMCPSAFX	CACSX04	SAFEXIT load module that enables security for the source server. A value of "" disables security for the source server. This value defines the SAFEXIT configuration parameter of the operator, administration, and monitor services for the source server. For more information, see Securing a Classic data server.
For more information about the following parameter settings for the SAFEXIT, see Security.		

Table 6. Source server parameters and default settings for USERSAMP(CECCUSPI) (continued)

Parameters	Default value	Description
IMCPADMV	N	VALIDATE=Y/N parameter on the administration service for the SAFEXIT, when the exit is enabled. This value instructs the service to validate the access authority of a user by using the resource profile or the security class.
IMCPADMR	""	ADMPROF resource profile parameter on the administration service for the SAFEXIT. This value is valid when the exit is enabled and VALIDATE=Y.
IMCPADMC	""	ADMCLASS security class parameter on the administration service for the SAFEXIT. This value is valid when the exit is enabled and VALIDATE=Y.
IMCPOPRV	N	VALIDATE=Y/N parameter on the operator service for the SAFEXIT when the exit is enabled. This value instructs the operator service to validate the access authority of a user by using the resource profile or the security class.
IMCPOPRR	""	OPRPROF resource profile parameter on the operator service for the SAFEXIT. This value is valid when the exit is enabled and VALIDATE=Y.
IMCPOPRC	""	OPRCLASS security class parameter on the operator service for the SAFEXIT. This value is valid when the exit is enabled and VALIDATE=Y.
IMCPMONV	N	VALIDATE=Y/N parameter on the monitor service for the SAFEXIT, when the exit is enabled. This value instructs the monitor service to validate the access authority of a user by using the resource profile or the security class.
IMCPMONR	""	MONPROF resource profile parameter on the monitor service for the SAFEXIT. This value is valid when the exit is enabled and VALIDATE=Y.
IMCPMONC	""	MONCLASS security class parameter on the monitor service for the SAFEXIT. This value is valid when the exit is enabled and VALIDATE=Y.
Log stream parameters for the source server		
IMCPLGST	IMSSRC.DIAGLOG	z/OS log stream name for the source server diagnostic log. If a log stream name is not specified, the log service is configured to write to the CACLOG DD data set. This value defines the STREAMNAME configuration parameter of the log service for the source server.
IMCPLGDS	Y	Identifies whether the z/OS log stream should use DASD or the coupling facility: • Y: DASD • N: Coupling facility This value is valid when IMCPLGST is specified.
IMCPLGRT	7	Retention period, in days, to retain the log records before they for they are eligible to be deleted. This value is valid when IMCPLGST is specified.
IMCPLGSC	STG1	Storage class (STG_DATACLAS) for the log stream. This value is valid when IMCPLGST is specified.
IMCPLGSR	CCL1	Coupling facility structure name (STRUCTNAME). This value is valid when IMCPLGST is specified and the coupling facility is chosen (LGSTRDASD=”N”).

Table 6. Source server parameters and default settings for USERSAMP(CECCUSPI) (continued)

Parameters	Default value	Description
IMCPEVST	IMSSRC.EVENTS	z/OS log stream for the Classic event log. This value defines the EVENTLOG configuration parameter of the log service for the source server.
IMCPEVDS	Y	Identifies whether the z/OS log stream should use DASD or the coupling facility: • Y: DASD • N: Coupling facility This value is valid when IMCPEVST is specified.
IMCPEGRT	14	Retention period, in days, to retain the log records before they for they are eligible to be deleted. This value is valid when IMCPEVST is specified.
IMCPEVSC	STG1	Storage class (STG_DATACLAS) for the log stream. This value is valid when IMCPEVST is specified.
IMCPEVSR	CCP1	Name of the coupling facility structure (STRUCTNAME). This value is valid when IMCPEVST is specified and the coupling facility is chosen (LGSTRDASD=”N”).

Customization parameters file: Target server section:

The target server section of the customization parameters contains parameters that are specific to a target server.

The following table lists the parameters the target server section of the customization parameters file, the parameter default values, and a description of each parameter.

Table 7. Target server parameter and default settings for USERSAMP(CECCUSPI)

Parameters	Default value	Description
Target server parameters section		
IMAPHLQD	USER.V11R3M00.IMSTRG	High-level qualifier (HLQ) for data sets created for the target server: • Binary configuration data sets • Subscription data set • Replication mapping data set This value replaces the references to &APHLQ.. in other keyword values such as IMAPCFGD, IMAPCFGX, and IMAPSUBS.
IMAPCFGD	&APHLQ..CACCFGD	Name of the configuration data file for the target server. This value is prefixed with the IMAPHLQD high-level qualifier.
IMAPCFGX	&APHLQ..CACCFGX	Name of the configuration index file for the target server. This value is prefixed with the IMAPHLQD high-level qualifier.
IMAPHST	0.0.0.0	Host name or IP address where the target server will run. This value is used in the definition of the COMMSTRING configuration parameter of the connection handler for the target server.
IMAPPORT	9087	Port number that the target server connection handler service listens on. This listen port communicates with the Classic Data Architect to monitor and manage the target server. This value is used in the definition of the COMMSTRING configuration parameter of the connection handler for the target server.

Table 7. Target server parameter and default settings for USERSAMP(CECCUSPI) (continued)

Parameters	Default value	Description
IMAPSUBS	&APHLQ..SUB	Data file name for target server subscriptions. This value is prefixed with the IMAPHLQD high-level qualifier.
IMAPRMDS	&APHLQ..RM	Data file name for target server replication mapping. This value is prefixed with the IMAPHLQD high-level qualifier.
IMAPLPRT	5002	Port that the apply service listens on to communicate with the capture service. This value is used in the definition of the LISTENURL configuration parameter of the apply service for the target server.
IMAPCDPG	" "	Host code page of the system for the target server. This value defines the global HOSTCODEPAGE configuration parameter for the target server.
IMANMICS	" "	Network management interface (NMI) communication string for the target server. Sets the NMICOMMSTRING configuration parameter for the monitoring service.
IMAPIHLQ	IMSAP	High-level qualifier (HLQ) assigned to the IMS libraries that the target server uses to access IMS. This HLQ is applied to the IMS data sets in the STEPLIB sections of the generated JCL. This value replaces the references to &IMS.. in other keyword values such as IMAPSTPL.
IMAPSTPL	&IMS..SDFSRESL	Library concatenation, for example STEPLIB concatenation. You can specify this keyword multiple times. The order of the multiple IMAPSTPL keywords defines the order in which the files are included in the concatenation for the generated JCL members.
IMAPMDOL	MODSTAT	IMS DBD validation. This value specifies how the Classic data server identifies the active ACB library for the IMS subsystem. Valid values are MODSTAT or "". Specifying "" indicates that the server will load the module OLCSTAT from the STEPLIB DD statement and dynamically allocate the OLCSTAT data set defined in the module. The OLCSTAT module is created during IMS system definition by creating a DFSMDA macro instruction with TYPE=OLCSTAT. In most cases, this module is placed in the *IMSVS.EXITLIB*. If you manage your IMS subsystems by using global online change, specify "" to dynamically allocate the OLCSTAT data set.
IMAPMODS	MODSTAT.DSN	Use the IMCPMODS parameter to identify the name of the MODSTAT data set for the subsystem. If your IMSPlex uses global online change, you should set the IMCPMDOL parameter to "" for dynamic allocation of the OLCSTAT data set and this parameter will be ignored. If your IMSPlex does not use global online change, specify the data set name for MODSTAT updated in a data sharing environment. Because each IMS subsystem in an IMSPlex must perform online changes separately, the contents of the MODSTAT data sets for individual subsystems can be unsynchronized. The subsystem that you update first is considered the master MODSTAT.
IMAPACB	&&IMS..ACBLIB	IMSPACB DD library concatenation. You can specify this keyword multiple times. The order of the multiple IMCPACB keywords defines the order in which the files are included in the concatenation for the generated JCL members.
IMAPACBA	&&IMS..ACBLIBA	IMSPACBA DD library concatenation. You can specify this keyword multiple times. The order of the multiple IMCPACBA keywords defines the order in which the files are included in the concatenation for the generated JCL members.

Table 7. Target server parameter and default settings for USERSAMP(CECCUSPI) (continued)

Parameters	Default value	Description
IMAPACBB	&&IMS..ACBLIBB	IMSPACBB DD library concatenation. You can specify this keyword multiple times. The order of the multiple IMSPACBB keywords defines the order in which the files are included in the concatenation for the generated JCL members.
IMSAPBKDB	CECBKMK	DBD name for the bookmark database.
IMSDRAUS	DRAUSER	User ID that the DRA initialization services uses to access IMS.
IMSDRASX	00	Suffix for customizing the IMS PZP module.
Bookmark parameters for the target server		
IMAPUORT	Y	Enables completed UOR tracking. This value sets the apply service DONEUORTRACKING parameter and tailors the bookmark database appropriately. Specify N if all segments being replicated have unique sequence fields. Otherwise, do not change the default value. See "Creating a bookmark database" for more information about completed UOR tracking and the changes that are required to the bookmark database when this option is enabled.
IMAPUORL	100000	Identifies the maximum number of source UORs that will be recorded in the bookmark database for a subscription when completed UOR tracking is enabled. Sets the DONEUORLIMT configuration parameter. Setting a limit of the number of UORs that can be tracked allows you to size the bookmark database when completed UOR tracking is active.
IMAPBKDB	CECBKMK	DBD name for the bookmark database.
IMAPBKDD	CECBKMK	Bookmark database DD or AREA name.
IMAPBKDS	512	Bookmark database block size. You need to increased this value if you plan to use completed UOR tracking.
Security parameters for the target server		
IMAPSAFX	CACSX04	SAFEXIT load module that enables security for the target server. A value of " " disables security for the server. This value defines the SAFEXIT configuration parameter of the operator, administration, and monitor services for the target server. For more information, see Securing a data server.
For more information about the following parameter settings for the SAFEXIT, see Security.		
IMAPADMV	N	VALIDATE=Y/N parameter on the administration service for the SAFEXIT, when the exit is enabled. This value instructs the service to validate the access authority of a user by using the resource profile or the security class.
IMAPADMR	" "	ADMPROF resource profile parameter on the administration service for the SAFEXIT. This value is valid when the exit is enabled and VALIDATE=Y.
IMAPADMC	" "	ADMCLASS security class parameter on the administration service for the SAFEXIT. This value is valid when the exit is enabled and VALIDATE=Y.
IMAPOPRV	N	VALIDATE=Y/N parameter on the operator service for the SAFEXIT when the exit is enabled. This value instructs the operator service to validate the access authority of a user by using the resource profile or the security class.

Table 7. Target server parameter and default settings for USERSAMP(CECCUSPI) (continued)

Parameters	Default value	Description
IMAPOPRR	" "	OPRPROF resource profile parameter on the operator service for the SAFEXIT. This value is valid when the exit is enabled and VALIDATE=Y.
IMAPOPRC	" "	OPRCLASS security class parameter on the operator service for the SAFEXIT. This value is valid when the exit is enabled and VALIDATE=Y.
IMAPMONV	N	VALIDATE=Y/N parameter on the monitor service for the SAFEXIT, when the exit is enabled. This value instructs the monitor service to validate the access authority of a user by using the resource profile or the security class.
IMAPMONR	" "	MONPROF resource profile parameter on the monitor service for the SAFEXIT. This value is valid when the exit is enabled and VALIDATE=Y.
IMAPMONC	" "	MONCLASS security class parameter on the monitor service for the SAFEXIT. This value is valid when the exit is enabled and VALIDATE=Y.
Log stream parameters for the target server		
IMAPLGST	IMSTRG.DIAGLOG	z/OS log stream name for the target server diagnostic log. If a log stream name is not specified, the log service is configured to write to the CACLOG DD data set. This value defines the STREAMNAME configuration parameter of the log service for the target server.
IMAPLGDS	Y	Identifies whether the z/OS log stream should use DASD or the coupling facility: • Y: DASD • N: Coupling facility
IMAPLGRT	7	Retention period, in days, to retain the log records before they for they are eligible to be deleted.
IMAPLGSC	STG1	Storage class (STG_DATACLAS) for the log stream.
IMAPLGSR	CAL1	Coupling facility structure name (STRUCTNAME). This value is valid when IMAPLGST is specified and the coupling facility is chosen (LGSTRDASD=”N”).
IMAPEVST	IMSTRG.EVENTS	z/OS log stream for the Classic event log. This value defines the EVENTLOG configuration parameter of the log service for the target server.
IMAPEVDS	Y	Identifies whether the z/OS log stream should use DASD or the coupling facility: • Y: DASD • N: Coupling facility This value is valid when IMAPEVST is specified.
IMAPEGRT	14	Retention period, in days, to retain the log records before they for they are eligible to be deleted. This value is valid when IMAPEVST is specified.
IMAPEVSC	STG1	Storage class (STG_DATACLAS) for the log stream. This value is valid when IMAPEVST is specified.
IMAPEVSR	CAE1	Name of the coupling facility structure (STRUCTNAME). This value is valid when IMAPEVST is specified and the coupling facility is chosen (LGSTRDASD=”N”).

Installing Classic data servers

Follow the customization procedure for one of the following types of IMS server environments:

- Source server on a separate LPAR from the target server
- Target server on a separate LPAR from the source server
- Both a source server and a target server on the same LPAR

Installing IMS source servers

You can follow the installation customization process to install and customize an IMS source server on one LPAR. In this case, the corresponding target server is installed on a different LPAR.

Before you begin

Before you begin the installation customization process, you must complete the SMP/E installation and the steps required to prepare the installation environment.

Procedure

1. Edit the user samples allocation utility JCL in the installation samples member for Data Replication for IMS, SCACSAMP(CECCUSI1). Follow the instructions in the JCL to edit the job card and procedure variables and to specify the following input parameters:

   CACINHLQ=CAC.V11R3M00
   The value specified for the CACINHLQ keyword must match the high-level qualifier of the installation data sets that the SMP/E installation produces for Data Replication for IMS.

   CACUSHLQ=USER.V11R3M00.IMSSRC
   The value specified for the CACUSHLQ keyword is the high-level qualifier for the *USERHLQ*.USERSAMP and *USERHLQ*.USERCONF data sets that the samples allocation utility creates or updates.

   CACDUNIT=SYSALLDA
   The value specified for the CACDUNIT keyword identifies the disk unit that is used when allocating the *USERHLQ*.USERSAMP and *USERHLQ*.USERCONF data sets. This is an optional parameter. You do not need to specify a value for CACDUNIT if SMS manages the data sets.

   CACDVOLM=
   The value specified for the CACDVOLM keyword identifies the disk volume that is used when allocating the *USERHLQ*.USERSAMP and *USERHLQ*.USERCONF data sets. This is an optional parameter. You do not need to specify a value for CACDVOLM if SMS manages the data sets.

   CACSTGCL=
   The value specified for the CACSTGCL keyword identifies the SMS storage class that is used when allocating the *USERHLQ*.USERSAMP and *USERHLQ*.USERCONF data sets. This is an optional parameter. Specify a value for CACSTGCL only when a specific storage class is required.

   CACMGTCL=
   The value specified for the CACMGTCL keyword identifies the SMS management class that is used when allocating the

USERHLQ.USERSAMP and *USERHLQ*.USERCONF data sets. This is an optional parameter. Specify a value for CACMGTCL only when a specific management class is required.

ISPFHLQ=ISP
The value specified for the ISPFHLQ keyword identifies the high-level qualifier for the ISPF installation. The samples allocation utility runs a TSO batch application and uses TSO functions.

ISPFLANG=ENU
The value specified for the ISPFLANG keyword identifies the language prefix for the ISPF installation.

SERVERROLE=IMSR_SOURCE
The value specified for the SERVERROLE keyword identifies the type of Classic data server to install and customize. Specify the value IMSR_SOURCE to generate the JCL and components required for a source server environment for Data Replication for IMS.

2. Submit SCACSAMP(CECCUSI1) to allocate the *USERHLQ*.USERSAMP and *USERHLQ*.USERCONF data sets. Verify that all job steps result in a return code <= 4.

   This job populates the *USERHLQ*.USERSAMP data set with the necessary objects and the customization parameters file for Data Replication for IMS, CECCUSPI.
3. Edit the Data Replication for IMS customization parameters file *USERHLQ*.USERSAMP(CECCUSPI) to provide customization parameters.

   This file will contain only the parameters that apply to a source server environment. See the customization parameters file settings for details.
4. Submit the generated *USERHLQ*.USERSAMP(CECCUSI2) customization utility JCL. Verify that all job steps result in a return code <= 4.

   The *USERHLQ*.USERSAMP data set is populated with the customized JCL and objects needed to run a source server.
5. Define the z/OS log stream for the event log and the diagnostic log for the source server.
   a. Verify that you have the authority required to run the Administrative Data Utility (IXCMIAPU). The job that defines the logs runs this utility.
   b. Submit the generated *USERHLQ*.USERSAMP(CECISSLS) JCL to define the log stream and logs.
   c. Verify that all job steps result in a return code = 0.
6. Submit the generated JCL to create a new source server configuration or to migrate a source server configuration.
   - If you are creating a new configuration data set for the source server, submit the generated *USERHLQ*.USERSAMP(CECISCFG) JCL to allocate and initialize the source server configuration. This job populates the configuration with the service definitions needed for a source server environment.
   - If you are migrating the source server from V10R1M00 or later to V11R3M00, submit the generated *USERHLQ*.USERSAMP(CECISMCF) JCL to allocate new configuration data sets for the source server and populate the new source server configuration based on the previous source server configuration.

   Verify that all job steps result in a return code = 0.
7. Submit the generated JCL to create new subscription and replication mapping data sets or to migrate subscription and replication mapping data sets.

- If you are creating new subscription data sets, submit the generated *USERHLQ*.USERSAMP(CECISSUB) JCL to allocate the subscription and replication mapping data sets for the source server.
- If you are migrating the source server from V10R1M00 or later to V11R3M00, submit the generated *USERHLQ*.USERSAMP(CECISSUB) JCL to allocate the subscription and replication mapping data sets for the source server. Then submit the generated USERHLQ.USERSAMP(CECISMSU) JCL to copy the subscription and replication mapping data sets from the previous version of the data sets to the V11R3M00 data sets before starting the source server.

8. Submit the generated *USERHLQ*.USERSAMP(CECIMSSC) JCL to start the source server. When the Classic data server starts, the following services are customized and running in the source server:
   - Administration service
   - Capture service
   - Connection handler service
   - IMS log reader service
   - Logger service
   - Operator service
   - Monitor service
   - Region controller service

Results

The source server is now operational. When the corresponding target server is running, you can connect to both the source and target servers by using the Classic Data Architect and configure replication.

Installing IMS target servers

You can follow the installation customization process to install and customize an IMS target server on one LPAR. In this case, the corresponding source server is installed on another LPAR.

Before you begin

Before you begin the installation customization process, you must complete the SMP/E installation and the steps required to prepare the installation environment.

Procedure

1. Edit the user samples allocation utility JCL in the installation samples member for Data Replication for IMS, SCACSAMP(CECCUSI1). Follow the instructions in the JCL to edit the job card and procedure variables and to specify the following input parameters:

   CACINHLQ=CAC.V11R3M00
   The value specified for the CACINHLQ keyword must match the high-level qualifier of the installation data sets that the SMP/E installation produces for Data Replication for IMS.

   CACUSHLQ=USER.V11R3M00.IMSTRG
   The value specified for the CACUSHLQ keyword is the high-level qualifier for the *USERHLQ*.USERSAMP and *USERHLQ*.USERCONF data sets that the samples allocation utility creates or updates.

   CACDUNIT=SYSALLDA
   The value specified for the CACDUNIT keyword identifies the disk

unit that is used when allocating the *USERHLQ*.USERSAMP and *USERHLQ*.USERCONF data sets. This is an optional parameter. You do not need to specify a value for CACDUNIT if SMS manages the data sets.

CACDVOLM=
The value specified for the CACDVOLM keyword identifies the disk volume that is used when allocating the *USERHLQ*.USERSAMP and *USERHLQ*.USERCONF data sets. This is an optional parameter. You do not need to specify a value for CACDVOLM if SMS manages the data sets.

CACSTGCL=
The value specified for the CACSTGCL keyword identifies the SMS storage class that is used when allocating the *USERHLQ*.USERSAMP and *USERHLQ*.USERCONF data sets. This is an optional parameter. Specify a value for CACSTGCL only when a specific storage class is required.

CACMGTCL=
The value specified for the CACMGTCL keyword identifies the SMS management class that is used when allocating the *USERHLQ*.USERSAMP and *USERHLQ*.USERCONF data sets. This is an optional parameter. Specify a value for CACMGTCL only when a specific management class is required.

ISPFHLQ=ISP
The value specified for the ISPFHLQ keyword identifies the high-level qualifier for the ISPF installation. The samples allocation utility runs a TSO batch application and uses TSO functions.

ISPFLANG=ENU
The value specified for the ISPFLANG keyword identifies the language prefix for the ISPF installation.

SERVERROLE=IMSR_TARGET
The value specified for the SERVERROLE keyword identifies the type of Classic data server to install and customize. Specify the value IMSR_TARGET to generate the JCL and components required for a target server environment for Data Replication for IMS.

2. Submit SCACSAMP(CECCUSI1) to allocate the *USERHLQ*.USERSAMP and *USERHLQ*.USERCONF data sets. Verify that all job steps result in a return code <= 4.

   This job populates the *USERHLQ*.USERSAMP data set with the necessary objects and the customization parameters file for Data Replication for IMS, CECCUSPI.
3. Edit the Data Replication for IMS customization parameters file *USERHLQ*.USERSAMP(CECCUSPI) to provide customization parameters.

   This file will contain only the parameters that apply to a target server environment. See the customization parameters file settings for details.
4. Submit the generated *USERHLQ*.USERSAMP(CECCUSI2) customization utility JCL. Verify that all job steps result in a return code <= 4.

   The *USERHLQ*.USERSAMP data set is populated with the customized JCL and objects needed to run a target server.
5. Define the z/OS log stream for the event log and the diagnostic log for the target server.

a. Verify that you have the authority required to run the Administrative Data Utility (IXCMIAPU). The job that defines the logs runs this utility.
b. Submit the generated *USERHLQ*.USERSAMP(CECITSLS) JCL to define the log stream and logs.
c. Verify that all job steps result in a return code = 0.

6. Submit the generated JCL to create a new target server configuration or to migrate a target server configuration.
   - If you are creating a new configuration data set for the target server, submit the generated *USERHLQ*.USERSAMP(CECITCFG) JCL to allocate and initialize the target server configuration. This job populates the configuration with the service definitions needed for a target server environment.
   - If you are migrating the target server from V11R1M00 to V11R3M00, submit the generated USERHLQ.USERSAMP(CECITMCF) JCL to allocate new target server configuration data sets and populate the new target server configuration based on the previous target server configuration.

   Verify that all job steps result in a return code = 0.
7. Submit the generated JCL to create new subscription and replication mapping data sets or to migrate subscription and replication mapping data sets.
   - If you are creating new subscription data sets, submit the generated *USERHLQ*.USERSAMP(CECITSUB) JCL to allocate the subscription and replication mapping data sets for the target server.
   - If you are migrating the target server from V11R1M00 to V11R3M00, submit the generated *USERHLQ*.USERSAMP(CECITSUB) JCL to allocate the subscription and replication mapping data sets for the target server. Then submit the generated *USERHLQ*.USERSAMP(CECITMSU) JCL to copy the subscription and replication mapping data sets from the previous version of the data sets to the V11R3M00 data sets before starting the target server.
8. Submit the generated *USERHLQ*.USERSAMP(CECIMSTG) JCL to start the target server. When the Classic data server starts, the following services are customized and running in the target server:
   - Administration service
   - Apply service
   - Connection handler service
   - DRA initialization service
   - Logger service
   - Monitor service
   - Operator service
   - Region controller service

Results

The target server is now operational. When the corresponding source server is running, you can connect to both the target and source servers by using the Classic Data Architect and configure replication. See Preparing a target IMS subsystem for replication to complete the setup required to prepare the target server environment.

Installing IMS source and target servers on the same LPAR

You can follow the installation customization process to install and customize both an IMS source server and an IMS target server on the same LPAR.

Before you begin

Before you begin the installation customization process, you must complete the SMP/E installation and the steps required to prepare the installation environment.

About this task

If you choose to run both a source and a target server on the same LPAR, you can install the installation components for both Classic data servers into the same *USERHLQ*.USERSAMP and *USERHLQ*.USERCONF data sets. You specify the installation parameters for the source server and target server separately in the customization parameters file (CECCUSPI) that is generated during the installation customization process. The parameters that you specify are not shared between the Classic data servers. As a result, the source and target servers point to separate IMS instances, and separate installation components such as configurations, subscriptions, and replication mappings.

As an alternative, you can run SCACSAMP(CECCUSI1) twice to allocate separate *USERHLQ*.USERSAMP and *USERHLQ*.USERCONF data sets:

- For the source server, with SERVERROLE= IMSR_SOURCE, targeting one set of PDS files (*USERHLQ*.SOURCE.USERSAMP and *USERHLQ*.SOURCE.USERCONF). See Installing IMS source servers.
- For the target server, with SERVEROLE=IMSR_TARGET, targeting a different set of PDS files (*USERHLQ*.TARGET. USERSAMP and *USERHLQ*.TARGET.USERCONF). See Installing IMS target servers.

Procedure

1. Edit the user samples allocation utility JCL in the installation samples member for Data Replication for IMS, SCACSAMP(CECCUSI1). Follow the instructions in the JCL to edit the job card and procedure variables and to specify the following input parameters:

   CACINHLQ=CAC.V11R3M00
   The value specified for the CACINHLQ keyword must match the high-level qualifier of the installation data sets that the SMP/E installation produces for Data Replication for IMS.

   CACUSHLQ=USER.V11R3M00.IMSREPL
   The value specified for the CACUSHLQ keyword is the high-level qualifier for the *USERHLQ*.USERSAMP and *USERHLQ*.USERCONF data sets that the samples allocation utility creates or updates.

   CACDUNIT=SYSALLDA
   The value specified for the CACDUNIT keyword identifies the disk unit that is used when allocating the *USERHLQ*.USERSAMP and *USERHLQ*.USERCONF data sets. This is an optional parameter. You do not need to specify a value for CACDUNIT if SMS manages the data sets.

   CACDVOLM=
   The value specified for the CACDVOLM keyword identifies the disk volume that is used when allocating the *USERHLQ*.USERSAMP and *USERHLQ*.USERCONF data sets. This is an optional parameter. You do not need to specify a value for CACDVOLM if SMS manages the data sets.

   CACSTGCL=
   The value specified for the CACSTGCL keyword identifies the SMS

storage class that is used when allocating the *USERHLQ*.USERSAMP and *USERHLQ*.USERCONF data sets. This is an optional parameter. Specify a value for CACSTGCL only when a specific storage class is required.

CACMGTCL=
The value specified for the CACMGTCL keyword identifies the SMS management class that is used when allocating the *USERHLQ*.USERSAMP and *USERHLQ*.USERCONF data sets. This is an optional parameter. Specify a value for CACMGTCL only when a specific management class is required.

ISPFHLQ=ISP
The value specified for the ISPFHLQ keyword identifies the high-level qualifier for the ISPF installation. The samples allocation utility runs a TSO batch application and uses TSO functions.

ISPFLANG=ENU
The value specified for the ISPFLANG keyword identifies the language prefix for the ISPF installation.

SERVERROLE=(IMSR_SOURCE, IMSR_TARGET)
The value specified for the SERVERROLE keyword identifies the type of Classic data server to install and customize. Specify the values IMSR_SOURCE, IMSR_TARGET to generate the JCL and components required for a source and target server environment for Data Replication for IMS.

2. Submit SCACSAMP(CECCUSI1) to allocate the *USERHLQ*.USERSAMP and *USERHLQ*.USERCONF data sets. Verify that all job steps result in a return code <= 4.

   This job populates the *USERHLQ*.USERSAMP data set with the necessary objects and the customization parameters file for Data Replication for IMS, CECCUSPI.

3. Edit the Data Replication for IMS customization parameters file *USERHLQ*.USERSAMP(CECCUSPI) to provide customization parameters. This file will contain only the parameters that apply to source and target server environments.

   Note: The values for the IMCPPORT and IMAPPORT parameters must be different.
   See the customization parameters file settings for the source server components and the target server components for details.

4. Submit the generated *USERHLQ*.USERSAMP(CECCUSI2) customization utility JCL. Verify that all job steps result in a return code <= 4.

   The *USERHLQ*.USERSAMP data set is populated with the customized JCL and objects needed to run a source and a target server.

5. Define the z/OS log stream for the event log and the diagnostic log for the source server.
   a. Verify that you have the authority required to run the Administrative Data Utility (IXCMIAPU). The job that defines the logs runs this utility.
   b. Submit the generated *USERHLQ*.USERSAMP(CECISSLS) JCL to define the log stream and logs.
   c. Verify that all job steps result in a return code = 0.

6. Submit the generated JCL to create a new source server configuration or to migrate a source server configuration.

- If you are creating a new configuration data set for the source server, submit the generated *USERHLQ*.USERSAMP(CECISCFG) JCL to allocate and initialize the source server configuration. This job populates the configuration with the service definitions needed for a source server environment.
- If you are migrating the source server from V11R1M00 to V11R3M00, submit the generated USERHLQ.USERSAMP(CECISMCF) JCL to allocate new configuration data sets for the source server and populate the new source server configuration based on the previous source server configuration.

Verify that all job steps result in a return code = 0.

7. Submit the generated JCL to create new subscription and replication mapping data sets or to migrate subscription and replication mapping data sets for the source server.
   - If you are creating new subscription data sets, submit the generated *USERHLQ*.USERSAMP(CECISSUB) JCL to allocate the subscription and replication mapping data sets for the source server.
   - If you are migrating the source server from V11R1M00 to V11R3M00, submit the generated *USERHLQ*.USERSAMP(CECISSUB) JCL to allocate the subscription and replication mapping data sets for the source server. Then submit the generated USERHLQ.USERSAMP(CECISMSU) JCL to copy the subscription and replication mapping data sets from the previous version of the data sets to the V11R3M00 data sets before starting the source server.
8. Submit the generated *USERHLQ*.USERSAMP(CECIMSSC) JCL to start the source server. When the Classic data server starts, the following services are customized and running in the source server:
   - Region controller service
   - Logger service
   - Operator service
   - Connection handler service
   - Administration service
   - Monitor service
   - Capture service
   - IMS log reader service
9. Submit the generated JCL to create a new target server configuration or to migrate a target server configuration.
   - If you are creating a new configuration data set for the target server, submit the generated *USERHLQ*.USERSAMP(CECITCFG) JCL to allocate and initialize the target server configuration. This job populates the configuration with the service definitions needed for a target server environment.
   - If you are migrating the target server from V11R1M00 to V11R3M00, submit the generated USERHLQ.USERSAMP(CECITMCF) JCL to allocate new configuration data sets for the target server and populate the new target server configuration based on the previous target server configuration.

   Verify that all job steps result in a return code = 0.
10. Submit the generated JCL to create new subscription and replication mapping data sets or to migrate subscription and replication mapping data sets for the target server.

- If you are creating new subscription data sets, submit the generated *USERHLQ*.USERSAMP(CECITSUB) JCL to allocate the subscription and replication mapping data sets for the target server.
- If you are migrating the target server from V11R1M00 to V11R3M00, submit the generated *USERHLQ*.USERSAMP(CECITSUB) JCL to allocate the subscription and replication mapping data sets for the target server. Then submit the generated *USERHLQ*.USERSAMP(CECITMSU) JCL to copy the subscription and replication mapping data sets from the previous version of the data sets to the V11R3M00 data sets before starting the target server.

11. Submit the generated *USERHLQ*.USERSAMP(CECIMSTG) JCL to start the target server. When the Classic data server starts, the following services are customized and running in the target server:
    - Region controller service
    - Logger service
    - Operator service
    - Connection handler service
    - Administration service
    - Monitor service
    - Apply service
    - DRA initialization service

Results

The source and target server are now operational. You can connect to both the source and target servers by using the Classic Data Architect and configure replication. To complete the setup required to prepare the Classic data servers, see:

- Preparing a source IMS subsystem for replication
- Preparing a target IMS subsystem for replication

Installing the Classic Data Architect

To install the Classic Data Architect you extract an installation zip package and run IBM Installation Manager.

Before you begin

If you have installed an earlier beta version of the Classic Data Architect (CDA) Version 11.3, you must uninstall it first. See “Uninstalling the Classic Data Architect” on page 60.

Ensure that your client computer meets the following minimum system requirements:

Operating system:

- Microsoft Windows 8.1, 8, and 7

Memory

1024 MB.

Disk space

800 MB for both IBM Installation Manager and the Classic Data Architect.

Procedure

Start the CDA installation as a non-Administrator or as an Administrator.

Method	Description
To start the CDA installation as a non-Administrator	1. Unzip the installation package to a temporary directory. 2. From the command line, change to the temporary directory and take one of the following actions: • On Windows: Run `userinst.exe`. • On Linux: Run `userinst`. 3. Proceed through the Installation Manager wizard.
To start the CDA installation as an Administrator	1. Unzip the installation package to a temporary directory. 2. From the command line, change to the temporary directory and take one of the following actions: • On Windows: Run `install.exe`. • On Linux: Run `install`. 3. Proceed through the Installation Manager wizard.

Results

When the `userinst` or `install` program starts, Installation Manager is installed if it is not already on your computer and automatically started. Installation Manager is configured with the location of the repository (installation files) for IBM InfoSphere Classic Data Architect V11.3.

What to do next

You can launch the product:

- From Windows: **Start** > **Programs** > **IBM Classic Data Architect** > **IBM InfoSphere Classic Data Architect V11.3**. Alternatively, from a command line run `<installRoot>/eclipse.exe`.
- From Linux: **Applications** > **IBM Classic Data Architect** > **IBM InfoSphere Classic Data Architect V11.3**. Alternatively, from a command line run `<installRoot>/eclipse`.

Starting IBM Installation Manager

If you start the Classic Data Architect installation from the downloadable image, IBM Installation Manager starts automatically.

About this task

When you start the installation of Classic Data Architect from the downloadable image, either by running the install or userinst program, IBM Installation Manager automatically starts. If Installation Manager is not installed, it will be installed and automatically started.

If you already installed Installation Manager, you can start it by using one of the methods in the following procedure.

Procedure

Start the Installation Manager from Windows or Linux.

Method	Description
To start the Installation Manager from Windows	Click **Start** > **All Programs** > **IBM Installation Manager** > **IBM Installation Manager**.
To start the Installation Manager from Linux	Click **Applications** > **IBM Installation Manager** > **IBM Installation Manager** or alternatively change to *Installation Manager directory* /eclipse and run IBMIM.

Uninstalling the Classic Data Architect

You can uninstall the Classic Data Architect by using the IBM Installation Manager.

Before you begin

To uninstall the IBM InfoSphere Classic Data Architect product package, you must log in to the system by using the same user account that you used to install the product package. You must close the programs that you installed by using IBM Installation Manager.

About this task

You can use the **Uninstall** option in IBM Installation Manager to uninstall the IBM InfoSphere Classic Data Architect product package from a single installation location. You can also uninstall all of the installed packages from every installation location.

To uninstall the CDA product package, complete the steps in the following procedure.

Procedure

1. Start IBM Installation Manager.
2. On the Start page, click the **Uninstall** button.
3. On the Uninstall Packages page, from the **Installation Packages** list, select IBM InfoSphere Classic Data Architect version 11.3.0, and click **Next**.
4. On the Summary page, review the list of packages that will be uninstalled. The Complete page displays after the packages are removed.
5. Click **Finish**.

Chapter 3. Configuring Data Replication for IMS

Configuring Data Replication for IMS requires setup tasks that include gathering information, preparing the IMS subsystems at the source and target, deploying and securing Classic data servers, creating subscriptions, and performing an initial load of the target IMS databases.

Gathering information and securing your environment

Data Replication for IMS requires initial setup tasks, such as obtaining library authorizations and port assignments, setting up access to the RECON data sets, and securing your data servers.

Obtaining library authorizations for the APF

Data Replication for IMS requires authorized program facility (APF) authorization for the installation load library on both the source and the target logical partitions (LPARS).

About this task

The source and target servers and utility programs use z/OS facilities and services that require APF authorization. The load library data sets must be defined to z/OS as APF authorized.

Procedure

- Before setting up and configuring the source and target data servers, ensure that the installation load library SCACLOAD is APF authorized.

Validation operations require access to the following IMS libraries:

- The active library for IMS application control blocks (ACBLIB)
- For direct entry (DEDB) databases, the IMS library for database descriptions (DBDLIB)

  This DBDLIB requirement applies only to the source server.

Obtaining ports for replication

Data Replication for IMS uses TCP/IP to communicate between various components, which requires multiple port numbers.

Before you begin

Obtain the port assignments that you need from your network administrator to ensure that you are using dedicated ports for the exclusive use of Data Replication for IMS.

About this task

Each subscription that replicates change data to the target server uses a TCP/IP connection. Work with your network administrator to ensure that source and target servers have authorization to use the assigned ports.

Procedure

1. You must obtain a port assignment for each of the following communication channels on the source server.
   a. A port for connections with the Classic Data Architect.

      You define this port number as part of the COMMSTRING configuration parameter for the INIT service. 9087 is the well-known and reserved port for connections between the Classic Data Architect and a connection handler service (INIT) that runs in the source server.
   b. Define a listening port for the IMS log reader service. You define this port number as part of the NOTIFICATIONURL configuration parameter for the LRSI service.
   c. An authorized outbound port for a TCP/IP connection to each target server that you deploy.
2. Obtain a port assignment for each of the following communication channels on the target server.
   a. A port for connections with the Classic Data Architect.

      You must define a port number as part of the COMMSTRING configuration parameter for the INIT service on the target server. 9087 is the well-known port for connections between the Classic Data Architect and a connection handler service (INIT) that runs in the target server.
   b. An inbound port for which you previously obtained authorization to use for connections with a source server.

      You must define this port number as part of the LISTENURL configuration parameter for the APLY service on the target server.

Setting up IMS DBRC access for a source data server

The source server requires access to the recovery control (RECON) data sets in the source IMS subsystem. IMS Database Recovery Control (DBRC) provides this access.

About this task

DBRC cannot access the IMS RECON1, RECON2, and RECON3 data sets on behalf of the Classic data server unless the user ID associated with the address space has RACF® UPDATE or equivalent authority to the dataset names.

You can use the application programming interface (API) for DBRC to grant installation control to individual requests that users issue. If your site uses DBRC command authorization, you might also have to authorize the following query requests:

- STARTDBRC
- STOPDBRC
- QUERY RECON
- QUERY LOG STARTTIME
- QUERY OLDS SSID=

For information about the security features of the DBRC API and how these requests correspond with SAF resources, see the IMS system programming documentation.

Procedure

1. Ask your Security Administrator to grant the user associated with the address space for the Classic data server RACF UDPATE authority to the IMS RECON1, RECON2, and RECON3 dataset names.
2. If your site uses DBRC command authorization, authorize any required query requests.

Security

To implement security, work with your Security Administrator to define any required classes and profiles, and then secure your Classic data servers.

Use these different layers to secure your replication environment:

- "z/OS-level security"
- "Server security"

z/OS-level security

The System Authorization Facility (SAF) is a z/OS interface that programs use to communicate with an external security manager (ESM), such as the Resource Access Control Facility (RACF). SAF and your ESM work together to grant access rights to system resources, such as the following:

- Classic data servers
- Services

RACF *classes* organize profiles into groupings of related system resources. *Profiles* define security for specific users, groups, and protected resources. Your security administrator creates classes and profiles, then grants users or groups READ or CONTROL access to the resources in the profiles.

For more information about z/OS security, see the *z/OS Security Server RACF Security Administrator's Guide*.

Server security

The following table describes specific tasks on a Classic data server and the required SAF access that a user needs to perform them. A user with CONTROL access automatically has READ access.

Table 8. Tasks and required user access

Task	Service to configure	Minimum required access
View subscriptions	Administration service	READ
Create, update, and delete subscriptions	Administration service	CONTROL
Manage replication (start, stop, change state)	Administration or Operator service	CONTROL
Monitor metrics	Monitoring service	READ
Issue remote console commands	Operator service	READ

The SAFEXIT service parameter

Secure your Classic data server by using the SAFEXIT service parameter. The following services have a SAFEXIT parameter:

- Administration service (PAA)
- Monitoring service (MAA)
- Operator service (OPER)

SAFEXIT and protected resources

If you define the SAFEXIT parameter by specifying the SAFEXIT value `CACSX04,VALIDATE=N` the Classic data server performs user ID and password authentication only. No other user validation takes place.

Use the SAFEXIT value `CACSX04,VALIDATE=Y` to grant multiple users different levels of access to system resources based on classes and profiles. For each of these services, you can override the default class and profile by specifying different z/OS class or profile names as values for service parameters. Your ESM then authenticates user access by checking the specified profiles.

Table 9. Default classes and profiles per service, with override service parameters

Service name	Default class	Default profile	Override class parameter	Override profile parameter
Administration service	SERVAUTH	CEC.ADMIN	ADMCLASS	ADMPROF
Monitoring service*	SERVAUTH	CEC.MONITOR	MONCLASS	MONPROF

You can supply values for these parameters during the installation customization process or by setting them in the Console Explorer in the Classic Data Architect. For example:

`SAFEXIT="CACSX04,VALIDATE=Y,ADMCLASS=`*`xxxxxxxx`*`,ADMPROF=`*`yyyy.yyyyy`*`"`

Administration service

The administration service secures user connections to the Classic data server by checking z/OS credentials and access rights to protected resources.

The user account that the Classic Data Architect uses to connect to the Classic data server authenticates to the administration service.

Monitoring service

The monitoring service secures access to subscription states, statuses, and metrics.

A Classic data server provides different ways of accessing monitoring information, depending on your solution. Data Replication for IMS uses the Classic Data Architect to display subscription status and metrics.

The Classic Data Architect accesses monitoring information by using the same user account that logged in to the administration service and connected with the Classic data server.

Operator service

The operator service authenticates users who run remote console commands on the Classic data server, including the MTO command files run from the Classic Data Architect. Console users are generally system operators who make z/OS system console requests to a Classic data server.

The operator service does not secure operator commands that you enter from a z/OS console or equivalent interface, such as the System Display

and Search Facility (SDSF). When you issue commands to the console you have implied authority to issue commands to the Classic data server, so you must secure these command interfaces to prevent unrestricted access. Ensure that users who run remote operator commands have READ access to connect the data server and to issue the DISPLAY command, and CONTROL access for all other commands.

Securing a Classic data server

To secure your Classic data server, create security classes and profiles in your external security manager (ESM) and configure service-level parameters in the server.

About this task

Work with your Security Administrator to determine the appropriate levels of security for your site. For best results, secure as many operations as you can while maintaining site standards and performance. You can secure administrative connections to your Classic data server, remote operator commands, monitoring, and the replication process itself.

When you secure the configuration data sets and the VSAM files for subscriptions and replication mappings, use the following table to locate them. You specified high level qualifiers for these data sets in the customization parameters file when you performed the installation customization process.

Table 10. High level qualifiers and corresponding data sets.

High level qualifier (HLQ)	Server type	Function
IMCPCFGD="&&CPHLQ..CACCFGD" IMCPCFGX="&&CPHLQ..CACCFGX"	Source	Configuration data sets
IMCPSUBS="&CPHLQ..SUB"	Source	Subscription data sets
IMCPRMDS="&CPHLQ..RM"	Source	Replication mapping data sets
IMAPCFGD="&APHLQ..CACCFGD" IMAPCFGX="&APHLQ..CACCFGX"	Target	Configuration data sets
IMAPSUBS="&APHLQ..SUB"	Target	Subscription data sets
IMAPRMDS="&APHLQ..RM"	Target	Replication mapping data sets

Procedure

1. Determine which users require access to your Classic data servers and the level of access that each user requires.

   Decide whether each user requires READ or CONTROL access to subscriptions, administrative functions, and remote console commands.
2. Ensure that each user has a valid z/OS user account.
3. If you grant different levels of access to system resources for multiple users, follow these steps.
   a. Ask your System Administrator to define the profiles and classes that you require by using the System Authorization Facility (SAF).

      Follow the remaining steps for each applicable service:

      - Administration service (PAA)

- Monitoring service (MAA)
- Operator service (OPER)

b. Ensure that VALIDATE=Y for the **SAFEXIT** parameter.

   VALIDATE=Y is the default setting, so omitting the keyword has the same effect as specifying VALIDATE=Y.

   If you use SAFEXIT with VALIDATE=N then the Classic data server only validates that each user has a valid user account and password. Users then have full access to objects on the Classic data server.

c. Optional: Override the default class and profile names in the *xxx***CLASS** and *xxx***PROF** parameters.

4. Perform the following security tasks for the source server.
   a. Grant UPDATE access to the IMS RECON data sets.
   b. If you use IMS DBRC command authorization, the Classic data server needs LIST.* authority for DBRC.
   c. Ensure that the data server has READ authority for the IMS DBDLIB data sets that the data server references. This library is also accessed for EXIT validation purposes if a subscription contains a DEDB database.

      Grant READ authority for all IMS log data sets that need to be accessed.

      - For a DB/DC or DBCTL subsystem these data sets include the primary and secondary online log data sets and the corresponding primary and secondary system log data sets.
      - For logs produced by DL/I batch jobs one of these types of logs will only be accessed if it contains data capture log records. When processing historical changes, the log can potentially contain data capture log records.

5. Perform the following security tasks for both the source and target server.
   a. Grant READ access for CLASS(FACILITY) RESOURCE(BPX.CONSOLE) to the job or started task for the Classic data server.
   b. Grant UPDATE access to the configuration data sets and the VSAM files for subscriptions and replication mappings.
   c. The user ID associated with the job or started task needs an OMVS segment for TCP/IP access.
   d. Grant READ access to the data sets referenced by IMSACB, IMSACBA, and IMSACBB and the data set that is used to determine the active ACB library when replication starts for a subscription.

      Whether IMSACBA or IMSACBB is used is based on whether the server contains a MODSTAT DD statement. Or if OLCSTAT is used, most likely it will be dynamically allocated using information found in the IMS OLCSTAT dynamic allocation entry that is found in the IMS EXITLIB data set that is referenced in the STEPLIB DD statement. The server needs read authority to either the IMS MODSTAT or OLCSTAT data set.
   e. Grant READ access to any IMS library referenced in the STEPLIB DD statement for the data server. Those libraries require APF authorization.

Using the SAF exit for security validations

The SAF exit verifies access for user and client connections and controls the operations that a user can perform.

The SAF exit controls access to system resources based on classes and profiles. The exit authenticates user passwords at connection time. In addition, you can configure the exit to perform the following functions:

- Validate user authorization to access system resources, access metrics data, or run remote console commands.
- Authenticate the TCP/IP address of client connections at connection time

Recommendation: Use the supplied sample exit (CACSX04) in the SAMPLIB data set. When you use the sample exit, you do not need to re-assemble or re-linkedit the exit.

If you choose to modify or replace the supplied sample exit, you must assemble and bind the module as described in the overview of the SAF exit API.

Activating the SAF exit:

To activate the SAF exit, you set up security resources and define configuration parameters during the installation customization process.

Before you begin

Ensure that the STEPLIB DD statement in the JCL for the Classic data server references the library where the SAF exit load module (CACSX04) is located. The library must be APF-authorized

About this task

Ensure that only valid z/OS users can access resources by setting SAFEXIT=CACSX04 for the administration service, monitoring service and operator service.

You can configure the supplied SAF exit CACSX04 by using the following configuration parameters. You can supply configuration parameters only when you use the IBM-supplied version of the SAF exit CACSX04.

VALIDATE=*Y/N*

Indicates whether the exit should validate that the user ID has authority to access system resources, access metrics data, or run remote console commands.

Specify an operand of **N** to indicate that the exit routine should not perform access rights validation.
The exit routine is invoked for access rights checking regardless of the **Y** or **N** value of the operand. The default value for VALIDATE is **Y**.

This parameter helps you to control access with greater precision:

- Ensure that only valid users can access resources by setting VALIDATE=Y against the administration, monitor, or operator service.

  VALIDATE=Y authenticates each individual resource.
- Eliminate the overhead of verifying that the user has authority to access a resource by setting VALIDATE=N against the administration, monitor, or operator service. (The Classic data server performs user ID and password authentication only.)

NETACCESS=*Y/N*

Indicates whether the exit should validate the IP address of the connected client to authenticate access to the Classic data server.

Set the value to Y when the IP address of the connected client is known and the SERVAUTH parameter of the RACROUTE REQUEST=VERIFY invocation is supplied. The RACROUTE operation is successful when the

associated user ID has at least READ-level access rights to the network security zone resource. If the security system indicates that it cannot make a decision in response to the request because a corresponding network security zone resource profile does not exist, the SAF exit regards the response as Access Denied.

A value of N indicates that the SERVAUTH parameter is omitted from the RACROUTE REQUEST=VERIFY invocation. This is the default.

ADMCLASS=*administrator-class-name*
Indicates the name of the security class that contains a profile that provides access authentication.

This parameter is valid if VALIDATE=Y on the administration service for the SAF exit. If this parameter is not specified, SERVAUTH is the name of the default security class.

ADMPROF=*administrator-profile-name*
Indicates the name of the resource profile that provides access authentication.

This parameter is valid if VALIDATE=Y on the administration service for the SAF exit. If this parameter is not specified, CEC.ADMIN is the default profile name.

MONCLASS=*monitor-class-name*
Indicates the name of the security class that contains a profile that provides access authentication.

This parameter is valid if VALIDATE=Y on the monitoring service for the SAF exit. If this parameter is not specified, SERVAUTH is the default name of the monitor security class.

MONPROF=*monitor-profile-name*
Indicates the name of the resource profile that provides access authentication.

This parameter is valid if VALIDATE=Y on the monitor service for the SAF exit. If this parameter is not specified, CEC.MONITOR is the default profile name.

OPERCLASS=*operator-class-name*
Indicates the name of the security class that contains a profile that provides access authentication.

This parameter is valid if VALIDATE=Y on the operator service for the SAF exit. If this parameter is not specified, SERVAUTH is the default name of the operator security class.

OPERPROF=*operator-profile-name*
Indicates the name of the resource profile that provides access authentication.

This parameter is valid if VALIDATE=Y on the operator service for the SAF exit. If this parameter is not specified, CEC.OPER is the default profile name.

If you want to modify or replace the supplied sample exit, you must assemble and bind the module as described in the overview of the SAF exit API.

To configure the SAF exit and verify that it is working:

Procedure

1. Edit the sample SAF exit (SCACSAMP member CACSX04) if you need to customize CACSX04.
2. Modify any required SAFEXIT configuration parameters to the service definition for the administration, monitor, or operator service classes (PAA, MAA, or OPER) by using the Classic Data Architect or MTO SET,CONFIG command.
3. After changing the SAF definition for any services that you need to secure, stop and restart the Classic data server.

Example

Ensure that only a valid z/OS user can connect to the operator, administration, and monitoring services by using the following command. This command runs the supplied SAF exit with the default values:

```
F <Data-Server-Name>,SET,CONFIG,SERVICE=<Service-Name>,SAFEXIT=CACSX04
```

If you want to take advantage of the flexibility of the exit by using SAF exit parameters against a monitoring service, add name/value pairs to the command, as follows:

```
F <Data-Server-Name>,SET,CONFIG,SERVICE=<Monitor-Service-Name>,
SAFEXIT="CACSX04,VALIDATE=Y"
```

The first value of the SAFEXIT parameter must be the exit name. You must enclose the SAFEXIT values in double quotes.

SAF exit: API overview:

The parameters passed to the SAF exit are defined by the CACSXPL4 member located in the SCACMAC library. The SAF exit is called for one of three functions: initialization, validation, or termination.

The CACSXPL4 macro describes the interface to the SAF exit. The comments provided in the CACSXPL4 macro describe the SAF structure fields and their intended usage. Comments contained within the CACSX04 source code describe the interface to and the intended behavior of the SAF exit.

Assembling and binding the SAF exit

Member CACALX04 in the USERSAMP data set contains a JCL stream that you can use to assemble and bind the sample SAF exit, CACSX04.

Requirements for assembling and binding CACSX04:

- You must direct the assembler to produce Extended Format (GOFF) object code. Specifying the assembler options GOFF,OBJ,NODECK satisfies this requirement and directs the assembler to write the object code to the file referenced by the SYSLIN DD statement.
- The CACSX04 executable module (a program object) must reside in a PDSE that is included in the list of APF-authorized libraries.
- The traditional load module format is not supported.
- You must direct the z/OS Program Management Binder to produce a program object at the z/OS V1R10 level or higher with support for mixed-case external symbol names. The resultant program object must contain the re-entrant attribute and must not contain the AC(1) attribute.

Specifying the binder options RENT,CASE=MIXED,COMPAT=ZOSV1R10 meets these requirements.

For more information about assembler options, see the High Level Assembler documentation in the z/OS information center.

For more information about the z/OS binder options see the z/OS MVS™ Program Management documentation in the z/VM® information center.

Setting up IMS for replication

Setting up IMS for replication requires setup tasks on the source and target IMS subsystems.

Preparing a source IMS subsystem for replication

On the source IMS subsystem, augment the database descriptions (DBDs) for each database that you want to replicate.

Augmenting DBDs for change data capture

To generate IMS records for change data capture, you must augment each DBD by adding an **EXIT** parameter to the DBD control statement.

About this task

In some cases, you must also add an **EXIT** parameter to a segment definition to override the default setting for the DBD.

To augment a DBD to capture changes from all segments:

Procedure

1. Add an EXIT keyword to the DBD statement to define default values for all segments in the DBD.

   Ideally, all segments in the database have unique keys. In this case, use the following values for the EXIT parameter:

   ```
   EXIT=(*,KEY,DATA,NOPATH,(NOCASCADE),LOG)
   ```

   If you have IMS Version 13.1 installed at the source, include the following options:

   - For any segments with an insert rule of HERE, include the INPOS option. This option includes positioning information that will be used to properly insert new instances at the target:

     ```
     EXIT=(*,KEY,DATA,NOPATH,INPOS,(NOCASCADE),LOG)
     ```

   - For a DEDB database that has subset pointers defined, include the SSPCMD option:

     ```
     EXIT=(*,KEY,DATA,NOPATH,SSPCMD,(NOCASCADE),LOG)
     ```

   - For applications that use IMS FLD calls to update a DEDB database, include the FLD option as follows:

     ```
     EXIT=(*,KEY,DATA,NOPATH,FLD,(NOCASCADE),LOG)
     ```

   - If necessary you can include some or all of these IMS Version 13.1 options as follows:

     ```
     EXIT=(*,KEY,DATA,NOPATH,INPOS,SSPCMD,FLD,(NOCASCADE),LOG)
     ```

2. If a child segment has a parent segment with no key or a non-unique key, supply an **EXIT** parameter on the SEGM statement to override the default values for the DBD and generate path information in the log record.

Example: `EXIT=(*,KEY,DATA,PATH,(NOCASCADE),LOG)`

3. Run DBDGEN for the updated DBD.
4. Run the ACBGEN utility to update all program specification blocks (PSBs) that reference the DBD.
5. Add the updated DBD and PSB members to your production libraries for access control blocks (ACBLIB).

Augmentation of DBD and segment statements:

Follow these guidelines when you augment database descriptions (DBDs) for IMS replication.

The DBD modifications in this section affect only the DBD definition in the DBD and ACB libraries and do not affect the physical database.

Augmentation concepts

The log records that IMS generates for recovery purposes do not contain enough contextual information for change data capture. Therefore, you must augment each replicated database to generate special log records. IMS creates this type of log record for every segment in the database when you supply an **EXIT** parameter on the DBD control statement in the DBD source definition.

Choose the right level of augmentation for your DBD. You generate too much data in the records and place an unnecessary load on your system if, for example, you include unnecessary path information. If you add fewer **EXIT** options than you require, you might not capture all of the changes that you want to process.

The minimum requirement for Data Replication for IMS is to capture any deletes, inserts, or updates to a segment and concatenated key information. Data Replication for IMS also requires additional before image (PATH) data for parent segments that either do not have a sequence field or have a non-unique sequence field.

Choose the EXIT options that work best with your data to keep the size of your log records to a minimum.

Cascade deletes

Data Replication for IMS does not require cascade delete information. If you augmented a DBD for use by other applications that require cascade delete information, Data Replication for IMS discards this additional information.

EXIT parameter in DBD and SEGM statements:

To augment the DBD for which you want to capture changes, specify the information that you want to capture in the **EXIT** parameter.

Purpose

IMS supports an **EXIT** parameter for the DBD control statement and the SEGM control statement. An **EXIT** parameter on the DBD statement defines default values for all segments in the DBD. An **EXIT** parameter on a SEGM statement overrides the default values.

Format

The format of the **EXIT** parameter is as follows:

```
EXIT=(Exit-Name,KEY|NOKEY,DATA|NODATA,PATH|NOPATH,FLD|NOFLD,INPOS|NOINPOS,SSPCMD|
NOSSPCMD(CASCADE|NOCASCADE,KEY|NOKEY,DATA|NODATA,PATH|NOPATH),LOG|NOLOG)
```

For more information about the format of the EXIT parameter and the CASCADE operand, see the documentation for the IMS system utility *Database Description (DBD) Generation utility*.

Keywords

Table 11. Keywords for the EXIT parameter

Keyword	Purpose
Exit-Name	Indicates the name of a DPropNR synchronous data capture exit. Specify an asterisk (*) to indicate that there is no exit. Specify NONE to deactivate an exit routine on a SEGM statement. Data Replication for IMS does not use data capture exits. However, it can co-exist with DPropNR or your own exits. If you do not have any data capture exits, specify an asterisk (*) for the **Exit-Name** keyword.
KEY | NOKEY	Indicates whether you want the log records to contain concatenated key information about the physical path to deleted, inserted, or updated segments. Data Replication for IMS requires the default value of `KEY`.
DATA | NODATA	Indicates whether you want to include before images and after images of the changed segment in the log records for deleted, inserted, or updated segments. Data Replication for IMS requires the default value of `DATA`.
PATH | NOPATH	Indicates whether you want to include physical segment data in the log records for the parents of deleted, inserted, or updated segments. The default is `NOPATH`.
FLD | NOFLD	Identifies whether IMS should generate a data capture log record when an application updates a DEDB database using a FLD call. The default is NOFLD. If you are unsure whether any of your applications use FLD calls, include the FLD option on a segment level EXIT specification. Enabling this option does not create additional overhead from an IMS perspective, unless FLD updates are encountered. In that case you will want the changes to be captured and replicated to the target.

Table 11. Keywords for the EXIT parameter (continued)

Keyword	Purpose
INPOS | NOINPOS	Identifies whether IMS should include positioning information in the data capture log record when an application inserts a new instance for a segment with an insert rule of HERE. The default is NOINPOS. When enabled the data capture log record can contain the image of the segment before which the new instance was inserted. Data Replication for IMS uses this information to attempt to insert the new segment instance at the same twin chain location at the target. **Recommendation**: If you have any segments with an insert rule of HERE, code the INPOS option. Data Replication for IMS does not validate that this option is enabled.
SSPCMD | NOSSPCMD	Identifies whether IMS should include subset pointer update information in data capture log records. When enabled IMS will also generate data capture log records when subset pointers are updated using get calls. In some situations the data capture log records that IMS produces also include information about a subset pointer that was used to retrieve/position in the source database. The default is NOSSPCMD. Data Replication for IMS performs the corresponding subset pointer update operations at the target and in most cases uses R command code information to assist in positioning at the target site. **Recommendation** If a DEDB has any subset pointers defined, code the SSPCMD option at the SEGM level. Enabling this option does not create additional overhead from an IMS perspective, unless subset pointers are updated. In that case you will want the changes to be captured and replicated to the target.
CASCADE | NOCASCADE	Indicates whether you want log records to contain cascade delete information for deleted segments that have child segments. For Data Replication for IMS, specify `NOCASCADE`.
LOG | NOLOG	Indicates whether you want to generate log records for data capture in the IMS log files. If you specify an asterisk (*) for **`Exit-Name`**, the default is `LOG`. For Data Replication for IMS, specify `LOG`.

Example

```
EXIT=(*,KEY,DATA,NOPATH,(NOCASCADE),LOG)
```

Setting up notification exits

IMS replication uses its own versions of two IMS notification exits (DFSPPUE0 and DFSFLGX0) to inform the IMS log reader service about the start of IMS subsystems and the start and stop of DL/I batch jobs.

About this task

You must perform some tasks to set up these Classic versions of the IMS exits:

The partner program exit (DFSPPUE0)

Informs the IMS log reader service about the start of IMS subsystems, enabling change data capture to include participating subsystems when they start during replication.

The logger exit routine (DFSFLGX0)

Informs the IMS log reader service about the start and stop of DL/I batch jobs, enabling change data capture to suspend ordering operations for participating subsystems for the duration of each job step that issues DL/I calls and has the Classic logger exit installed.

Customizing the configuration table module for notification exits:

To use the notification exits, the configuration table module (CECE1OPT) must be customized for your environment.

About this task

The notification exits communicate their event notification to IMS change-capture through TCP/IP. The system administrator must specify the TCP/IP host name and port information for this exit by customizing, compiling, and linking the assembler module named CECE1OPT. CECE1OPT requires customization because it contains default values that may not be applicable to your specific environment. Samples of the configuration module CECE1OPT are provided in the sample library *USERHLQ*.SAMPLIB.

The address specified in CECE1OPT must be the same as the configuration parameter NOTIFICATIONURL in the log reader service.

Procedure

1. Edit a copy of the CECE1OPT module specification (in the *USERHLQ*.*. SAMPLIB library)., specifying the following parameters:

   IPVSN
   Indicates whether to use Internet Protocol Version 4 (IPv4) or Internet Protocol Version 6 (IPv6). If 4 is specified, the exit connects to the Classic data server by using the Ipv4 protocol. If 6 is specified, the exit connects to the server by using the IPv6 protocol. Values other than 4 or 6 result in an error.

   TCPADDR4
   The IP address to be used by the IMS exits for event notification if an IPVSN of 4 is specified. (Ignored if an IPVSN of 6 is specified.) The address is specified as a four byte hexadecimal value.

   TCPADDR6
   The IP address to be used by the IMS exits for event notification if an IPVSN of 6 is specified. (Ignored if an IPVSN of 4 is specified.) The address is specified as a sixteen byte hexadecimal value.

   Port The port number to be used by the IMS exits for event notification. The port is specified as a two byte hexadecimal value.

```
IPVSN    DC    AL1(4)              | 4=IPv4, 6=IPv6
         DC    AL1(0)              | Reserved
TCPADDR4 DC    AL1(192),AL1(168),AL1(4),AL1(37)   IPv4 addr
TCPADDR6 DC    AL2(9)               IPv6 addr - 1st 16 bits
         DC    AL2(3155)            IPv6 addr - 2nd 16 bits
         DC    AL2(3155)            IPv6 addr - 3rd 16 bits
         DC    AL2(3155)            IPv6 addr - 4th 16 bits
         DC    AL2(0)               IPv6 addr - 5th 16 bits
         DC    AL2(0)               IPv6 addr - 6th 16 bits
         DC    AL2(0)               IPv6 addr - 7th 16 bits
         DC    AL2(0)               IPv6 addr - 8th 16 bits
PORT     DC    AL2(1234)                     TCP PORT
```

. Example of a CECE1OPT module specification with an IPv4 address of 192.168.4.37, and a port of 1234. The IPv6 address ignored.

2. Assemble and link the module. The module must be made available in any authorized data set concatenated in the IMS STEPLIB DD statement.

Installing the partner program exit routine:

You must install the partner program exit routine (CECPPUE0) to enable change data capture for IMS to identify and manage IMS subsystems start events that might occur during replication.

Before you begin

Customize the configuration table module.

Procedure

- If you did implement your own IMS partner program exit or are using a different exit, complete the following steps. If you did not, go to the next bullet. When linking the modules, you can link them directly into any authorized data set concatenated in the IMS STEPLIB DD statement.
  1. Edit the sample *USERHLQ*.USERSAMP(CECLRIL6) following the instructions in the file.
  2. Run the sample job *USERHLQ*.USERSAMP(CECLRIL6) to create a backup of the exit that you are using and link the IMS partner program exit CECPPUE0 object with your pre-existing partner program exit object. The new composite exit must be named DFSPPUE0 for the call made by IMS to succeed.
- If you did not implement your own IMS partner program exit and you are not using a different exit, use the provided DFSPPUE0 as is.

What to do next

Verify the correct exit installation by starting the DB/DC or DBCTL region where you installed the notification exit. You must also start the source server and start replication for at least one subscription. Look for the following operator messages in the IMS region JES messages:

```
CECZ0751I Subsystem start notification issued for <subsystem-name>
CECZ0757I TCP/IP connection established for subsystem <subsystem-name>
to IP address <ip-address> on port number <port-number>
CECZ0991I SUBSYSTEM START NOTIFICATION RECEIVED FOR <subsystem-name>.
```

Excluding IMS subsystems from ordering decisions

You can specify a list of IMS subsystems to exclude from change capture operations.

About this task

You might choose to exclude selected subsystems from processing if they are no longer active, or if they change no databases or records that affect ordering decisions.

Exclude subsystems from processing by adding them to the list parameter **SSIDEXCLUDELIST** in the IMS log reader service. You can add values to the list during installation customization or by adding values in the Console Explorer view of the Classic Data Architect.

You cannot replicate any changes that a subsystem makes if you exclude it from ordering decisions. Make sure that each subsystem that you include on the list updates databases that you do not replicate.

Procedure

Add the subsystem id (SSID) of each IMS subsystem that you want to exclude from processing to the **SSIDEXCLUDELIST** parameter. **Tip:** If the Classic data server is running when you set this parameter, stop and restart the server to enable your change to take effect.

Examples

When you use operator commands to exclude or include IMS subsystems from ordering decisions, you must add or remove each subsystem from the list separately.

Use the ADD,CONFIG command to add an IMS subsystem to the list parameter **SSIDEXCLUDELIST**:

```
F server-name,ADD,CONFIG,SERVICELIST=SSIDEXCLUDELIST,SERVICE=IMSLRS,VALUE=IMSA
```

Use the DELETE,CONFIG command to remove an IMS subsystem from the list parameter **SSIDEXCLUDELIST**:

```
F server-name,DELETE,CONFIG,SERVICELIST=SSIDEXCLUDELIST,SERVICE=IMSLRS,VALUE=IMSA
```

Using a BMP program to manage subsystem inactivity automatically

To avoid halting change data capture or increasing latency in a data sharing environment when an IMS subsystem becomes inactive, generate activity in that subsystem to flush the log buffers for the online data sets (OLDS).

About this task

To replicate data in the correct order across multiple DB/DC or DBCTL subsystems, all subsystems that are active at the source site must generate continuous activity. Otherwise, your deployment is unable to track whether an inactive subsystem has records to capture in the buffers for the online data sets (OLDS).

Use these approaches to manage subsystem inactivity automatically:

- Run a batch message processing (BMP) program to flush the log buffers when a subsystem becomes inactive.
- If you want to save space in your system log data sets (SLDS), ensure that the IMS Log Archive utility does not archive the records that the BMP generates.

Notification of inactive subsystems

The CECZ0400W message provides information about IMS subsystems that become inactive:

```
CECZ0400W IMS change capture halted due to lack of activity from
sub-system subsystem-id. No new log data has been received since:
time-stamp. Current inactivity message threshold value: threshold-value.
```

Exception: If all participating subsystems are inactive, the source server does not issue the CECZ0400W message.

Threshold-value indicates the time interval at which the source server issues the CECZ0400W message. By default, the CECZ0400W message is displayed at 30-second intervals, but you can adjust this by changing the value of the **INACTTHRESHOLD** parameter in the IMS log reader service.

Tip: If a participating IMS subsystem fails and you are not running fast database recovery (FDBR), you must manually close the log for the failed subsystem. A failed IMS subsystem with an open log is equivalent to an inactive IMS subsystem.

The BMP program

Using the first sample in the Examples section as a guide, run the IMS-supplied DFSDDLT0 utility to issue LOG DL/I calls that generate logging activity and flush the OLDS buffers. Create enough copies of the DATA lines to generate sufficient data to flush the buffers. The logged data must equal or exceed the size of the OLDS buffer that you specified in the **BLKSZ DCB** parameter when you allocated the OLDS data sets.

A single LOG call in the program DFSDDLT0 can never be greater than 9999 bytes, and cannot exceed the **IOASIZE** parameter in the IMS Program Specification Block (PSB) generation utility. If the IOAREA for this job is too large, IMS issues an AT status code.

Tip: The log records that the BMP program writes have an ID in the first two bytes of data (`x'D3D6'` in these examples). If your applications or tools access IMS log records, ensure that this log record ID does not conflict with these operations.

Save SLDS space

Use the second sample in the Examples section to help you customize the job control language (JCL) for the IMS Log Archive utility to exclude records that the BMP generates from archiving activities.

Set the **NOLOG** parameter to the hexadecimal value of the log record ID by using this syntax:

```
SLDS NOLOG (D3)
```

Use comma-separated values to specify additional IDs to exclude from logging:

```
SLDS NOLOG (D3,10,45,5F,67,69)
```

Restriction: Do not exclude log records for data capture from the archive logs by adding `'99'` to this list. Data Replication for IMS requires these type `0x99` log records for data capture.

For more information about the IMS resource configuration parameters in the first example, see the *IMS System Administration Guide*. For more information about the

Log Archive utility, see the IMS documentation for system utilities.

Procedure

1. Create automation that intercepts the message CECZ0400W.
2. Set up the automation to run a BMP program that flushes the OLDS buffers.
3. Optional: Customize the JCL for the Log Archive utility (DFSUARC0).

Examples

In this example, you run the utility DFSDDLT0 as an IMS BMP. The control statements use LOG calls to generate 35224 bytes of log data by using a GPSB with the name STLGPSBB.

```
//* CECLRIFS PROVIDE VALID JOB CARD
//*******************************************************************
//**        EXAMPLE NOT INTENDED FOR EXECUTION
//**
//**  Licensed Materials - Property of IBM
//**
//**  5655-R54, 5655-R55, 5655-R56, 5655-R57
//**
//**      Copyright IBM Corp. 2008,2010 All Rights Reserved
//**
//**  US Government Users Restricted Rights - Use, duplication or
//**  disclosure restricted by GSA ADP Schedule contract with
//**  IBM Corp.
//**
//*******************************************************************
//**
//** JOB CONTROL STATEMENTS TO GENERATE SUFFICIENT ACTIVITY AND    *
//** LOG RECORDS TO FLUSH THE OLDS BUFFERS.                        *
//** THIS WILL TEMPORARILY RESUME Data Replication for IMS LOG ORDERING     *
//** OPERATIONS FOR THE SUBSYSTEM IDENTIFIED IN THE IMSID FIELD.   *
//** SPECIFY THE FOLLOWING PARAMETERS ACCORDING TO SITE STANDARDS: *
//** o VALID JOBCARD                                               *
//** o IMSID = IMS CONTROL REGION SUBSYSTEM ID                     *
//** o EXEC BMP PROCEDURE NAME                                     *
//** o MBR = GPSBNAME                                              *
//**                                                               *
//** APAR...  ID  PREREQ.  DATE....  DESCRIPTION..................*
//**                                                               *
//*******************************************************************
//BMP      EXEC BMP8CSAM,MBR=DFSDDLT0,IMSID=IMS1,PSB=STLGPSBB
//BMP.SYSIN DD *
S22 2 2 2 2    TP     2
L        LOG
L  Z2516 DATA  LOG FILL BUFF PART 1
L        LOG
L  Z2516 DATA  LOG FILL BUFF PART 2
L        LOG
L  Z2516 DATA  LOG FILL BUFF PART 3
L        LOG
L  Z2516 DATA  LOG FILL BUFF PART 4
L        LOG
L  Z2516 DATA  LOG FILL BUFF PART 5
L        LOG
L  Z2516 DATA  LOG FILL BUFF PART 6
L        LOG
L  Z2516 DATA  LOG FILL BUFF PART 7
L        LOG
L  Z2516 DATA  LOG FILL BUFF PART 8
L        LOG
L  Z2516 DATA  LOG FILL BUFF PART 9
L        LOG
L  Z2516 DATA  LOG FILL BUFF PART 10
```

```
L        LOG
L  Z2516 DATA  LOG FILL BUFF PART 11
L        LOG
L  Z2516 DATA  LOG FILL BUFF PART 12
L        LOG
L  Z2516 DATA  LOG FILL BUFF PART 13
L        LOG
L  Z2516 DATA  LOG FILL BUFF PART 14
/*
```

The next example shows customized JCL for the Log Archive utility DFSUARC0. The **NOLOG** parameter specifies that the utility does not write log records to your SLDS if the log record ID begins with x'D3'.

```
/ARCHIVE2 JOB MSGCLASS=A,CLASS=A,MSGLEVEL=(1,1)
//*
//ARC2    EXEC PGM=DFSUARC0,PARM='SYSA'
//STEPLIB   DD DSN=IMS.&SYS2..SDFSRESL,DISP=SHR
//*   COPY FROM 2 OLDS TO DUAL SLDS                    */
//DFSOLP02  DD DSN=OLP902,DISP=SHR
//DFSOLP00  DD DSN=OLP900,DISP=SHR
//DFSOLS00  DD DSN=OLS900,DISP=SHR
//DFSOLS02  DD DSN=OLS902,DISP=SHR
//DFSSLOGP  DD DSN=SLDSP.D82001.N001,DISP=(,KEEP),
//             UNIT=TAPE,VOL=(,,,99),LABEL=(,SL)
//DFSSLOGS  DD DSN=SLDSS.D82001.N001,DISP=(,KEEP),
//             UNIT=TAPE,VOL=(,,,99),LABEL=(,SL)
//RECON1   DD  DSN=RECON1,DISP=SHR
//RECON2   DD  DSN=RECON2,DISP=SHR
//SYSPRINT DD  SYSOUT=A
//SYSUDUMP DD  SYSOUT=A
//SYSIN    DD  *
  SLDS NOLOG (D3) FEOV (08000)
  /* DFSUARC0 FORCES AN EOV FOR A DATA SET           */
  /* AFTER WRITING 8000 BLOCKS, AND WRITES NO        */
  /* D3 LOG RECORDS TO THE DATA SET THAT THE         */
  /* DFSLOGP DFSLOGS DD POINTS TO.                   */
/*
```

What to do next

The SYSPRINT DD statement in DFSUARC0 points to the data set that contains a list of excluded records. Verify that records beginning with your log record ID are on the list.

Managing subsystem inactivity manually:

Follow these manual steps to prevent inactive IMS subsystems from affecting the performance of your replication environment.

About this task

Data Replication for IMS provides manual and automated options to manage the effects of inactive subsystems on latency or message activity.

Ideally, use the CECZ0400W message as a trigger to generate new activity automatically. You can do this by running a prepared BMP or WIFI region that generates a sufficient number of records to flush the buffers.

You can easily manage subsystem inactivity if your site can tolerate latency:

- Wait for new activity on inactive subsystems to flush the buffers naturally and permit ordering operations to continue.

- Set a higher value for the **INACTTHRESHOLD** parameter in the IMS log reader service (IMSL) to reduce message activity on the system console.

  If you do not expect update activity on inactive subsystems for an extended period of time, increasing this value reduces the frequency of the CECZ0400W message.

Use the following steps to manage inactive subsystems manually in response to the CECZ0400W message:

Procedure

- If you do not expect workload on the identified subsystem, shut it down.

  The source server removes this subsystem from ordering decisions.
- Issue the IMS command **/CHECKPOINT** for the identified subsystem.

  This is a temporary measure that flushes the buffers and writes their log records to the online data sets (OLDS). Ordering operations can continue to merge changes from all participating subsystems until one of the following events occurs:
  - The period of time elapses that you specified in the **INACTTHRESHOLD** parameter
  - A subsystem becomes inactive

Preparing a target IMS subsystem for replication

To set up your replication environment on the target site, define an apply PSB for each subscription, prepare the IMS subsystem, and create a bookmark database.

Before you begin

Complete the following tasks before you begin these steps:

1. Install an IMS subsystem on the target site whose configuration matches that of the source IMS subsystem.
2. Ensure that all libraries for application control blocks (ACBLIBs) on the source and target sites match.

About this task

A bookmark database stores information about the subscription restart positions so that the IMS log reader service knows where to resume processing the logs for the subscription in the event of an error or outage.

Because Data Replication for IMS sends only change data to the database replica, you must begin replication with matching copies of your source and target databases.

You must also have an apply PSB for each subscription, which defines a program control block (PCB) for each target database and a PCB for the related bookmark database. PCBs provide authorization and other information that enables the target server to write changes to the target databases for that subscription.

Before you start replication, synchronize the structure and contents of your source and target databases by installing the database definition (DBD) on the target IMS subsystem and performing an initial load.

To prepare a target IMS subsystem for replication :

Procedure

1. Create a bookmark database.
2. Install the DBD and perform an initial load for each target database.

Results

You can then set up your servers. You must have a source server for change data capture and a target server for apply processing.

Creating a bookmark database

You must have a bookmark database for each target server. Data Replication for IMS uses this database to store restart positions at the target IMS subsystem.

About this task

A bookmark database stores a restart position for each subscription. The restart position specifies where log reading resumes in the IMS logs on the source IMS subsystem if an error occurs or if replication stops. This restart information is stored in the BOOKMARK segment which is keyed by the 8-byte source system identifier. This identifier is a unique "shorthand" value assigned to each subscription.

If you enable completed UOR tracking, the bookmark database contains additional segments. Additional topics provide an overview of completed UOR tracking and guidelines for sizing and customizing the bookmark database to support completed UOR tracking.

You can use the sample DBD definitions to deploy a bookmark database. See Bookmark database deployment options for detailed information.

Procedure

- Follow these guidelines for naming the bookmark database.

  Use any unique and valid IMS database name. Ensure that the name matches the value of the following parameters:

  – The **IMSBOOKMARKDB** configuration parameter for the apply service
  – The DBDNAME keyword in the program communication block (PCB) definition for the bookmark database in the apply PSB (program specification block)
  – The DBD keyword in the DATABASE macro that you use to define the bookmark database to IMS

    See the supplied sample that immediately follows the procedure.

    For more information about the **DATABASE** macro, see the *IMS V13 System Definition Reference*. For more information about the **CREATE DB** command, see *IMS Command Reference, Vol. 1*. For more information about creating a database description (DBD) for input to the IMS DBDGEN utility, see the *IMS DB Administration Guide*.

- Define the bookmark database to IMS by using a customized copy of the sample in the Examples section as input to the IMS stage 1 process. Alternatively, issue the **CREATE DB** command in the example.

  Run one of the following customized sample members.

Option	Description
If you plan to use the HDAM version of the bookmark database	Run the following customized sample members from *USERHLQ*.USERSAMP in the order listed: 1. CECIBKDB – to create the bookmark database in the DBD library 2. CECIBKDA – to create the bookmark database dynamic allocation definitions 3. CECIBKRG – to register the bookmark database with DBRC 4. CECIBKPL – to create a load PSB 5. CECIBKLD – to allocate and load the bookmark database with a dummy record
If you plan to use the DEDB version of the bookmark database	Run the following customized sample members from *USERHLQ*.USERSAMP in the order listed: 1. CECIBKFP – to create the bookmark database in the DBD library 2. CECIBKDA – to create the bookmark database dynamic allocation definitions 3. CECIBKRD – to register the bookmark database with DBRC **Note:** Before you run CECIBKIN, ensure that a DBD entry exists in the IMS ACBLIB. DEDB AREA initialization requires that a DBD entry exist in the IMS ACBLIB. You need to take one of the following actions: • Option 1: Define a subscription and generate an apply PSB. • Option 2: Use the sample load PSB provided in CECIBKPL to include the bookmark database in the IMS ACBLIB by changing PROCOPT=L to PROCOPT=G. With either method, before running CECIBKIN you need to define the PSB needs in the IMS PSBLIB by running PSBGEN. When that step is complete you can run the ACBGEN utility with a BUILD PSB=PSB-Name statement that identifies the name of the either the apply PSB (option 1) or CECBKL1 (option 2). Ensure that the ACBGEN utility updates the same ACBLIB that is referenced in the CECIBKIN JCL. If you use option 2, you do not need to define the CECBKL1 PSB to IMS because this PSB will not be used to access the bookmark database. 4. CECIBKIN – to allocate the bookmark database and perform area initialization

Examples

The sample shows a DATABASE macro statement that you can customize with the name of your bookmark database and use as input to the IMS stage 1 process to define the database to IMS.

```
*******************************************************
*   Define Data Replication for IMS bookmark Database  *
*******************************************************
         DATABASE DBD=CECBKMK,ACCESS=UP
```

The following example shows a **CREATE DB** command that you can use to define the database to IMS. This approach uses dynamic resource definition:

```
TSO SPOC input:

   CREATE DB NAME(CECBKMK) SET(ACCTYPE(UPD))

TSO SPOC output:

   DBName   MbrName    CC
   CECBKMK  IMS1        0
```

Bookmark database deployment options:

You can use the sample DBD definitions to deploy a HDAM or a DEDB database.

The following sample members are provided to assist you in the definition process.

Table 12. Sample DBD definitions for the bookmark database.

Sample member	Description
CECIBKDB	Customized HDAM bookmark DBD definition.
CECIBKFP	Customized DEDB bookmark DBD definition.
CECIBKDA	Sample JCL to run the IMS IMSDALOC procedure and customized control card input to allow the bookmark database to be dynamically allocated by IMS.
CECIBKRG	Customized sample JCL and DBRC commands to register the bookmark database in the target IMS RECON.
CECIBKPL	Sample JCL that controls card input used to create a PSB that is used to load the HDAM version of the bookmark database.
CECIBKLD	Customized JCL that defines and loads the HDAM version of the bookmark database.
CECIBKIN	Customized JCL that defines and initializes the DEDB version of the bookmark database.

The contents of these members are tailored based on the information that you provide in the CACCUSPI customization parameters file and are placed in your *USERHLQ*.USERSAMP when you run the CECCUSI2 job.

Recommendation: Use a DEDB bookmark database because it is more efficient than using a HDAM version.

To further decrease the cost of bookmark maintenance, the customization process tailors the DEDB DBRC INIT.DB and INIT.DBDS commands to use the data space version of the virtual storage option (VSO) to eliminate physical I/O operations against the bookmark database and reduce locking overhead to a minimum.

Completed UOR tracking:

Completed UOR tracking is a capability that allows UORs that typically require serialization to be applied in parallel.

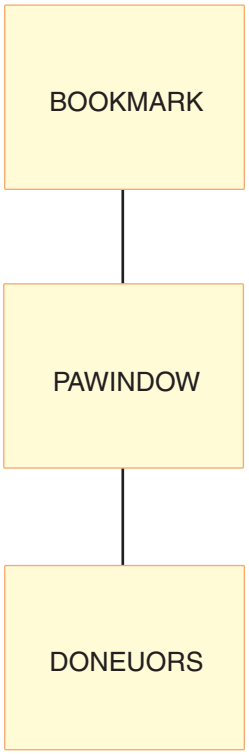
A UOR that performs an action involving a segment with a non-unique sequence field, or uses a subset pointer, is eligible for completed UOR tracking.

When completed UOR tracking is enabled, the bookmark database contains the additional segments PAWINDOW and DONEUORS. These additional segments record information about source UORs that have been applied that cannot use adaptive apply logic during restart processing.

PAWINDOW
The PAWINDOW segment is used as a grouping entity. Each subscription contains one PAWINDOW segment instance.

DONEUORS
The DONEUORS segment identifies a source UOR that contains references to segments without a unique sequence field or updates a subset pointer that is applied at the target.

The database structure required for completed UOR tracking is shown in the following illustration:



You cannot use adaptive apply to re-apply a UOR that is eligible for completed UOR tracking. By tracking the ones that have completed, if you stop replication immediately for a subscription or if there is a failure, upon restart, you can use this "list" of DONEUORS to identify whether a source UOR has already been applied at the target during restart processing.

For performance reasons, the DONEUORS segment does not have a key. As UORs that require tracking complete, they are simply added to the end of the twin chain. Each DONEUORS instance tracks information about the source units of recovery included in the UOR that was just processed. This includes the IMS recovery token and the system clock values for the first and last change record for a given recovery token. Because the source server can group multiple source UORs into a single target UOR, the DONEUORS segment is variable length. A single segment can reference multiple source UORs.

The PAWINDOW segment tracks UORs that were applied in parallel that require completed UOR tracking. When changes are applied in parallel, processing completes for a UOR at random points in time. At that point in time for that subscription:

- All UORs that completed before that UOR have been applied at the target.
- No UORs that completed after that UOR has been applied at the target.

These points in time are referred to as the *contiguous committed UOR*. When completed UOR processing is in effect and one of these points is reached, the PAWINDOW segment is deleted which removes all completed UOR tracking information since it is no longer needed. For additional information about contiguous committed UORs, see the topic "Processing historical changes if replication stops" on page 11.

When enabled, maintenance of completed UOR tracking information is automatic. However, identifying when a contiguous committed UOR condition will be encountered is unpredictable. In a situation with the following conditions:

- Most of your source UORs require completed UOR tracking.
- The maximum number of PSBs that can be scheduled for the subscription is less than the number of concurrent source applications.

potentially one of the contiguous committed UOR points will not be encountered until some time after the source workload subsides to a level that is below the apply PSB limit for that subscription.

Example: You can only schedule 30 apply PSBs for a subscription. But you are running 90 concurrent BMPs that all generate UORs that require completed UOR tracking. In this case, you are not likely to reach a contiguous committed UOR" point until sometime after all of the BMPs have completed. If each BMP is generating 100,000 UORs (and UOR grouping is not active or eligible), potentially 9,000,000 DONEUORS instances will be recorded under a single PAWINOW segment.

Because the cost of inserting DONEUORS segments is fairly fixed, there is not much difference between inserting a few, several hundred, several thousand, or millions of DONEUORS segments under a single PAWINDOW segment. More challenging is bookmark database sizing, particularly if you have multiple subscriptions that might require completed UOR tracking.

Sizing the bookmark database:

This topic provides guidance for sizing the bookmark database.

The installation customization process configures your bookmark database to support completed UOR tracking by using the IMAPUORT parameter. The default setting generates a bookmark database that does not support completed UOR

tracking. Setting this parameter to Y generates a bookmark database that is capable of supporting at least five subscriptions that require completed UOR tracking.

When UOR tracking is not enabled, a bookmark database that uses a 512-byte block or control interval size is recommended. A one cylinder allocation is more than sufficient. The recommended UOW/SIZE settings of the AREA definition for this kind of configuration is: UOW=(42,35),ROOT=(2,1)

When completed UOR tracking is enabled, the recommendation is to use a block or control interval size of 4096. The HDAM version of the bookmark database uses a 50-cylinder primary allocation and 10 cylinder secondary allocation. The DEDB version allocates a 55-cylinder VSAM data set. The recommended UOW/SIZE settings are: UOW=(30,20),ROOT=(320,300).

If you need to support completed UOR tracking for more subscriptions, or for more source UORs (as described below), you should increase the SIZE settings to allocate more control intervals to independent overflow areas. The bookmark database does not contain an SDEP segment which is a consideration when increasing the size of the ESDS data set to support larger environments.

Sizing the bookmark database to support completed UOR tracking

The DONEUORLIMT configuration parameter allows you to control how many DONEUORS segments exist for a single subscription. The DONEUORLIMT configuration parameter identifies the frequency that source UORs that require completed UOR processing forces serialization. For example, the default DONEUORLIMT is 100,000. This means that for a subscription at 100,000 UOR intervals, no dependency analysis information is provided for that UOR which forces serialization and creates a 'contiguous committed UOR' condition. This condition deletes completed UOR tracking information. The DONEUORLIMT is only applied to source UORs that require completed UOR tracking.

Use of DONEUORLIMT introduces periodic serialization points which decreases performance. Since serialization does not take into account when the last contiguous committed UOR condition was reached, using this approach occasionally forces serialization when it is not necessary. The benefit is that use of DONEUORLIMT gives you some control over how large the bookmark database will grow.

From a bookmark sizing perspective, when completed UOR tracking is required you can assume that 64 bytes of space is required to track a source UOR. This estimate includes IMS overhead, unused space, etc.

Example: Calculate the total number of blocks/control intervals that you need to allocate for your bookmark database. This calculation uses the recommended 4096 block/control interval size.

- Entries per block/control interval = `4096 / 64= 64 entries per block`
- Per subscription requirement = `DONEUORLIMIT / Entries per block/control interval`
- Total = `Number of subscriptions requiring completed UOR tracking * Per subscription requirement`

Customizing the bookmark database:

You can customize the parameter setting that you set during installation customization for completed UOR tracking.

The installation customization process includes parameters that assist you in tailoring your bookmark database definition. The values that are provided create a bookmark database that does not support completed UOR tracking. To enable tracking, you need to change the parameter values to the recommended settings listed in the following table.

Table 13. Recommended installation parameters settings for completed UOR tracking.

Parameters	Recommended value	Description
IMAPUORT	Y	Enables completed UOR tracking and includes the PAWINDOW and DONEUORS segment definitions in the customized bookmark DBD definition.
IMAPBKBS	8192	Increase the bookmark block size so that IMS can insert multiple DONEUORS segments in a single block.

Defining an apply PSB for a subscription

Each subscription has its own program specification block, or *apply PSB*, that contains a program communication block (PCB) for each target database in the subscription.

Before you begin

Typically, you generate a PSB source file automatically when you create or modify a subscription in the Classic Data Architect (CDA). Alternatively, you can use this procedure to create your own PSB source file based on the sample that follows the steps.

Whether you create your apply PSB manually or generate it automatically, you must first create a data set for the PSB on the target logical partition (LPAR):

1. Create a fixed-length data set with a record length of 80 bytes.
2. Grant update authority for the data set to the user ID for the target server.

Each apply PSB also contains a PCB that references a bookmark database. Before defining your apply PSB, generate the bookmark database that the PSB refers to.

About this task

A database PCB is the PCB that defines an application program's interface to a database. Apply processing requires a database PCB for each database view that the subscription uses.

After completing the steps, the PSB is available in IMS to process transactions that update the databases in the subscription.

If your subscription supports parallel apply, the PSB must support parallel scheduling. In this case, specify `SCHDTYP=PARALLEL` on the APPLCTN definition.

For each database referenced by a PCB statement you need to include SENSEG definitions for every segment in the database.

Typically, you specify the PCB processing option PROCOPT=AP on a PCB statement and do not need to specify a value on the SENSEG statements. However, if one of the segments in the PCB is a Fast Path sequential dependent (SDEP) segment, specify PROCOPT=GI on that segments SENSEG definition.

If you plan to replicate subset pointer updates, include a SSPTR specification that allows each subset pointer to be updated on each SENSEG statement for segments that have subset pointers.

If you use CDA to generate the apply PSB for a subscription, the PCB and SENSEG statements created contain the proper information to support replication. Additionally if the apply configuration parameter IMSV11CMPAT is set to FALSE (the default), CDA also tailors the IOASIZE and SSASIZE parameters on the PSBGEN statement so that the PSB uses the minimum IMS resources required.

In addition to running PSBGEN for the PSB, you must also define the PSB to IMS. The examples show you how to do this by using the IMS stage 1 process or by issuing the IMS Type-2 **CREATE PGM** command.

Restriction: Data Replication for IMS does not support an IMS PSB defined with the PSBGEN option LANG=PL/I. The valid options are LANG=ASSEM or LANG=COBOL.

For more information about PCBs and PSBs, see Programming for IMS.

To define an apply PSB for a subscription:

Procedure

1. Assemble the PSB, using the PSB source that you generated in the Classic Data Architect or your customized sample as input.
2. Link the PSB.
3. Run ACBGEN.
4. Perform the IMS online change process.
5. Update the ACBLIB with the apply PSB.
6. Define the application (PSB) to IMS.

Examples

The following sample shows a PSB source I2ISUB1 for a subscription that contains a replication mapping for the database DI21PART. The first PCB definition CECBKMK is for the bookmark database.

```
************************************************************
*  Define PSB used for Data Replication for IMS apply process *
*  for a subscription.                                     *
************************************************************
CECBKMK  PCB  TYPE=DB,DBDNAME=CECBKMK,PROCOPT=AP,KEYLEN=9
         SENSEG NAME=BOOKMARK,PARENT=0
         SENSEG NAME=PAWINDOW,PARENT=BOOKMARK
         SENSEG NAME=DONEUORS,PARENT=PAWINDOW


DI21PART PCB    TYPE=DB,DBDNAME=DI21PART,PROCOPT=AP,KEYLEN=43
         SENSEG NAME=PARTROOT,PARENT=0
         SENSEG NAME=STANINFO,PARENT=PARTROOT
         SENSEG NAME=STOKSTAT,PARENT=PARTROOT
```

```
        SENSEG NAME=CYCCOUNT,PARENT=STOKSTAT
        SENSEG NAME=BACKORDR,PARENT=STOKSTAT
        PSBGEN LANG=ASSEM,PSBNAME=I2ISUB1
        END
```

The following sample shows an APPLCTN macro statement that you can customize and use as input to the IMS stage 1 process to define the program name to IMS.

```
*************************************************************
*  Define applications used for the Data Replication for IMS apply     *
*  process for each subscription                              *
*************************************************************
        APPLCTN PSB=I2ISUB1,SCHDTYPE=PARALLEL,LANG=ASSEM
        APPLCTN PSB=I2ISUB2,SCHDTYPE=PARALLEL,LANG=ASSEM
```

The following example shows a **CREATE PGM** command that you can use to define the application to IMS. This approach uses the dynamic resource definition process.

```
TSO SPOC input:

    CREATE PGM NAME(I2ISUB1,I2ISUB2) SET(LANG(ASSEM))

TSO SPOC output:

    PgmName  MbrName    CC CCText
    I2ISUB1  IMS1        0
    I2ISUB2  IMS1        0
```

Setting up target IMS databases for the first time

Before you start IMS replication, you must synchronize the source and target IMS databases so that replication begins with matching copies of the data.

About this task

Data Replication for IMS updates the target databases as an application rather than updating the target data sets directly, so perform an initial load at the target site for each database that you want to replicate. For example, if a database has three database data sets that contain the data, you must take an image copy of all three and install them at the target site before starting replication.

Take a nonconcurrent image copy of all areas in a direct entry database (DEDB) or all the partitions of a high availability large database (HALDB), and then load the target databases before starting replication.

Restriction: Data Replication for IMS does not support using concurrent image copies for load processes because you might include uncommitted or backed out changes in the target database.

For any given subscription, you must install and load at the same time all the following types of databases and structures:

- All areas for a single database
- All partitions for a high availability large database (HALDB)
- The primary index and all secondary indexes
- All logically related databases

If two databases in the same subscription are not logically related, you do not have to load them at the same time.

To set up a target IMS database for the first time:

Procedure

1. Install your database structures at the target site by sending copies of your database descriptions (DBDs) to the target site and defining them to the target IMS subsystem.

   The procedure is the same whether you are installing for the first time or reinstalling after a reorganization.
2. Perform an initial load at the target site for each database that you want to replicate.
   a. Create a nonconcurrent image copy of the data sets, areas, or partitions that contain the data in your source databases.
   b. Create and populate the data set replicas on the target site.
   c. Install or rebuild the primary and secondary index databases.

      You have these options:

      - Create image copies of the index databases and copy them to the target site when you transfer the database data sets.
      - Run an index builder utility at the target site after you load the target database. (This option is faster.)

Setting up a source server for change capture

Setting up a source server for change capture requires preparation tasks to customize, start, and validate the server.

Meeting requirements for universal character set (UCS) support

To enable the administration service and the monitoring service to use the z/OS Unicode environment, ensure that you load the required conversion tables for universal character set (UCS) support and that they are available.

About this task

The z/OS Unicode environment exists on every logical partition (LPAR) either because you explicitly configured the environment or because z/OS loaded the default environment. Confirm that the z/OS Unicode environment is available by issuing the z/OS operator command **D UNI**.

You specify the codepage for a Classic data server by setting the value of the **HOSTCODEPAGE** parameter for the region controller service. The default value is 37.

The administration service and monitoring service send and receive data in the following code pages:

Table 14. Code page conversions by service

Service	Code page
Administration service	UTF-8
Monitoring service	UCS-2

The z/OS operating system supports multiple techniques for codepage conversion, and a Classic data server uses the LER technique for search order (Language

environment, Enforced subset, Round trip). If UCS conversion tables are not available in your z/OS Unicode environment, ask your system programmer to define them.

The following conversion tables must be available:

- *Server codepage* to UTF-8 (1208)
- UTF-8 (1208) to *server codepage*
- *Server codepage* to UCS-2 (1200)
- UCS-2 (1200) to *server codepage*

For information about codepage conversion techniques and z/OS handling for codepage 1200, see *z/OS Support for Using Unicode: Using Unicode Services*. In z/OS version 12, see *Unicode Services User's Guide and Reference*

For information about updating conversion tables by using the **SETUNI** command or the **D UNI** command, see *MVS System Commands*.

Procedure

1. Issue the **D UNI** command to determine whether the z/OS Unicode environment is available on your LPAR.

   See the example for sample output.
2. Issue the following commands to verify whether the required conversion tables exist.

   If you changed the value of the **HOSTCODEPAGE** parameter, use that value in the command instead of 37.

   a. `D UNI,FROMID=37,TOID=1208`

   b. `D UNI,FROMID=1208,TOID=37`

   c. `D UNI,FROMID=37,TOID=1200`

   d. `D UNI,FROMID=1200,TOID=37`
3. If the required conversion tables are not available, ask your system programmer to define them by using one of the LER techniques.

Examples

This example shows one possible output for the **D UNI** command when you issue it without arguments:

```
D UNI
CUN3000I 08.20.58 UNI DISPLAY 398
 ENVIRONMENT: CREATED       03/02/2009 AT 06.10.27
              MODIFIED      04/29/2009 AT 14.54.15
              IMAGE CREATED --/--/---- AT --.--.--
```

The remaining examples show sample output for the **D UNI** command when you issue it with the FROMID and TOID arguments. The CONVERSION value indicates that the conversion table exists, and the trailing L, E, or R indicates the conversion technique.

Command processing substitutes 1200 for codepage 13488 because the z/OS Unicode environment has special handling for codepage 1200.

```
D UNI,FROMID=37,TOID=1208
CUN3000I 08.03.18 UNI DISPLAY 750
CONVERSION: 00037-01208-R 00037-01208-L
```

```
D UNI,FROMID=1208,TOID=37
CUN3000I 08.03.28 UNI DISPLAY 752
CONVERSION: 01208-00037-E 01208-00037-L

D UNI,FROMID=37,TOID=1200
CUN3000I 15.11.37 UNI DISPLAY 558
  CONVERSION: 00037-01200(13488)-ER

D UNI,FROMID=1200,TOID=37
CUN3000I 15.11.56 UNI DISPLAY 560
  CONVERSION: 01200(13488)-00037-ER
```

If the conversion table does not exist, the output is similar to the following example:

```
D UNI,FROMID=1200,TOID=99999
CUN3000I 08.07.14 UNI DISPLAY 846
CONVERSION: NO CONVERSIONS FOUND
```

Setting up the database resource adapter (DRA)

Data Replication for IMS requires the use of the database resource adapter (DRA) interface on the target server.

Before you begin

Before you set up the DRA interface, estimate the maximum count of parallel apply PSBs across all subscriptions that you plan to deploy on your target server. See the topic "Optimizing parallel apply processing."

About this task

Work with your IMS system administrator or system programmer to set up the DRA. The target server uses the IMS DRA interface to apply replicated updates to target databases.

The default load module for the DRA, DFSPZP00, is in the IMS.SDFSRESL library. The remainder of the DRA modules reside in a load library that is dynamically allocated by DFSPRRC0 (the startup). The default DDNAME and DSNAME of the load library (IMS.SDFSRESL) are specified in the default startup table DFSPZP00. DFSPZP00 contains default values for the DRA initialization parameters.

You must also configure a DRA service that references the DRA startup table in the target server.

Procedure

1. Code a new start-up table. Name it, for example, DFSPZP01. You might want to use DFSPZP00 as an example.
2. Specify the required values.
3. Copy any unchanged values from the default table.
   a. Set a value of at least 3 for MINTHRD.
   b. Set a value for MAXTHRD that is greater than the maximum count of parallel apply PSBs across all subscriptions on your target server and at least equal to the value of the MAXWRITERTHREADS configuration parameter for the apply service.
   c. Consider Fast path buffers. If you are replicating changes for DEDB databases you need to specify appropriate CNBA, FPBUF, and FPBOF parameters that allow your source UORs to be applied at the target. In these

cases, if you use the DEDB version of the bookmark database you probably do not need to increase these values since the buffers required for bookmark maintenance are small.

Use the following guidelines to calculate the required buffer settings:

- Allocate one fixed buffer (FPBUF) for bookmark access
- If you have subscriptions that require completed UOR tracking, allocate one overflow buffer (FPBOF) for bookmark access.
- The calculation for the total number of buffers (CNBA) required for bookmark access is `(FPBUF+FPBOF) * MAXTHRD`.

Apply these additional requirements to your own application usage calculations.

4. Assemble and link the new module into a load library. If the new module is named DFSPZP01, the IMS DRA initialization, DRATABLESUFFIX configuration parameter, specifies a value of 01 in the DRA startup-table suffix.

What to do next

For information about DFSPZP00 see the IMS documentation about installation and customization. For information about using the DRA start-up table to optimize performance, see the CICS documentation about performance.

To enable active connections for all writer threads, the MAXTHRD value specified in the DFSPZPxx member must be greater than or equal to the value specified for the WRITERTHREADS apply service configuration parameter

Starting a data server

When you start a data server, you start those services that are configured to start in the configuration file for the server.

About this task

You can start Classic data servers using either of the methods listed.

Note: You can run a source server despite whether IMS is running on the source LPAR. If IMS stops on the target LPAR, apply processing detects it and replication stops for all subscriptions.

Procedure

Start a data server by using one of the following methods:

- Issue a console command to start the JCL procedure for the data server.

  `S` *procname*

  procname
  The 1-8 character PROCLIB member name to be started. When you issue commands from the SDSF product, prefix all operator commands with the forward slash (/) character.
- Submit the JCL as a batch job.

Setting up a target server for apply processing

Setting up a target server for apply processing requires preparation tasks to set up access to IMS, and then customize, start, and validate the Classic data server.

Setting up access to IMS

To enable the target server to access IMS data, estimate the number of program specification blocks (PSBs) that you will need for apply processing, then configure and enable the interface for the IMS database resource adapter (DRA).

Optimizing parallel apply processing

To optimize parallel processing capacity, coordinate the maximum number of program specification blocks (apply PSBs) that the target server can schedule in IMS with the maximum total of apply PSBs that you specify for your subscriptions.

About this task

You can skip this topic if your subscription contains databases that participate in logical relationships with other databases. Subscriptions that have this type of structure can only perform serial apply.

The following information helps you to estimate the maximum count of parallel apply PSBs across all subscriptions in your replication environment, optimize parallel apply processing, and avoid hanging your subscriptions.

Scheduled PSBs in IMS and PSBs per subscription

You specify the maximum number of apply PSBs that IMS can schedule for the use of the target server in the **MAXTHRD** parameter when you set up the interface for the database resource adapter (DRA).

The value that you set for **Maximum Parallel Apply PSBs** when you create or modify a subscription determines the maximum number of apply PSBs that the subscription can use to apply changes in parallel.

Avoid setting too low a value for **MAXTHRD**. Set it to a value that is equal to or greater than the total of **Maximum Parallel Apply PSBs** across all subscriptions in the target server.

For example, if you have 5 subscriptions and each can schedule up to 5 PSBs, set **MAXTHRD** to 25 or higher. A lower value might hang one of the subscriptions if a writer thread tries to schedule an additional PSB that exceeds the **MAXTHRD** limit. In this case, IMS waits indefinitely for an additional DRA thread to become available.

Estimating the number of required PSBs

The number of parallel apply PSBs that your subscription requires depends on the amount of concurrent update activity against the target databases in the subscription. This amount varies from subscription to subscription, so consider these criteria when you estimate a value for **Maximum Parallel Apply PSBs**:

- Given the databases and records that your units of recovery (UORs) update, how much concurrent processing is possible?

  Data Replication for IMS must apply changes to the same record in order, so if your UORs repeatedly update the same records you will have less concurrent processing.
- How frequently do updates occur in parallel, and over how long a time interval?

  Do your UORs occur in a burst, or over an extended period of time? The number of applications that update your databases is also a factor.

These criteria apply to both batch update windows and online updates. If your updates occur during a batch update window, consider the volume of concurrent processing and the duration of the window.

Examples:

If the batch update window is three hours long and on average you run three batch message processing (BMP) programs, the subscription might require three parallel apply PSBs. If the number of BMPs varies or the window is shorter, then you might need fewer.

If on average you run only one BMP, or two BMPs in parallel that make relatively few updates, then the subscription might require only one apply PSB. If two concurrent BMPs generate many updates, or if you run them in a relatively short burst, then you might need two. If you have a large number of UORs that update a small number of different database records, then perhaps you need three apply PSBs.

Target server calculations

On startup, the target server calculates the maximum count of parallel apply PSBs across all subscriptions. The target server makes this calculation against the subscriptions that are active when it starts, so if you add or remove subscriptions later the calculation might not be correct.

When you start the target server, it issues the following event message:

`CECA0030I x Subscriptions loaded with a maximum total of y parallel PSBs`

Procedure

1. Before you set up the DRA interface, estimate the maximum count of parallel apply PSBs across all subscriptions that you plan to deploy on your target server.
2. Set up the DRA interface and specify a value for **MAXTHRD** that is greater than or equal to the maximum count of PSBs that you calculated in the previous step.
3. After you complete all configuration steps, start your target server and review message CECA0030I.
   a. Confirm that the value of **MAXTHRD** in the DRA startup table is greater than the maximum total of parallel PSBs across all subscriptions.
   b. If the value of **MAXTHRD** is too low, adjust it to optimize parallel apply processing and prevent the writer service from hanging.

Setting up the database resource adapter (DRA)

Data Replication for IMS requires the use of the database resource adapter (DRA) interface on the target server.

Before you begin

Before you set up the DRA interface, estimate the maximum count of parallel apply PSBs across all subscriptions that you plan to deploy on your target server. See the topic “Optimizing parallel apply processing.”

About this task

Work with your IMS system administrator or system programmer to set up the DRA. The target server uses the IMS DRA interface to apply replicated updates to target databases.

The default load module for the DRA, DFSPZP00, is in the IMS.SDFSRESL library. The remainder of the DRA modules reside in a load library that is dynamically allocated by DFSPRRC0 (the startup). The default DDNAME and DSNAME of the load library (IMS.SDFSRESL) are specified in the default startup table DFSPZP00. DFSPZP00 contains default values for the DRA initialization parameters.

You must also configure a DRA service that references the DRA startup table in the target server.

Procedure

1. Code a new start-up table. Name it, for example, DFSPZP01. You might want to use DFSPZP00 as an example.
2. Specify the required values.
3. Copy any unchanged values from the default table.
   a. Set a value of at least 3 for MINTHRD.
   b. Set a value for MAXTHRD that is greater than the maximum count of parallel apply PSBs across all subscriptions on your target server and at least equal to the value of the MAXWRITERTHREADS configuration parameter for the apply service.
   c. Consider Fast path buffers. If you are replicating changes for DEDB databases you need to specify appropriate CNBA, FPBUF, and FPBOF parameters that allow your source UORs to be applied at the target. In these cases, if you use the DEDB version of the bookmark database you probably do not need to increase these values since the buffers required for bookmark maintenance are small.

      Use the following guidelines to calculate the required buffer settings:

      - Allocate one fixed buffer (FPBUF) for bookmark access
      - If you have subscriptions that require completed UOR tracking, allocate one overflow buffer (FPBOF) for bookmark access.
      - The calculation for the total number of buffers (CNBA) required for bookmark access is `(FPBUF+FPBOF) * MAXTHRD`.

      Apply these additional requirements to your own application usage calculations.
4. Assemble and link the new module into a load library. If the new module is named DFSPZP01, the IMS DRA initialization, DRATABLESUFFIX configuration parameter, specifies a value of 01 in the DRA startup-table suffix.

What to do next

For information about DFSPZP00 see the IMS documentation about installation and customization. For information about using the DRA start-up table to optimize performance, see the CICS documentation about performance.

To enable active connections for all writer threads, the MAXTHRD value specified in the DFSPZPxx member must be greater than or equal to the value specified for the WRITERTHREADS apply service configuration parameter

Connecting to servers for replication and change data capture

You can use the Console Explorer view to create, edit, import, export, delete, and copy connections. Use connections as source and target servers for replication and change data capture.

About this task

When you create or modify a subscription, you can select a source and target server from a list of available connections that you will create or modify using this procedure. You connect to the servers in the Console Explorer view, which is visible in both the Replication perspective and the Data perspective. To open the Replication perspective, select **Window** > **Open Perspective** > **Replication**. To open the Data perspective, select **Window** > **Open Perspective** > **Data**.

The following tasks are available in the Console Explorer view:

- create new connections and edit their attributes
- delete existing connections
- export existing connections to a file for use in another workspace
- import connections from a file
- copy existing connections that exist in the Data Source Explorer view for use in replication

Procedure

To create a new connection:

- From the Console Explorer view, right-click **Connections** and select **New Connection**.
- In the New Operator Connection window, specify the following information:
  1. **Connection name:** A unique and meaningful name for the connection.
  2. **Host:** The full domain name or IP address of the host.
  3. **Port number:** Use the port number for the connection handler (the INIT service).
  4. **User ID:** Your TSO user name.
  5. **Password:** Your TSO password.
  6. **Save password:** If you leave this field cleared, you will be prompted for your password every time you re-establish connection to the server.

  When the new connection is successfully created, it is added to the Connections folder in the Console Explorer. You can view connection details in the Properties view.
- Do the same for the target server or target engine. When both connections have been made, you can create or edit subscriptions for these servers.

To edit a connection:

- From the Console Explorer view right-click the connection you want to edit and select **Edit Connection**.
- In the Edit Operator Connection window, change any of the fields that are displayed.

To copy a connection:

- From the Console Explorer view right-click Connections and select **Copy Connections**.
- In the Copy Operator Connections window, select one or more connections from the list of Data Source Explorer connections to copy.

To import a connection:

- From the Console Explorer view right-click Connections and select **Import Connections**.

- In the Import Connections window, specify a file with previously exported connections. If a connection already exists, its properties are updated based on the information in the file.

To export a connection:

- From the Console Explorer view right-click Connections and select **Export Connections**.
- In the Export Connections window, specify the following information:
  1. **Select the connections to be exported:** Select one or more connections to be exported.
  2. **To directory:** Destination directory for the new export connections file.
  3. **File name:** File name for the new export connections file.

Results

When you create replication subscriptions, the dialog detects whether a server contains a capture service or apply service and presents it as a candidate source server or target server.

Configuring for GDPS/Active-Active

Data Replication for IMS is a replication engine component of GDPS/Active-Active. To participate in GDPS/Active-Active, you configure the Classic data server to send Event Integration Facility (EIF) events to an event server.

About this task

To send an EIF event to an event server, the Classic data server establishes TCP/IP connections to the event server specified in the URL values of the EIFEVENTSERVERS service list entry defined for the logger service.

When establishing a connection, the Classic server first checks the configuration values when the logger service starts and attempts to open a connection. The server checks the configuration values again when sending an EIF event.

If the Classic data server fails to open a session or fails to send an Event Integration Facility (EIF) event to an event server, the Classic data server will attempt to reopen that session. If the attempt to reopen the session fails, the Classic data server will suspend EIF event processing to that URL for a fixed period of time (3 seconds).

The Classic data server will attempt to reopen the connection to the EIF event server when processing the first EIF event that occurs after the suspend period ends. If successful, normal EIF event processing resumes. If unsuccessful, the Classic data server will suspend EIF processing again.

You can add event server URLs to the service list entry or remove event server URLs from the service list entry. The Classic data server opens a connection when you add an entry or closes a connection when an entry is removed.

Any change to the URL value of the EIFEVENTSERVERS service list entry is recognized when processing the next EIF event, regardless of whether EIF processing is suspended.

Changing the URL value for the EIFEVENTSERVERS service list entry determines the action that the Classic server takes when sending an EIF event.

- If the URL value matches the non-null value specified for the current session, the Classic server opens the session if it is not already opened.
- If the value differs from the value specified for the current session, the server closes the current connection and opens a new connection to the event server identified by the URL value.
- If a URL value is not specified (that is, the EIFEVENTSERVERS service list is empty), the server closes the current connection and no new EIF events are sent to an event server.
- If a new URL value is specified, and no value was specified previously, the server attempts a new connection by using the new URL value.

If the EIFEVENTSERVERS service list is empty, the Classic server issues z/OS console messages instead of EIF events. The value of the latency threshold for a subscription determines when console messages are issued. You can use these console messages to determine when to automate an action when the corresponding EIF event occurs.

The following table lists the console messages and the corresponding EIF event class.

Table 15. Console messages and corresponding EIF events.

Console Message	**Corresponding Event Class**
CECA0137I The workload entered the replication continuous state SUB=*subscription-name* SRCSYSID=*source-system-id* DATASRC=*data-source* SUBSTATE=*subscription-state* WORKLOAD=*workload-name* WORKLOADTYPE=*workload-type* LATENCYSTATE=*latency-state* ABSBOOKMARK=*absolute-bookmark*	AA_replication_workload_up
CECA0138I The workload entered the replication inactive state SUB=*subscription-name* SRCSYSID=*source-system-id* DATASRC=*data-source* SUBSTATE=*subscription-state* WORKLOAD=*workload-name* WORKLOADTYPE=*workload-type* LATENCYSTATE=*latency-state* ABSBOOKMARK=*absolute-bookmark* WORKLOADDOWNMETHOD=*method*	AA_replication_workload_down

Table 15. Console messages and corresponding EIF events. (continued)

Console Message	Corresponding Event Class
CECA0139I The maximum latency threshold was exceeded SUB=*subscription-name* SRCSYSID=*source-system-id* DATASRC=*data-source* WORKLOAD=*workload-name* WORKLOADTYPE=*workload-type* LATENCYSTATE=*latency-state* MAXTIME=*maximum-latency-time* MEANTIME=*averaging-time* ABSBOOKMARK=*absolute-bookmark* LATENCY=*latency-time*	AA_replication_max_latency_exceeded AA_replication_max_average_latency_exceeded
CECA0140I The reset latency threshold was achieved SUB=*subscription-name* SRCSYSID=*source-system-id* DATASRC=*data-source* WORKLOAD=*workload-name* WORKLOADTYPE=*workload-type* LATENCYSTATE=*latency-state* RESETTIME=*reset-latency-time* MEANTIME=*averaging-time* ABSBOOKMARK=*absolute-bookmark* LATENCY=*latency-time*	AA_replication_reset_latency_met AA_replication_reset_average_latency_met
CECA0141I The critical latency threshold was exceeded SUB=*subscription-name* SRCSYSID=*source-system-id* DATASRC=*data-source* WORKLOAD=*workload-name* WORKLOADTYPE=*workload-type* LATENCYSTATE=*latency-state* CRITICALTIME=*critical-latency-time* MEANTIME=*averaging-time* ABSBOOKMARK=*absolute-bookmark* LATENCY=*latency-time*	AA_replication_critical_latency_exceeded AA_replication_critical_average_latency_exceeded
CECA0142I The constrained latency threshold was exceeded SUB=*subscription-name* SRCSYSID=*source-system-id* DATASRC=*data-source* WORKLOAD=*workload-name* WORKLOADTYPE=*workload-type* LATENCYSTATE=*latency-state* CONSTRAINEDTIME=*constrained-latency-time* MEANTIME=*averaging-time* ABSBOOKMARK=*absolute-bookmark* LATENCY=*latency-time*	AA_replication_constrained_latency_exceeded AA_replication_constrained_average_latency_exceeded

Table 15. Console messages and corresponding EIF events. (continued)

Console Message	Corresponding Event Class
CECA0143I The subscription heartbeat SUB=*subscription-name* SRCSYSID=*source-system-id* DATASRC=*data-source* SUBSTATE=*subscription-state* WORKLOAD=*workload-name* WORKLOADTYPE=*workload-type* LATENCYSTATE=*latency-state* HEARTBEAT=*heartbeat-time* MEANTIME=*averaging-time* COMMITS=*commits* ABSBOOKMARK=*absolute-bookmark* LATENCY=*latency-time*	AA_replication_heartbeat
CECA0144I The discrete latency threshold was exceeded SUB=*subscription-name* SRCSYSID=*source-system-id* DATASRC=*data-source* SUBSTATE=*subscription-state* WORKLOAD=*workload-name* WORKLOADTYPE=*workload-type* LATENCYSTATE=*latency-state* DISCRETETIME=*discrete-latency-time* ABSBOOKMARK=*absolute-bookmark* LATENCY=*latency-time* TRANSID=*transaction-id*	AA_replication_discrete_latency_exceeded

For detailed information about the data elements for each console messages, see "Event classes and attributes."

Procedure

Configure a URL to define the event server that receives EIF events. See EIFEVENTSERVERS..

Event classes and attributes

The body of an event contains an event class and the attributes that define the event class.

An event class defines a specific type of Event Integration Facility (EIF) event that a Classic data server sends to an event server.

Event classes

The Classic data server emits EIF events to an event server based on the subscription state, the subscription apply latency, and the heartbeat event. The following table lists each event class, a description of the event class, and the event attributes that define the event class.

Table 16. Event classes

Event class name	Description	Attributes
AA_replication_max_latency_exceeded AA_replication_max_average_latency_exceeded	Emitted when the value of the averaged or non-averaged end-to-end apply latency of a subscription exceeds the specified maximum apply latency threshold time.	adapter host apply_latency_time averaging_time cac_eventmsg cac_eventmsg_id consistency_group consistency_group_total credibility data_source job_name latency_state max_latency_time msg node origin point_in_time_consistency reserved01 - reserved05 severity source_system_id status sysplex_name system_name taskno time_stamp version workload_name workload_type
AA_replication_constrained_latency_exceeded AA_replication_constrained_average_ latency_exceeded	Emitted when the value of the averaged or non-averaged end-to-end apply latency of a subscription exceeds the constrained apply latency threshold time.	adapter_host apply_latency_time averaging_time cac_eventmsg cac_eventmsg_id consistency_group consistency_group_total constrained_latency_time credibility data_source job_name latency_state msg node origin point_in_time_consistency reserved01 - reserved05 severity source_system_id status sysplex_name system_name taskno time_stamp version workload_name workload_type

Table 16. Event classes (continued)

Event class name	Description	Attributes
AA_replication_critical_latency_exceeded AA_replication_critical_average_latency_exceeded	Emitted when the value of the averaged or non-averaged end-to-end apply latency of a subscription exceeds the critical apply latency threshold time.	adapter_host apply_latency_time averaging_time cac_eventmsg cac_eventmsg_id consistency_group consistency_group_total credibility critical_latency_time data_source job_name latency_state msg node origin point_in_time_consistency reserved01 - reserved05 severity source_system_id status sysplex_name system_name taskno time_stamp version workload_name workload_type
AA_replication_discrete_latency_exceeded	Emitted when the value of the non-averaged end-to-end apply latency of a subscription exceeds the discrete apply latency threshold time.	adapter_host apply_latency_time cac_eventmsg cac_eventmsg_id consistency_group consistency_group_total credibility data_source discrete_latency_time job_name latency_state msg node origin point_in_time_consistency reserved01 - reserved05 severity source_system_id status sysplex_name system_name taskno time_stamp version workload_name workload_type

Table 16. Event classes (continued)

Event class name	Description	Attributes
AA_replication_heartbeat	Emitted at the frequency established by the value of the heartbeat apply latency threshold time.	adapter_host apply_latency_time averaging_time cac_eventmsg cac_eventmsg_id commits consistency_group consistency_group_total credibility data_source heart_beat_time job_name latency_state msg node origin point_in_time_consistency reserved01 - reserved05 severity source_system_id status subscription_name sysplex_name system_name taskno time_stamp version workload_name workload_type
AA_replication_reset_latency_met AA_replication_reset_average_latency_met	Emitted when the value of the averaged or non-averaged end-to-end apply latency of a subscription falls below the reset apply latency threshold time, after previously exceeding the maximum apply latency threshold time.	adapter host apply_latency_time cac_eventmsg cac_eventmsg_id consistency_group consistency_group_total credibility data_source job_name latency_state msg node origin point_in_time_consistency reserved01 - reserved05 reset_latency_time severity source_system_id status sysplex_name system_name taskno time_stamp version workload_name workload_type

Table 16. Event classes (continued)

Event class name	Description	Attributes
AA_replication_workload_down	Emitted by the target server when a subscription transitions to the state SUB_STATE_REPL_INACTIVE	adapter host apply_latency_time cac_eventmsg cac_eventmsg_id consistency_group consistency_group_total credibility data_source job_name latency_state msg node origin point_in_time_consistency reserved01 - reserved05 severity source_system_id status subscription_state sysplex_name system_name taskno time_stamp version workload_down_method workload_name workload_type
AA_replication_workload_up	Emitted by the target server when a subscription transitions to the state SUB_STATE_CONTINUOUS.	adapter host apply_latency_time cac_eventmsg cac_eventmsg_id consistency_group consistency_group_total credibility data_source job_name latency_state msg node origin point_in_time_consistency reserved01 - reserved05 severity source_system_id status subscription_state sysplex_name system_name taskno time_stamp version workload_name workload_up_method workload_type

Event attributes

The following tables list the event attributes, a description of the attributes, and examples of attribute values.

Table 17. Event attributes

Attribute name	Description	Value
adapter_host	The host on which the adapter, the Classic source or target data server, is running.	*Adapter host*
averaging_time	The time across which apply latency times are averaged. • When the averaging time is zero, no averaging is performed and the event is emitted if a single instance exceeds a threshold. • When the averaging time is greater than zero, apply latency times are averaged across that time span. The apply latency times are then compared against the thresholds to determine whether or not to emit an event.	*Averaging time*
apply_latency_time	The end-to-end apply latency time.	*Apply latency time*
cac_eventmsg_id	The ID of the event message that the Classic data server emits from which the EIF event was adapted.	Event message ID. For example: *0x0101008C*
commits	The number of transactions in the averaged end-to-end apply latency sample set.	Number of committed transactions
consistency_group	The name of the consistency group. For Data Replication for IMS, this name is the same as the workload name.	*Consistency group name*
consistency_group_total	The total number of subscriptions in the consistency groups. For Data Replication for IMS, this total is 1.	1
constrained_latency_time	The latency threshold to which the apply latency time is compared to determine if a constrained latency exceeded event should be emitted..	*Constrained latency threshold time*
credibility	The credibility indicates how the adapter sends the event. The value 0 indicates that the event is sent from a non-Tivoli® Management Environment (TME) adapter.	0
critical_latency_time	The apply latency threshold time to which the apply latency time is compared to determine if a critical latency exceeded event should be emitted.	*Critical latency threshold time*
data_source	Indicates the type of data being replicated.	*IMS*
discrete_latency_time	The apply latency threshold time to which the apply latency time is compared to determine if a discrete latency exceeded event should be emitted.	*Discrete latency threshold time*

Table 17. Event attributes (continued)

Attribute name	Description	Value
heart_beat_time	The apply latency threshold heartbeat time.	*Heartbeat time*
job_name	The job name of the Classic data server where the event originated.	*Job name*
latency_state	Based on where the apply latency time falls in relation to the setting of the apply latency threshold values, the latency state can be one of the following: • Normal • Constrained • Critical • Acute	*Latency state*
max_latency_time	The apply latency threshold time to which the apply latency time is compared.	*Maximum latency threshold time*
msg	A message that describes the event in text format.	This message varies depending on the event. Possible values: • The replication workload is up. • The replication workload is down. • The replication workload's end-to-end apply latency has exceeded its maximum apply latency threshold. • The replication workload's average end-to-end apply latency has exceeded its maximum apply latency threshold. • The replication workload's end-to-end apply latency has fallen below its reset apply latency threshold. • The replication workload's average end-to-end apply latency has fallen below its reset apply latency threshold. • The replication workload's end-to-end apply latency has exceeded its critical apply latency threshold. • The replication workload's average end-to-end apply latency has exceeded its critical apply latency threshold. • The replication workload's end-to-end apply latency has exceeded its constrained apply latency threshold. • The replication workload's average end-to-end apply latency has exceeded its constrained apply latency threshold. • The replication workload's apply latency heartbeat. • The replication workload's end-to-end apply latency has exceeded its discrete apply latency threshold.
node	The ASID of the Classic data server.	ASID. For example: *1300*

Table 17. Event attributes (continued)

Attribute name	Description	Value
origin	The IP address or host name of the source system.	IP address. For example: *10.1.1.2*
point_in_time_consistency	The current point-in-time of the data source.	The absolute bookmark of the subscription in the format: *yyyy-mm-dd-hh.mm.ss.uuuuuu*. For example: *2012-02-28-13.42.53.969000*.
reserved01	Reserved	
reserved02	Reserved	
reserved03	Reserved	
reserver04	Reserved	
reserved05	Reserved	
reset_latency_time	The apply latency threshold time to which the apply latency time is compared to determine if a reset latency met event should be emitted.	*Reset latency threshold time*
severity	The severity of the event. The value INFO indicates that the severity of the event is informational.	INFO
source	The source of the event that defines the adapter type.	Replication engine adapter.
source_system_id	The ID that the source server assigns to the subscription.	*Source system ID*
status	The severity of the event. The value OPEN indicates that the severity of the event is open.	OPEN
subscription_name	The name of the subscription.	Subscription name. For example: *PAYROLL*
subscription_state	The state of the subscription.	SUB_STATE_REPL_CONTINUOUS or SUB_STATE_INACTIVE
sysplex_name	The name of the sysplex on which the event occurred.	*Sysplex name*
system_name	The name of the system on which the event occurred.	*System name*
taskno	The task number indicates the TCB address of the service task where the event originated.	Task number. For example: *9001272*
time_stamp	The time at which the replication engine generated the event.	Timestamp, in the format: *yyyy-mm-dd-hh.mm.ss.uuuuuu*. For example: *2012-02-28-13.42.53.969000*
transaction_id	The identification of the transaction whose end-to-end apply latency time exceeded the discrete latency threshold.	Transaction ID. For example: *'000000000001D730IM1A 54bc6d3f'*
version	The version of the EIF event.	Version. For example: *1*.
workload_down_method	Identifies why the AA_replication_workload_down was entered.	Operator or Failed.

Table 17. Event attributes (continued)

Attribute name	Description	Value
workload_name	The workload name associated with the apply latency threshold time to which the apply latency time is compared. • When the Classic data server does not participate in GDPS/Active-Active, the workload name associated with the first threshold set is the same as the subscription name. No workload name is associated with the second and third threshold sets. • When the Classic data server participates in GDPS/Active-Active, the workload name is the value assigned by the GDPS/Active-Active policy that is associated with the subscription.	*Workload name*
workload_type	The type of workload.	Possible values: • Active/Standby • Active/Query

For additional information about event classes and attributes, see the product information for Tivoli Enterprise Console.

Event formats

The format of an Event Integration Facility (EIF) event consists of a header, the event body, and an ending sequence.

The Classic data server sends EIF events to an event server in the ASCII (ISO-8859-1) code set. The structure of an event message includes the following elements:

header

The event header identifies the start of the event and the format of the event information.

```
HEADER
Offset Data        Description
   0   c'<START>>' Beginning of header, character sequence denoting the start of the event
   8   x'00000000' Message ID (not used)
   C   x'00000000' Message "from" (not used)
  10   x'00000000' Message "to" (not used)
  14   x'00000000' Message type (not used)
  18   x'00000000' IPC message type (not used)
  1C   x'nnnnnnnn' Message length (includes the length of data from offset x'24' to the
                   end of the event, including the END sequence)
  20   x'00000024' Header data length (some senders fill this in, but typically not used)
  24               Beginning of the data
```

body The body of the event message contains data that describes the event.

The format of the body consists of an event class name followed by the attributes of that event. The *event class* is a classification of an event that describes the attributes of the event to the event server. Each attribute is in the format *attribute=value*. The event class name and the attributes are delimited by a semi-colon (;).

For example, the event class `replication_restored` contains a specific set of attribute values, such as `subscription_name=STRING`, that indicate how the event is sent to the event server.

end

The ending sequence identifies the end of the event message.

```
END sequence
(offset n + 24 - 5)
       c'END'      Character sequence denoting the beginning of the end.
       x'0A01'     Last two bytes
```

Examples of apply latency thresholds and EIF events

The examples in this topic illustrate how to set workload names for apply latency thresholds and the EIF events associated with apply latency thresholds.

Setting workload names for of apply latency thresholds

The workload names that you define for apply latency thresholds represent GDPS/ActiveActive active/standby workloads:

- Threshold set one represents the GDPS/ActiveActive active/standby workload. This threshold set is named when any of its thresholds are set to non-zero values.
- Thresholds sets two and three represent GDPS/ActiveActive active/query workloads. **Important:** The workload name that you define for these threshold sets must be unique.

The workload name of threshold set one is cleared when all of its threshold values are zeros. The workload names of threshold sets two and three can be cleared only when all the threshold values are zeros.

The following examples illustrate how to set workload names for apply latency thresholds.

Example 1: Setting a workload name

```
SET,REPL,SUBSCR=SUB1,THRESHOLDS2,WORKLOAD=INVENTORY
SET,REPL,SUBSCR=SUB1,THRESHOLDS3,WORKLOAD=COSTOFGOODSOLD
```

In this example, the workload names are set for both threshold sets two and three.

Example 2: Clearing a workload name

```
SET,REPL,SUBSCR=SUB1,THRESHOLDS3,WORKLOAD=""
```

In this example, the workload name for threshold set three is cleared.

Example 3: Displaying no workload names

```
DISPLAY,REPL,SUBSCR=TS,THRESHOLDS
18.11.43.087000 CAC00200I DISPLAY,REPL,SUBSCR=TS,THRESHOLDS
18.11.43.087000 CECM0058I SUBSCRIPTION THRESHOLDS REPORT
SrcSysID Subscription Name
======== ============================================================
TS       TS
No Thresholds for Subscription
Number of source subscriptions reported: 1
```

This example displays a subscription when no workload name or threshold values are set.

Example 4: Displaying any workload names from the source server

```
DISPLAY,REPL,SUBSCR=TS,THRESHOLDS
```

```
CAC00200I DISPLAY,REPL,SUBSCR=TS,THRESHOLDS
CECM0058I SUBSCRIPTION THRESHOLDS REPORT
SrcSysID Subscription Name
======== ============================================================
ORDRENT  ORDRENT
  Thresholds1
    ORDRENT
    Heartbeat        10000 (ms)              Meantime      10000 (ms)
    Maximum            251 (ms)
    Reset              191 (ms)
    Critical           225 (ms)
    Constrained        200 (ms)
    Discrete          2001 (ms)
  Thresholds2
    INVENTORY
    Heartbeat        10000 (ms)              Meantime      10000 (ms)
    Maximum            258 (ms)
    Reset              192 (ms)
    Critical           232 (ms)
    Constrained        206 (ms)
    Discrete          2002 (ms)
  Thresholds3
    COSTOFGOODS
    Heartbeat        10000 (ms)              Meantime      10000 (ms)
    Maximum            358 (ms)
    Reset                0 (ms)
    Critical           322 (ms)
    Constrained        286 (ms)
    Discrete             0 (ms)
Number of source subscriptions reported: 1
END OF REPORT
```

This example displays threshold workload names and values from the source server. It shows the values of all of the configured threshold values and workload names for the subscription.

Example 5: Displaying any workload names from the target server

```
DISPLAY,REPL,SUBSCR=TS,THRESHOLDS
```

```
CAC00200I DISPLAY,REPL,SUBSCR=TS,THRESHOLDS
CECM0058I SUBSCRIPTION THRESHOLDS REPORT
SrcSysID Subscription Name
======== ============================================================
TS       TS
  Thresholds1             Latency State: Normal
    ORDRENT
    Heartbeat        10000 (ms)              Meantime      10000 (ms)
    Maximum            251 (ms)           4  2012-11-23-11.57.09
    Reset              191 (ms)           4  2012-11-23-12.01.22
    Critical           225 (ms)           0
    Constrained        200 (ms)           0
    Discrete          2001 (ms)           1  2012-11-23-23.59.02
  Thresholds2             Latency State: Normal
    INVENTORY
    Heartbeat        10000 (ms)              Meantime      10000 (ms)
    Maximum            258 (ms)           0
    Reset              192 (ms)           0
    Critical           232 (ms)           3  2012-10-24-15.12.29
    Constrained        206 (ms)           2  2012-10-24-15.11.10
    Discrete          2002 (ms)           2  2012-10-24-12.32.36
  Thresholds3             Latency State: Normal
    COSTOFGOODS
    Heartbeat        10000 (ms)              Meantime      10000 (ms)
```

```
    Maximum            358 (ms)           0
    Reset                0 (ms)           0
    Critical           322 (ms)           0
    Constrained        286 (ms)           0
    Discrete             0 (ms)           0
Number of target subscriptions reported: 1
END OF REPORT
```

This example displays the following information from the target server:

- Threshold values
- Workload names
- Count of the number of times the threshold was breached
- The time the threshold was last breached

Apply latency thresholds and associated EIF events

The following examples illustrate settings for apply latency thresholds and the associated EIF events.

Example 6: DISCRETETIME threshold

```
SET,REPL,SUBSCR=SUB1,THRESHOLDS1,DISCRETETIME=10000
SET,REPL,SUBSCR=SUB1,THRESHOLDS1,MEANTIME=0
```

In this example, apply latency values are not averaged. The Classic data server issues the following EIF event:

AA_replication_discrete_latency_exceeded
: Issued when the apply latency for each transaction is greater than 10000 milliseconds.

Example 7: MAXTIME threshold

```
SET,REPL,SUBSCR=SUB1,THRESHOLDS,MAXTIME=5000
SET,REPL,SUBSCR=SUB1,THRESHOLDS,RESETIME=2000
SET,REPL,SUBSCR=SUB1,THRESHOLDS,MEANTIME=5000
```

In this example, apply latency values are averaged across a 5 second (5000 millisecond) time period. The Classic data server issues the following EIF events:

AA_replication_max_average_latency_exceeded
: Issued after the averaged apply latency value is greater than 5 seconds (5000 milliseconds).
  - If the averaged apply latency falls below the MAXTIME threshold, while remaining above the RESETTIME threshold, and then exceeds MAXTIME again, this event is not issued.
  - If the averaged apply latency falls below both the MAXTIME and the RESETTIME thresholds, and then exceeds MAXTIME again, this event is issued.

AA_replication_critical_average_latency_exceeded
: Issued after the averaged apply latency value is greater than 4.5 seconds (4500 milliseconds).
  - If the averaged apply latency falls below 90% of the MAXTIME threshold, and then exceeds it again, this event is not issued.
  - If the averaged apply latency falls below both 90% of the MAXTIME and the RESETTIME thresholds, and then exceeds 90% of MAXTIME again, this event is issued.

AA_replication_constrained_average_latency_exceeded
Issued after the averaged apply latency value is greater than 4 seconds (4000 milliseconds).

- If the averaged apply latency falls below 80% of the MAXTIME threshold, and then exceeds it again, this event is not issued.
- If the averaged apply latency falls below both 80% of the MAXTIME and the RESETTIME thresholds, and then exceeds 80% of MAXTIME again, this event is issued.

Example 8: MAXTIME threshold

```
SET,REPL,SUBSCR=SUB1,THRESHOLDS,MAXTIME=5000
SET,REPL,SUBSCR=SUB1,THRESHOLDS,RESETIME=2000
SET,REPL,SUBSCR=SUB1,THRESHOLDS,MEANTIME=0
```

In this example, apply latency values are not averaged. The Classic data server issues the following EIF events:

AA_replication_max_latency_exceeded
Issued after the apply latency value is greater than 5 seconds (5000 milliseconds).

- If the apply latency falls below the MAXTIME threshold, and then exceeds MAXTIME again, this event is not issued.
- If the apply latency falls below both the MAXTIME and the RESETTIME thresholds, and then exceeds MAXTIME again, this event is issued.

AA_replication_critical_latency_exceeded
Issued after the apply latency value is greater than 4.5 seconds (4500 milliseconds).

- If the apply latency falls below 90% of the MAXTIME threshold, and then exceeds it again, this event is not issued.
- If the apply latency falls below both 90% of the MAXTIME and the RESETTIME thresholds, and then exceeds 90% of MAXTIME again, this event is issued.

AA_replication_constrained_latency_exceeded
Issued after the apply latency value is greater than 4 seconds (4000 milliseconds).

- If the apply latency falls below 80% of the MAXTIME threshold, and then exceeds it again, this event is not issued.
- If the apply latency falls below both 80% of the MAXTIME and the RESETTIME thresholds, and then exceeds 80% of MAXTIME again, this event is issued.

Example 9: RESETTIME threshold

```
SET,REPL,SUBSCR=SUB1,THRESHOLDS1,MAXTIME=5000
SET,REPL,SUBSCR=SUB1,THRESHOLDS1,RESETTIME=1000
SET,REPL,SUBSCR=SUB1,THRESHOLDS1,MEANTIME=3000
```

In this example, apply latency values are averaged across a 3 second (3000 milliseconds) time period. The Classic data server issues the following EIF events:

AA_replication_reset_average_latency_met
Issued after the averaged apply latency falls below 1 second (1000 milliseconds). If the apply latency exceeds the RESETTIME threshold, and then falls below it again, this event is not issued.

AA_replication_max_average_latency_exceeded
If the averaged apply latency exceeds both the RESETTIME and MAXTIME thresholds, and then falls below both MAXTIME and RESETTIME, this event is issued.

Example 10: RESETTIME threshold

```
SET,REPL,SUBSCR=SUB1,THRESHOLDS1,MAXTIME=5000
SET,REPL,SUBSCR=SUB1,THRESHOLDS1,RESETTIME=1000
SET,REPL,SUBSCR=SUB1,THRESHOLDS1,MEANTIME=0
```

In this example, apply latency values are not averaged. The Classic data server issues the following EIF events:

AA_replication_reset_latency_met
Issued after the apply latency falls below 1 second (1000 milliseconds). If the apply latency exceeds the RESETTIME threshold, and then falls below it again, this event is not issued.

AA_replication_max_latency_exceeded
If the apply latency exceeds both the RESETTIME and MAXTIME thresholds, and then falls below both MAXTIME and RESETTIME, this event is issued.

Example 11: HEARTBEAT threshold

```
SET,REPL,SUBSCR=SUB1,THRESHOLDS1,HEARTBEAT1=60000
```

In this example, apply latency values are averaged across the heartbeat interval of 1 minute (60000 milliseconds). The Classic data server issues the following EIF event:

AA_replication_heartbeat
Issued on the 1 minute interval.

Creating new subscriptions

Before you create subscriptions, you must understand basic replication concepts such as log reading, validation, and apply processing. Then plan your subscription deployment by using design principles that optimize replication for your site.

Subscription processing overview

You manage replication by creating *subscriptions*. Each subscription contains replication mappings that identify the databases that you want to replicate as a consistent unit between the source and target sites.

Subscriptions are independent entities that you start, stop, and monitor in your replication environment. You can deploy multiple subscriptions to optimize throughput and maintain application boundaries. If an outage occurs in one application, the remaining applications can continue to replicate independently.

A subscription also specifies a target uniform resource locator (URL) for communication between the source and target servers. If you mark a subscription as persistent (the default), replication restarts for that subscription automatically when you start the Classic data server. If you stop a persistent subscription manually, you must restart it manually.

Manage subscriptions by using the Classic Data Architect client application. With the Classic Data Architect, you connect to the source and target servers to access

database information. You then select source databases to create replication mappings for your subscription. After you define the subscriptions, you can start and stop replication from the Classic Data Architect.

When you start replication, the source server automatically performs a Describe process, which sends information about your subscriptions and replication mappings to the target server. This process ensures that your subscription definitions, replication objects, and replication mappings are in a synchronized state when replication begins.

Reading IMS logs

The IMS log reader service, which runs in the source server, captures change data from any registered DB/DC or DBCTL subsystem in the RECON data sets that the source server accesses.

It does this by processing active and archived IMS logs. The log reader service can also capture changes from registered DL/I batch subsystems in the RECON. The service then sends change messages to the capture service for processing. The change messages include log records that contain both change data and sync points.

Overview

The following diagram shows how IMS writes log data first to recent online logs, then to archive logs.

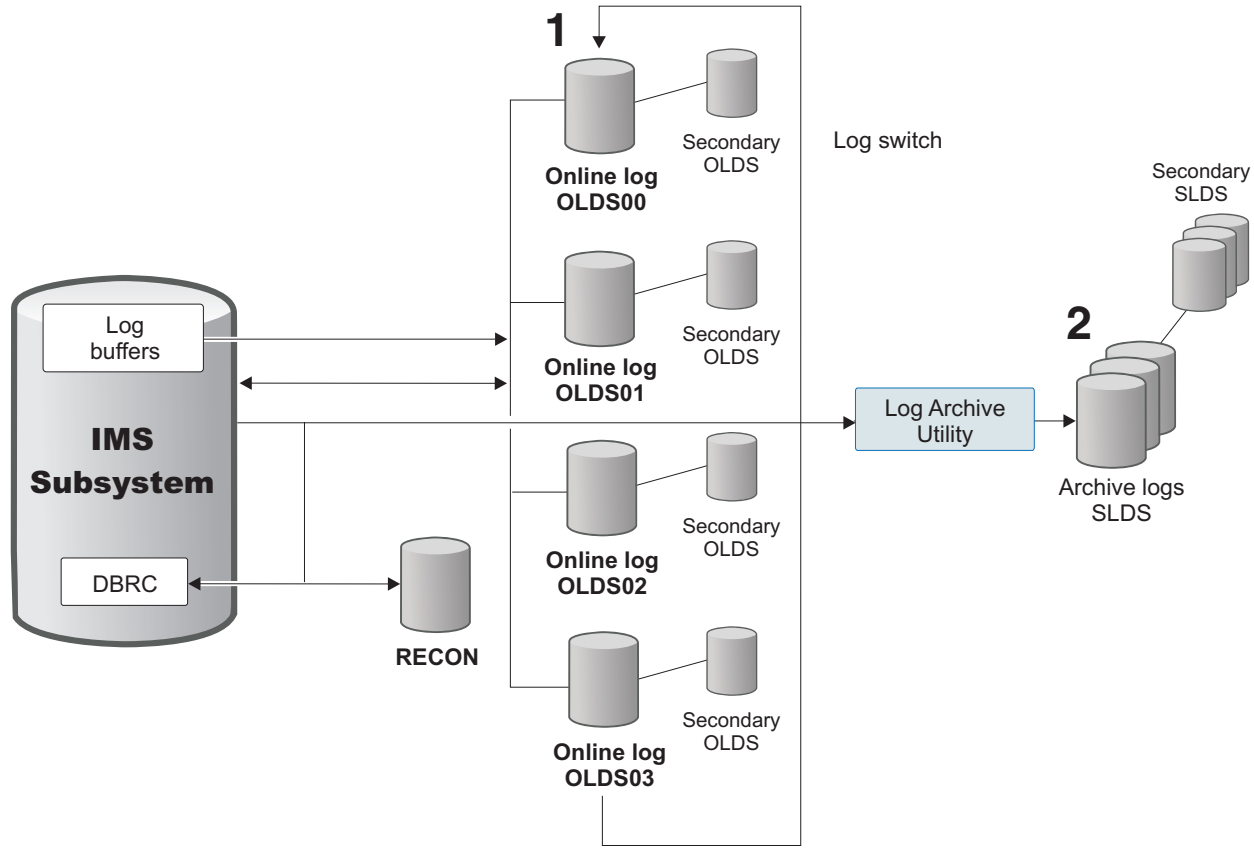


Figure 7. An IMS logging environment

Your IMS database administrator designs a logging environment that can sustain the logging rates and recovery requirements of the subsystem. The design deploys these sets of logs:

- Online data sets (OLDS) called *online logs*, one of which is the *active log* to which IMS is currently writing data
- Secondary OLDS, to which IMS writes data simultaneously with the OLDS
- System log data sets (SLDS) called *archive logs*, where IMS stores recovery data
- Secondary SLDS, to which IMS writes data simultaneously with the SLDS
- Recovery data sets that the IMS recovery utilities use

  The IMS log reader service does not process the recovery data sets, because they do not contain data capture log records.

The database administrator creates a fixed number of online logs for temporary storage of recovery data. When the active log is full, IMS automatically switches to the next online log in sequence. You can also force a log switch manually by using a command. Typically, the IMS Log Archive Utility transfers the data from the active log that is full to the archive logs when the log switch occurs.

If IMS marks a primary OLDS or SLDS in error, the log reader switches to reading the secondary logs.

IMS Database Recovery Control (DBRC)

DBRC stores information about all IMS logging activities in recovery control (RECON) data sets. DBRC performs key supporting activities for the IMS log reader service,, including the following:

- Identifying IMS subsystems whose changes are eligible for change data capture
- Tracking the log files that the online regions produce
- Identifying logs that IMS is currently using

When the IMS log reader service is actively capturing changes, it uses the DBRC API to determine which IMS log to process. The capture service provides the date and time where processing begins, and the log reader service generates a DBRC report to identify the log that spans this point in time.

The log reader service also uses DBRC reports to track IMS log usage and respond to log switches.

For more information about IMS logging and DBRC, see *An Introduction to IMS*.

Log reuse detection

In rare situations, IMS can cycle through the active logs and begin writing new changes at the log position where the log reader service is still processing changes from the previous cycle. If the log reader service encounters future log records that are out of sequence, it switches to reading the archive logs.

Active point detection

If the log reader service encounters a physical end-of-file (PEOF) marker from a previous log switch, it is possible that IMS will overwrite this intermediate end-of-file condition with new data. The log reader service knows how many blocks exist in an OLDS, and tracks the number of blocks that it reads. When the

service detects a physical end-of-file and has read all the blocks in a log, it accesses DBRC to determine the next file to process.

The log reader service can detect when it has reached the active point in the log where IMS is writing changes when it encounters log records with timestamps that are in the past relative to the previous block that the service processed. In this case, it suspends processing and waits for IMS to write new records.

Change streams:

A log reader reads log records for your data source and forwards them on *change streams* to the capture service.

A change stream consists of a sequence of log records that contain captured data and sync points. Sync points specify the end of a unit of recovery (UOR).

IMS data sources

A change stream performs these essential change-capture functions in Data Replication for IMS:

- Minimizes concurrent access to the physical logs that store the changes
- Merges changes from all participating IMS subsystems in the correct order
- Makes the changes available to the source server for retrieval and processing

When you start replication for the first subscription, the source server deploys a *primary stream* to transfer changes for all subscriptions on the source server. The bookmark setting for each subscription determines the point in time to begin processing, which continues with the goal of catching up to current processing.

In Data Replication for IMS, the capture service can also initiate *secondary streams* of limited duration to enable subscriptions that fall behind to catch up to current processing. You might notice tape mounts when you start replication or when subscriptions are catching up, indicating that the log reader service is accessing historical logs. If all available secondary change streams are in use, the source server might temporarily suspend change data capture for a subscription.

In situations where the restart position for a subscription is earlier than the oldest record available in the caches, an IMS subscription relies on historical logs and change streams to resume current processing. By design, primary and secondary change streams enable the source server to give priority to subscriptions with the oldest restart position and to subscriptions that are catching up.

Subsystem inactivity management:

If activity slows down in participating IMS subsystems, ensure that ordering operations can continue in your replication environment.

In an IMS data sharing environment where you are capturing changes from multiple DB/DC or DBCTL subsystems, one or more subsystems can stop producing new log records for an extended period of time. Data Replication for IMS cannot remove inactive subsystems from ordering decisions, because the source server must receive activity from all of them to avoid replicating changes out of order.

In some cases, an IMS subsystem does not produce enough records to fill a buffer, and does not physically write the data to the online log data sets (OLDS) until the buffer is full. However, you can use automation or manual intervention to flush the buffers and ensure that the source server receives a steady feed of changes from each participating subsystem.

To help you address this requirement, the source server issues the CECZ0400W message to identify the subsystem that became inactive. The user-configurable parameter **INACTTHRESHOLD** determines how frequently this message is displayed.

Validation of the replication environment

Data Replication for IMS validates that your environment is ready to start replication.

Before you can start replication successfully, you must resolve any configuration problems or inconsistencies in a subscription that can prevent it from starting. The Classic Data Architect provides the **Validate replication mappings** function to help you troubleshoot and resolve errors when you are setting up your replication environment.

Validation processing enables you to fix subscription errors in the setup stage if you do not want to wait until you start replication. For example, you can try to address all the problems with a subscription that can be detected so that the subscription is more likely to start with the other subscriptions. The validation process might flag errors that you prefer to fix later, such as a source database that you have not yet augmented or a target database that you have not yet created.

The source server can also detect schema changes while replication is active.

Subscription validation

IMS replication validates your subscription, replication mappings, and underlying data structures to help you to identify potential problems while you are setting up your replication environment.

The validation process compares the structure of the source and target DBDs in the subscription, and identifies configuration problems or inconsistencies in subscription metadata that can prevent replication from starting:

- A source database with incorrect augmentation
- A target database that does not exist
- A mismatch between source and target databases

When you run the subscription wizards to create or modify replication mappings, the Classic Data Architect also plays a role in exchanging information about the database descriptions (DBDs).

Validation processing continues to run to completion, regardless of whether it encounters one or more of these errors.

Validation at run time

When you start replication for a subscription, Data Replication for IMS performs strict validations of the database descriptions (DBDs) and the program specification block (apply PSB) for any replication mappings that require it.

Data Replication for IMS validates your replication environment regardless of which of the following methods you use to start the subscription:

- Start replication in the Classic Data Architect
- Issue a START,REPL command to the source server
- Mark a subscription as persistent and restart replication automatically

In addition, the target server schedules the apply PSB for the subscription and validates that a PCB exists in the apply PSB for each target database in an active replication mapping. It also validates access to the bookmark database.

Support for recursion protection

When you start replication for a subscription, the Classic Data Architect updates the apply PSB name at the target site. To prevent recursion, the target server also sends the apply PSB name back to the source server. If your environment replicates changes in both directions between the two sites, each site must filter out changes made by the other site to avoid replicating the same changes in a continuous loop.

Subscription validation:

When you create or modify a subscription, ensure that the source and target servers are running and can communicate, so that they can connect and exchange information about the subscription and its replication mappings.

When you create, modify, or start a subscription, the source server sends metadata to the target server. This metadata transfer, or *describe* process, synchronizes information between the two Classic data servers about your subscriptions and their respective replication mappings. The Classic Data Architect also plays a role in communicating information about database descriptions (DBDs). You can initiate the describe process by using the subscription wizards in the Classic Data Architect or by issuing a **START,REPL** command:

```
F job-name,START,REPL,SUBSCR=subscription-name,DESCRIBE
```

The describe process compares the structure of the source and target DBDs in the subscription, and identifies configuration problems or inconsistencies in the database metadata that can prevent replication from starting for the subscription:

- Source database descriptions (DBDs) without correct augmentation
- Target DBDs that do not exist
- Data mismatches between source and target databases

The describe process completes processing regardless of whether it encounters one or more of these errors.

Validation at run time:

When you start replication for a subscription, your environment performs strict validations of the database descriptions (DBDs) and the program specification block (apply PSB).

Validations when you start replication

IMS replication performs these validations regardless of the method that you use to start the subscription:

- Starting replication in the Classic Data Architect

- Issuing a START command to the source server
- Marking a subscription as persistent and initiating restart processing automatically

Runtime validations ensure that, before replication begins, you resolve any errors that the describe process discovered when you created or modified the subscription.

In addition, the target server schedules the apply PSB for the subscription and validates that a PCB exists in the apply PSB for each target database in an active replication mapping. It also validates access to the bookmark database.

Support for recursion protection

When you start replication for a subscription, the Classic Data Architect updates the apply PSB name at the target site. To prevent recursion, the target server also sends the apply PSB name back to the source server. If your environment replicates changes in both directions between the two sites, each site must filter out changes made by the other site to avoid replicating the same changes in a continuous loop.

DBD validation:

When you start replication, Data Replication for IMS validates that source and target database descriptions (DBDs) are consistent.

If differences are detected, the setting of the apply service STRICTVALIDATION configuration parameter controls whether replication is allowed to start. By default, replication is not allowed to start if differences are detected. See Synchronization of source and target IMS databases for information about the types of differences that can exist during replication and will not prevent replication from functioning properly.

DBD structures

The validation process uses the following information to validate DBDs:

- The name of the DBD
- The number of segments in the DBD

Segment structures

The validation process uses the following information to validate DBD segments:

- The name of the segment
- The segment code assigned to the segment, based on its location in the hierarchy
- The parent segment code
- The hierarchical level of the segment
- The maximum segment length
- The minimum segment length
- Bytes 1 and 2 of the segment flag, including the Insert HERE specification and SDEP indicators
- For DEBD databases, the number of subset pointers

Sequence fields

The validation process uses the following information to validate sequence fields:

- Starting offset
- Field length
- Whether the sequence field is unique

Apply PSB validation:

Data Replication for IMS also validates the program specification block (apply PSB) for each subscription.

The target server schedules the apply PSB before starting replication, and replication can start only if the PSB is valid. The target server performs validation of the contents of the apply PSB that includes:

- A PCB exists that references the bookmark database with the setting `PROCOPT=A`
- A PCB exists for each database that is active in the subscription

If you create the apply PSB for a subscription manually, ensure that each database PCB includes SESEG definitions for each segment in the database. If you plan to replicate subset pointers, ensure that, for each segment that has subset pointers, a SSPTR parameter is included that allows update operations to be performed for all subset pointers.

Using Classic Data Architect to generate the apply PSB for a subscription ensures that the all required PCB and SENSEG statements are included based on the databases defined for the subscription at the time that the PSB is generated.

Apply processing

Within a UOR, changes are applied at the target based on the order they are presented in the data capture log records produced by the source IMS.

For insert, delete, and update operations this sequence corresponds to the order performed by the source application. If the source application used a path call to insert or update one of more segments in a hierarchic path, these operations result in individual operations for each segment affected at the target.

The apply processing method that Data Replication for IMS™ uses depends on the IMS version that is installed. In IMS Version 13, enhancements to apply processing include:

- FLD call support
- Subset pointer replication (described in the topics “Target database positioning approaches” on page 122 and “Subset pointer replication” on page 123)
- Insert HERE positioning support (described in the topics “Target database positioning approaches” on page 122 and “Insert HERE positioning” on page 123)

FLD call support

If you have IMS Version 13 installed at the source and the appropriate IMS APARs installed, you can enable capture of updates made to your database when your applications issue FLD calls. They are treated as update (REPL) calls at the target and IMS produces one or more data capture log records immediately proceeding

the commit log records for the source UOR. Regardless of when the source application issues a FLD call, these updates are applied last at the target.

Parallel apply can improve the performance of your replication environment. *Adaptive apply* manages data conflicts that can arise between source and target databases.

Target database positioning approaches:

The apply processing method that Data Replication for IMS uses depends on the IMS version.

In prior releases of Data Replication for IMS, apply processing used a scanning approach when processing changes for segments without a unique sequence field. For these kinds of segments, the apply process generates qualified SSAs that position to the parent of the unkeyed segment and then issues a get unique call followed by one or more get next calls until a matching segment image is located. This process can be performed more that once if there are multiple non-uniquely keyed segments in a hierarchic path.

IMS Version 13 extends data capture capabilities to allow replication of subset pointers and to better support the ability for Data Replication for IMS to replicate non-uniquely keyed segments that use an insert rule of HERE. In particular, to support subset pointer replication for certain operations (most notably conditionally setting a subset pointer), the existing scanning approach that Data Replication for IMS uses is not adequate. A different approach is used to perform database positioning that relies on the use of SSAs that use the O command code.

Support for the O command code was added in IMS Version 12. Data Replication for IMS uses it to locate the first instance of a non-uniquely keyed segment by providing the entire contents of the source segment image and instructing IMS to look for a match. This allows Data Replication for IMS to build a set of qualified and unqualified SSAs that can be used to position in the target database using a single DL/I call.

You activate the new positioning method by using a new IMSV11CMPAT apply service configuration parameter. When IMSV11CMPAT is set to TRUE, the assumption is that the target site is running IMS Version 11 so that SSAs containing the O command code cannot be used and the old scanning approach must be used.

Setting IMSV11CMPAT to FALSE (the default) activates the new processing. When the new apply processing is active, the following optimizations are enabled:

- Direct insert of segment images that do not include positioning information or reference a segment with a unique sequence field with the DBLE pointer option.
- For uniquely keyed segments with the DBLE pointer option, an initial GHU call is issued using qualified SSAs and the last command code for the target segment (to position at the end of the twin chain), prior to inserting the new segment image.
- For positioning calls, current key feedback information is analyzed to determine whether a GHN call can be issued in preference to a GHU call.

As mentioned previously, IMS Version 13 allows information to be included in the data capture records to allow subset pointer updates to be captured and replicated and includes insert rule HERE enhancements. To process these log records, the new SSA generation logic must be active. This means that IMSV11CMPAT must be

set to FALSE. Attempts to process log records containing subset pointer information, or insert positioning information using the scanning approach, results in unpredictable results and eventually replication failure, with possible target database corruption.

Subset pointer replication:

This topic explains how to enable subset pointer replication and describes how the processing works.

You enable replication of subset pointer information by including the SSPCMD option in the EXIT statement for the DBD or those segments that contain subset pointers. Ensure that all segments that contain subset pointers have the SSPCMD option coded on their EXIT statements or that this option is specified at the DBD level. When enabled, the data capture log records produced will contain subset pointer update information.

New forms of the data capture log record are produced if an application updates a subset pointer using get calls. The data capture log records can include R command code information for insert operations or when a subset pointer is updated using a get call.

If your source application updates subset pointers using path calls, IMS generates a data capture record. The record reports information about inserts or updates and information about parent or intermediate level segments that had their subset pointers updated but were not inserted or updated. If these kinds of data capture log records contain R command code information, it is only used in the first insert or update call issued at the target. Each insert, update, or subset pointer update operation is processed as a separate DL/I call at the target even though the source application might have issued a single DL/I call to achieve the same results.

Insert HERE positioning:

This topic explains how replication processing works for segments with an insert rule of HERE.

When a source database is augmented to include INPOS information for database segments with an insert rule of HERE, the data capture records produced can include an additional segment image. That segment image identifies the twin before which a new segment occurrence was inserted. The target server performs the following processing:

- Generates a set of qualified SSAs and a final SSA using the O command code that contains the positioning segment image data
- Issues a get unique call
- If successful, inserts the new segment image at the target.

Provided that the positioning image identifies a unique instance for the source database record, the new segment instance is inserted in the proper location at the target. If not, the new segment is inserted before the first instance of the positioning segment which might not match the same position as the source database.

This same kind of rule also applies for update and delete operations referencing non-uniquely keyed segments, using the O command code approach. If the

segment data is unique (for a path), the correct instance is updated or deleted at the target. Otherwise the first instance that matches the duplicate data is affected.

With the IMS Version 13 maintenance applied at the source, if the source application uses a first or last command code to insert a segment this information is included in the data capture log record. The information is also included in an unqualified SSA that is used to insert the segment in the same location in the twin chain at the target.

Parallel apply:

Parallel apply processing improves the performance of your replication environment by applying units of recovery (UORs) concurrently to target databases.

Minimize latency

Parallel apply enables you to minimize latency by writing all available work across your subscriptions in near-real time. Within a single subscription, dependencies on UORs that update the same database record are a limiting factor. In this case, Data Replication for IMS places a higher priority on maintaining the precise order of changes as they occurred at the source database.

For UORs that change separate records or databases, delays or outages can prevent Data Replication for IMS from applying changes in strict order.

An administrator can disable parallel apply manually by modifying the subscription to perform serial processing.

Restrictions

Source UORs that require serialization

You might not get the full performance benefits of parallel apply if a UOR changes segments that do not have a unique concatenated key or updates subset pointers. The source server analyzes the UOR to determine whether changed segments have no keys, sequence fields are missing, or operations that affect subset pointers are detected. If so, the UOR is identified as requiring serial processing.

If you do not have completed UOR tracking enabled in your target server and a UOR requiring serial processing is encountered, subscription processing switches between parallel processing and serial processing as required. When completed UOR tracking is enabled, these kinds of UORs are allowed to be processed in parallel. During commit processing, the bookmark database is updated to record the fact that one of these types of UORs were applied at the target.

The topic “Creating a bookmark database” on page 81 describes the effects and system requirements for activating completed UOR tracking.

Databases that participate in logical relationships

You cannot define a subscription for parallel apply if a database participates in logical relationships between different physical databases.

Tip: You can replicate logically-related databases in a *serial* subscription if you include all the related physical databases in the subscription. The

replication mappings for logically-related databases must also share the same state. If you park or enable replication for one of the mappings, you must park or enable all of them.

Dependency analysis:

The target server supports parallel apply processing by evaluating the changes in a unit of recovery (UOR) for dependencies on changes that other UORs make.

By default, Data Replication for IMS balances transactional consistency with low latency and places a higher priority on applying changes in order to the same record than it does on maintaining precise order for different databases or records. In some situations this can cause an excessive number of deadlock situations and you can change the way dependency analysis is performed to eliminate deadlock situations.

Changes to the same database record

Data Replication for IMS applies changes to the same database record in order, regardless of how many IMS subsystems are updating the database or participating in ordering decisions. For each subscription that participates in parallel apply, the target server maintains a list of UORs in a queue, identifies dependencies, and waits for long UORs to finish.

With dependency analysis in effect, the target server tracks all data sets and the keys within those data sets for each UOR. If two UORs update records with the same key in the same data set, the earlier UOR must be applied first, and the later UOR becomes dependent upon the completion of that UOR. If they do not update records with the same key in the same data set, they are not dependent on each other and can be applied in parallel.

Data Replication for IMS also performs dependency analysis for serial subscriptions, but the memory requirements are smaller.

Dependency analysis assumes that no dependencies exist between different subscriptions.

Changes to different databases or records

Data Replication for IMS can also apply large numbers of UORs that update different databases and records in approximately the same order that they occurred at the source site. The number of records and databases that a UOR updates determines the extent to which the target server checks for dependencies.

To optimize memory and performance, the Classic data server determines whether to take the more granular approach of tracking root keys or merely to track the affected databases. If the number of updated records or databases exceeds a predefined threshold, the UOR becomes dependent on all prior UORs and all subsequent UORs are dependent on it.

Preventing deadlocks

Updates to the same database key are applied in the order they were made at the source. But updates that affect different keys in the same database can be applied in different orders than they occurred at the source.

In theory the approach of applying changes in key order would not cause any conflicts at the target because everything that is being applied in parallel updates a different database or minimally a different key in the same database. Unfortunately if you have two different UORs (being applied in parallel) that update different keys that randomize to the same block, IMS is going to detect a possible deadlock situation and will roll back one of the apply UORs. The target server is simply going to reschedule the failing UOR and try again. If that fails, the target server will wait a while and try again, and if that fails mark the UOR as requiring serial processing which fixes the problem.

In cases where a source UOR updates multiple database records, it is possible to encounter cross-database deadlocks between one or more UORs. These updates will automatically be re-tried and eventually all updates will be applied. If a lot of this kind of processing occurs, the apply rate degrades.

By default dependency analysis uses a two-component analysis approach. The first component identifies the resource that changed. The second component identifies the key of the database object that changed. For IMS, the key is identified as the root segment sequence field which is what IMS uses for locking purposes. Data Replication for IMS uses the following names as resource identifiers:

- The area name that was updated for a DEDB database
- The partition name that was updated for a HAL-DB database
- The name of the DBD for other database types

You can use the apply service DEPGRAPHKEYS configuration parameter to identify the resources to use for dependency analysis. Setting DEPGRAPHKEYS to 1 indicates to Data Replication for IMS that you only want dependency analysis to perform parallelism based on the names of the resources that have been updated. Using this option prevents deadlocks from occurring because updates to the same resource are applied in the order they occurred at the source site.

In situations where you have DEDB databases with a large number of areas and your typical workloads are restricted to updating a single area, using this "area-level" dependency analysis approach might provide better throughput than using the default full-key analysis that occurs when DEPGRAPHKEYS is set to 2 (the default).

Configuration parameters for dependency analysis:

A set of apply service configuration parameters control how Data Replication for IMS performs dependency analysis.

Table 18. Configuration parameters that control dependency analysis.

Parameters	Recommended value	Description
DEPGRAPHKEYS	2	Identifiers how many components to use: 1 – Resource names only 2 – Resource names and keys
DEPGRAPHUORLIMIT	200	Identifies the level of detail and analysis tracked for a UOR: • If a UOR contains more than DEPGRAPHUORLIMIT changes, only resource name analysis is performed. • If a UOR contains changes to DEPGRAPHUORLIMIT resource names, that UOR requires serial processing.

Table 18. Configuration parameters that control dependency analysis. (continued)

Parameters	Recommended value	Description
DEPGRAPHMEMORY	1000MB	Identifies the amount of 64-bit memory that is used for dependency analysis purposes. To prevent replication failures when a large number of UORs are waiting to be applied, dependency analysis will start removing detail – first by eliminating key information and then by eliminating resource names if necessary.
DEPGRAPHHASHSZ	64KB	Identifies the hash table size used for each component. If two values hash to the same slot they will be processed in the order these changes were performed at the source even though they update different objects at the target.

Adaptive apply:

Adaptive apply processing manages data conflicts that can arise while writing changes to target databases.

Adaptive apply processing is the default apply type for a replication mapping. If you choose to override the default and use standard apply, replication for the subscription stops when an error is encountered.

Example: A record insertion is replicated and a database record with a matching key is found that already exists in the target database

- With standard apply processing, this situation results in an error. In this case, a message is written to the log and the replication process stops and returns an error condition.
- With adaptive apply processing, this situation does not result in an error. In this case, a message is written to the log, the insert operation for the current change is skipped, and replication continues with the next change.

Adaptive apply is necessary in exceptional cases where your replication environment processes the same unit of recovery (UOR) more than once. Reprocessed UORs occur only during restart processing, when a parallel subscription is catching up to real-time processing after an immediate stop or system failure.

When you reprocess a UOR, apply processing can encounter data inconsistencies that generate errors:

- Missing records
- Missing parent segments
- Missing target segments
- Target segments that contain data that does not match before image data

Conflict handling

In most situations, adaptive apply processing discards changes that generate errors due to data inconsistencies:

- Insert (ISRT) processing encounters a duplicate segment, cannot locate the correct parent under which to insert, or cannot locate the correct target segment

- Delete (DLET) processing or update (REPL) processing cannot locate the correct parent or target segment
- Delete (DLET) processing or update (REPL) processing encounters a mismatch between the before image data and the data in the target segment

If adaptive apply processing encounters a status code for a condition other than one of those listed above, replication stops for the subscription.

Warning: When a conflict is detected while processing an insert, update, or delete operation the target database generally does not get updated when the inconsistency is detected. Partial target modifications might occur if the source application performed a path update operation and a conflict is detected for a child segment since apply processing updates each segment in a separate operation.

An exception to this general rule exists when you are replicating subset pointer updates, a segment with subset pointers has a sequence field, and an application uses a get call to modify or set the subset pointer. Because the target of the operation has a unique sequence field, the apply processing generates a qualified SSA to retrieve the segment and perform the subset pointer operations in a single call. When the call completes the data in the target segment image is compared against the source which can result in the detection of a conflict. But at that point the target subset pointer updates already occurred and since adaptive apply is enabled the inconsistency is ignored.

Exception: When replication is restarted for a subscription, changes can be applied by using adaptive apply processing regardless of the apply type setting for the replication mappings. Adaptive apply processing occurs for any UOR that existed during the parallel apply window that existed at restart.

The target server prevents target database corruption if completed UOR tracking is active and DONEUORS tracking information exists for a subscription. As UORs are received from the source server during the parallel apply window, the target server simply discards any UOR that exists on the DONEUORS list since it has already been applied at the target.

Conflict reporting

For standard apply, the first conflict encountered is reported as an error. A summary error message is written to the event log. Hexadecimal dumps of the appropriate records and record images involved in the conflict are written to the diagnostic log.

For adaptive apply, conflicts are reported as informational messages and written to the event log based on the value of the CONFLICTRPTLVL configuration parameter.

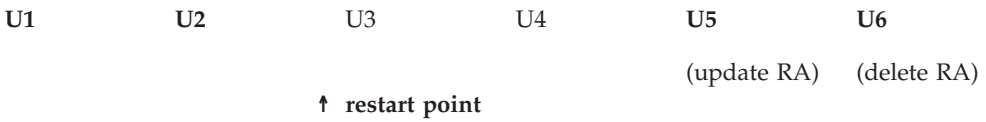
When a subscription is restarted after an immediate stop or system failure, adaptive apply is temporarily in effect for all replication mappings within a subscription until all UORs in an unknown state are applied, regardless of the apply type setting for a replication mapping.

Parallel apply and restart processing

To preserve data integrity and maintain the match between source and target databases, Data Replication for IMS applies changes to a given record in the correct

order during normal operations. However, an immediate stop or a system failure can result in unprocessed changes on the source server that fail to replicate, which can lead to processing the same changes twice.

For example, say that your replication environment is processing six sequential UORs (U1 to U6) when replication stops:



In this example, the UORs in bold (U1, U2, U5, and U6) have dependencies because they update the same record. In addition, U5 updates Record A and U6 *deletes* Record A. U3 is a long UOR, and for this reason Data Replication for IMS applies the UORs in bold but fails to apply U3 and U4 before replication stops.

Because the apply server sets a new restart time after applying the last contiguous change (U2), processing begins with U3 when the subscription restarts. Consistent with restart processing behavior, adaptive apply is in effect until the target server writes U6, which is the most recent change that completed before replication stopped.

When the target server tries to reprocess U5, the record is missing because U6 already deleted it. Adaptive apply handles the conflict by discarding the update. When the target server reprocesses U6, it also discards the delete.

In summary, adaptive apply processing discards conflicting changes on the assumption that they are a result of previous Data Replication for IMS processing. Adaptive apply discards other types of changes for similar reasons:

- Changes to missing segments
- Changes where the before image data does not match the data in the target segment

Conflict resolution in high availability environments:

If you replicate IMS data in both directions as part of a HADR solution, adaptive apply processing cannot handle all the data inconsistencies that can arise.

Failover and restart processing

In a HADR solution, you deploy a source server at the site that you are updating actively (in this example, site 1) and a target server at the failover site (site 2). You also set up a second deployment that consists of a source server at site 2 and a target server at site 1. Because you start replication in both directions, bookmarks advance at both sites during normal operations, even if you route all transactions to site 1. If a planned or unplanned outage occurs at site 1, you redirect transactions to site 2 during the outage.

When you restore service to site 1, these replication operations typically occur:

1. Restart processing replicates any committed UORs at site 1 that failed to replicate prior to the outage.
2. Failover processing replicates changes made at site 2 during the outage back to site 1.

These replication operations in both directions typically lead to mismatches between source and target databases that adaptive apply cannot resolve. You can use adaptive apply messages to resolve which UOR should have been replicated and make the appropriate corrections to the data sets. Ensure that the CONFLICTRPTLVL configuration parameter is set to generate messages for review.

Example

This example demonstrates how data inconsistencies can arise in a HADR environment. Consider the following scenario:

1. Prior to the failover, a unit of recovery (U1) at site 1 sets the BALANCE field in record A from 50 to 100.

   This is a committed UOR, but the failover occurs before the source server can replicate U1 to the target site.
2. After the failover, another unit of recovery (U2) at site 2 sets the BALANCE field in record A from 50 to 200.
3. After site 1 is back online, the source server at site 1 sends U1 to site 2.

   The balance information at site 2 (BALANCE=200) does not match the before image data in U1 (BALANCE=50), so adaptive apply processing discards U1. BALANCE=100 at site 1 and BALANCE=200 at site 2.
4. Site 2 replicates U2 to Site 1 when failover processing sends back the changes that occurred at the failover site during the outage.

   The balance information at site 1 (BALANCE=100) does not match the before image data in U2 (BALANCE=50), so adaptive apply processing discards U2. As before, BALANCE=100 at site 1 and BALANCE=200 at site 2.

Recursion protection

If you set up matching subscriptions that capture and replicate changes in both directions during normal operations, you must protect your environment from *recursion*, which occurs when you recapture changes written by a target server. Without recursion protection, your environment will replicate the same changes repeatedly in a continuous loop.

In a deployment like this, the subscriptions at the source and target server must be identical. They must have the same number of replication mappings, the replication mappings must be in the same states, and the names of the apply PSBs must match.

Data Replication for IMS protects you from recursion by automatically discarding any UORs where the name of the apply PSB for the subscription matches the name of the PSB in the first change message for that UOR. When you generate apply PSBs in the subscription wizards, ensure that the names match.

Designing subscriptions (guidelines)

Typically, you design subscriptions by adding replication mappings that represent a specific business application, such as Customers or Inventory. Use these guidelines to help you determine the best composition for your subscriptions.

About this task

In addition to maintaining application boundaries between subscriptions, workload management and parallel apply are also important considerations that help you to optimize your subscription design.

If you have a large number of independently updated databases, it can be difficult to identify the minimum number of subscriptions that you require and optimize the assignment of databases to your subscriptions. Use this procedure to guide your decisions.

Procedure

1. Work with your administrator to determine the number and composition of your subscriptions, with input from database administrators and application groups. In complex environments, subscription design can be an iterative process that is more art than science, requiring you to try different subscription configurations and adjust them based on test results.
2. Determine whether any of these situations apply to your replication goals and follow the applicable guidelines:
   a. Design subscriptions along application boundaries, so that a replication failure affects only one application. Wherever possible, define subscriptions to contain databases that service a single application, such as Payroll or Orders. If multiple applications update your databases, define multiple subscriptions that isolate replication failures to a single application.

      Dependency analysis does not consider dependencies across subscriptions, so designing your subscriptions along application boundaries is the best way to minimize the number of UORs that change databases that are in multiple subscriptions. See the topic "Dependency analysis."

      Dependencies across subscriptions can result in dividing a unit of recovery across multiple subscriptions that apply the UOR independently (a *split UOR*). In cases like this, the target server does not apply the changes to these databases in the correct sequence.
   b. Test replication workloads with one subscription, but use multiple subscriptions to avoid systemic failures.

      Avoid creating one large subscription that contains all the databases. This approach can be a good way to test how your environment responds to the workload, but it has the following disadvantages:

      - Replication failures affect replication for other databases related to other applications.
      - A single failure brings down all replication processing in the environment.
   c. Improve performance with parallel apply wherever appropriate.

      Some databases qualify for parallel apply and others do not, which might inform a decision to define some subscriptions for parallel apply and others for serial processing. For example, you cannot perform parallel apply with databases that participate in logical relationships between different physical databases. See the topic "Parallel apply."

      If you want to achieve the performance enhancements that parallel apply can provide, consider deploying parallel subscriptions for databases that qualify for concurrent processing and serial subscriptions for databases that do not.
3. Avoid situations where a single application updates a database that is in more than one subscription. Define multiple subscriptions for each database in situations like these:
   - You have a high-volume environment that replicates a small number of databases.
   - The partitioning method does not have a unifying attribute, such as the state where the customers reside.

4. Before deploying the subscriptions in a production environment, document your subscription design and monitor their performance in a test environment.

To create a subscription (IMS to IMS replication)

Use Classic Data Architect to create a subscription and add replication mappings for the subscription. (See Starting IMS replication.)

Procedure

1. Ensure you are in the **Replication** perspective (if not, select **Window** > **Open Perspective** > **Replication**).
2. Right-click any area in the Subscriptions view and select **Create Subscription**.
3. Provide identification, source and target server information for the subscription on the New Subscription dialog. For IMS Replication you can also specify the apply IMS PSB (program specification block) name
4. Click **Advanced Settings** to specify optional settings for the subscription (including the source ID, setting the subscription to automatically restart upon connection, capture and apply cache sizes, and apply attributes), and click **OK**.
5. Click **Yes** when prompted to add replication mappings for the subscription.
6. Add replication mappings to the subscription using the Add Replication Mappings wizard. Select the apply type to apply data to the target, and specify a log position from the Select Apply Type page, and click **Next**.
7. Search for source database descriptions (DBDs) to replicate using the Add Replication Mappings page. Type search criteria in the **Enter search** field to display a list of DBDs in the **Matched DBDs** list, Select the DBDs you want from this list and click the arrow icon to move it to the **Selected DBDs** list.

   Tip: For a fuzzy search, use the asterisk (*) as a wildcard character at the end of the search string.

   Numbers are displayed in brackets after the DBD name for logically related groups of DBDs. You must add or remove these groups as a unit. If you add a single member of a logically related group, you add the remaining DBDs in the group along with it. You also add any logically related DBDs that do not match the search query, and therefore are not displayed in the search results.

   The following types of DBDs are not in the search results, even if they match the search string:

   - DBDs that you already added to this subscription
   - DBDs that other replication mappings reference in another subscription, if that subscription uses the same source and target servers

   Restriction: You cannot start replication for the subscription until you augment the source DBD, the DBD exists at the target, and the source and target structures match.
8. Select a DBD from the list and click **View DBD Segments** to review the segments for the selected DBD.
9. Click **Close** on the View DBD Segments dialog to return to the Add Replication Mappings wizard.
10. Click **Next**.
11. Validate your replication mappings by clicking **Validate all** on the Validate Replication Mappings page. This validates all DBDs.

    Alternatively, you can select specific DBDs from the list (by clicking and holding down the **Ctrl** key) to validate, and clicking **Validate Selected DBDs**.

When you validate your DBDs, if there is a red X icon in the **Exists On Target** or **Matching Source/Target DBDs** columns, this indicates that you have not yet installed the target database. If this is the case, install the source DBDs at the target site. (See Loading or installing target IMS databases.)

12. Compare source and target DBD information on the Compare Source/Target DBDs dialog.
13. Click **Close** on the Compare Source/Target DBDs dialog.
14. Click **Next**.
15. Optional: Generate a PSB by selecting **Generate PSB** and providing a PSB location.

    The PSB generation options are disabled until validation has been performed on all items in the table and every DBD has been validated as existing on the target system. The PSB is generated when the wizard is complete.
16. Review the summary of replication mapping details. This includes replication mapping details.
17. Click **Finish** to exit the wizard. You are now ready to start replication.

Chapter 4. Administering Data Replication for IMS

Administering your replication environment involves ongoing tasks such as starting and stopping replication for subscriptions, loading or installing target IMS databases, and monitoring statistics, throughput, and latency.

Starting a data server

When you start a data server, you start those services that are configured to start in the configuration file for the server.

About this task

You can start Classic data servers using either of the methods listed.

Note: You can run a source server despite whether IMS is running on the source LPAR. If IMS stops on the target LPAR, apply processing detects it and replication stops for all subscriptions.

Procedure

Start a data server by using one of the following methods:

- Issue a console command to start the JCL procedure for the data server.

  `S procname`

  procname
  The 1-8 character PROCLIB member name to be started. When you issue commands from the SDSF product, prefix all operator commands with the forward slash (/) character.
- Submit the JCL as a batch job.

Stopping a data server

Stopping a data server stops all of the services that are running within it.

Procedure

Stop a data server by using one of the following methods:

- Issue the following command in an MTO interface:

  `F name,STOP,ALL`

  name Indicates the name of the started task or batch job of the data server.

  This form of the stop command requests a controlled shutdown of the data server. If issued against the source server, the controlled shutdown waits for all connected users to disconnect, ends replication for all actively replicating subscriptions, does an orderly shutdown of all services, and stops the source server. For persistent subscriptions, replication automatically restarts when the source server is restarted.
- Issue the z/OS **STOP** command:

  `STOP procname`

procname
Indicates the 1-8 character PROCLIB member name that represents the server that you want to stop.

Administering subscriptions

With Classic Data Architect, you can start and stop replication and work with existing subscriptions.

Starting replication

Use the Classic Data Architect to start replication.

Before you begin

- Create a subscription
- Activate the subscription's replication mappings

About this task

Tips:

- For best performance, set up your subscriptions as persistent so that they begin replicating automatically when you start the Classic data server.
- If you choose to start multiple subscriptions simultaneously, it is likely that they will be assigned to the same change stream, thereby optimizing resource consumption. (See Change streams.)

Procedure

From the **Subscriptions** view, right-click the subscription that contains the DBDs that you want to replicate and select **Start Replication**.
A message is displayed to indicate a successful start.

Stopping replication

Use the Classic Data Architect to stop replication.

Before you begin

- A subscription must be currently replicating.

Procedure

From the Subscriptions view, right-click the actively replicating subscription.

- Select **Stop Replication (Controlled)** if you want all transactions that are in process to finish before replication stops.
- Select **Stop Replication (Immediate)** if you want to stop all transactions immediately, rolling back any that are unfinished.

Controlled or immediate stop of replication

Classic data servers handle a request to stop replication differently, depending on whether you request a controlled stop or an immediate stop.

Stop Replication (Controlled)

If the source server receives a request for a controlled stop of replication, it stops sending change data for that subscription to the target server. When the target server receives the request to stop replication, it issues the

message CECA0002I. The target server continues to process all the transactions that it received before this point.

After processing all transactions, the target server unschedules any associated apply PSBs (program specification blocks). Then the target server issues message CECA0004I and sends a message to the source server indicating that replication stopped.

You can use a controlled stop request to help ensure that all changes will be reflected in a subscription's target data sets. You must first stop changes to the corresponding source data sets, then allow all changes in the replication log to be received at the target server, and then request a controlled stop.

Stop Replication (Immediate)

If the source server receives a request for an immediate stop of replication, it stops sending change data for that subscription to the target server. When the target server receives the request to stop replication, it issues the message CECA0003I. Unlike the controlled stop, the target server does not process any outstanding transactions.

A ROLB request backs out any units of recovery (UORs) that the subscription is processing and discards any unprocessed UORs. As with the controlled stop, the target server unschedules any associated apply PSBs. Then the target server issues message CECA0004I and sends a message to the source server indicating that replication stopped.

To modify an existing subscription

You can change an existing subscription using Classic Data Architect. For example, you can change the description, apply attributes, apply or capture cache size, and persistency for a subscription.

Procedure

1. Ensure you are in the **Replication** perspective (if not, select **Window** > **Open Perspective** > **Replication**).
2. Stop replication for the subscription you want to modify. Only inactive subscriptions can be modified. To stop replication, from the Subscriptions view, right-click the actively replicating subscription and select one of the following actions:

Option	Description
Stop Replication (Controlled)	Select this option if you want all transactions that are in process to finish before replication stops.
Stop Replication (Immediate)	Select this option if you want to stop all transactions immediately, rolling back any that are unfinished.

3. Right-click the subscription you want to modify from the Subscriptions view and select **Modify Subscription**.
4. Modify any of the attributes on the Subscription Properties dialog. Fields that are read-only and cannot be changed. To change any of the disabled fields, you must create a new subscription instead.

   Attention: Replication can only be started for a subscription that has at least one active replication mapping.

5. Click **Advanced Settings** to set the following attributes:
   - **Mark subscriptions as persistent (auto-restart)** If enabled, replication will be automatically restarted for this subscription once the server is restarted. This can occur after a normal shutdown and restart, or after a failure that does not require operator interaction to restart.
   - **Capture cache size** A value from 64 to 2048 MB.
   - **Apply cache size** A value from 64 to 2048 MB.
   - **Maximum parallel apply PSBs** A value from 1 to 255 representing the maximum number of program specification blocks (PSBs) that can be applied in parallel.

   Click **OK** to close the Advanced Subscriptions Settings dialog to save your changes and return to the Subscriptions Properties dialog.
6. Click **OK** to save your changes and close the Subscription Properties dialog.
7. Modify the replication mappings associated with this subscription on the Replications Mappings view. Click on the subscription in the Subscriptions view, and the details of the subscription will be shown in the Replication Mappings view. If the Replications Mapping view is not already in the list of displayed views then select **Window** > **Show View** > **Replication Mappings**.
8. Select the replication mapping to modify by right-clicking on it in the Replication Mappings view.
9. Use the **Change Apply Type** option to switch between the following options:
   - **Standard** - the standard apply type maintains an updated copy of the source database content in the target database
   - **Adaptive** - the adaptive apply type manages data conflicts that can arise between the source and target databases
10. Change the state of the replication mapping with either of the following options:
    - **Parked** - will not replicate when replication is started for the subscription
    - **Active** - include this mapping in the list of active mappings that will replicate when replication is started for the subscription. The Activate Replication Mappings window is displayed where you can either use the system's default log position (the current time) or specify a specific log position from which to start replication. Click **OK** to save the log position setting.

Results

The subscription and replication mappings are updated.

Administering replication mappings

You can manage the replication mappings in your subscriptions by changing their state to Active or Parked and setting a restart time for replication.

Activating replication mappings

For any subscription, you can set which mappings you want to be active for replication. Only the active mappings will replicate when replication is started for the subscription.

Before you begin

A subscription with replication mappings must be created and replication for the subscription must be stopped.

Procedure

1. Select a subscription and show its replication mappings.
2. In the Replication Mappings view, select the mappings that you want to activate.
3. Right-click the selected mappings and select **Activate**.
4. In the Activate Replication Mappings window, you can either use the system's default log position (the current time) or specify a specific log position from which to start replication.

   Attention: To be able to start replication for the subscription, at least one mapping must be active.

Results

- The table in the Replication Mappings view shows the state of the mappings, active or parked.
- If a DBD belongs to a logical group, then the activate action will be applied to all DBDs in the logical group.

Parking replication mappings

For any subscription, you can park replication mappings. Parked mappings will not replicate when replication is started for the subscription.

Before you begin

A subscription with replication mappings must be created and replication for the subscription must be stopped.

Procedure

1. Select a subscription and show its replication mappings.
2. In the Replication Mappings view, select the mappings that you want to park.
3. Right-click the selected mappings and select **Park (Do not replicate)**.
4.

Results

- The table in the Replication Mappings view shows the state of the mappings, active or parked.
- If a DBD belongs to a logical group, then the park action will be applied to all DBDs in the logical group.

Loading or installing target IMS databases

In some cases, you must synchronize your source and target IMS databases before starting or resuming replication.

About this task

Typically, Data Replication for IMS resumes automatically after an error or an outage. The target server sends bookmark information to the source server and the log reader service begins processing the IMS logs from the correct restart position.

Reinstall or reload a target database under the following conditions:

- Reinstall the target database structure after you reorganize a source database and make structural changes, such as adding or removing segments.

- Perform an initial load when you set up Data Replication for IMS for the first time.
- Reload the target database if inconsistencies, errors, or mass updates require it.

Follow these steps to reinstall or reload target IMS databases:

1. If you reorganized the source database, reinstall the database structure at the target site.
2. Create a nonconcurrent image copy of the database data sets for the source database.
3. Create and populate the data set replicas at the target site.

For more information about administrative tasks for IMS databases, see IMS Database Administration.

Installing database structures

In situations where the structure of a source IMS database is not yet present at the target site, install or reinstall the structure before you begin or resume replication.

About this task

Install or reinstall database structures in the following situations:

- You perform an initial load of a target database for the first time
- You add a new database to a subscription
- You reorganized a source IMS database and changed its structure

To begin replication with matching structure in the source and target databases, copy the source member for your database descriptions (DBDs) to the target site. If you replicate a source database to multiple targets, complete these steps for each subscription that includes that database.

The steps take place in these locations:

- The source IMS subsystem
- The target IMS subsystem

For more information about administrative tasks for IMS databases, see IMS Database Administration.

Procedure

1. **SOURCE SUBSYSTEM**: Take the source database offline by issuing the **/DBR** command.
2. **SOURCE SUBSYSTEM**: Perform DBDGEN.
3. **SOURCE SUBSYSTEM**: Point IMS update processing from the current library (ACBLIB) to an alternate library.
4. **SOURCE SUBSYSTEM**: Perform ACBGEN.
5. **SOURCE SUBSYSTEM**: Perform the online change process.
6. **SOURCE SUBSYSTEM**: Reorganize the database by using the unload and reload utilities of your choice.
   a. Unload the database to a temporary data set.
   b. Reload the database with the new structure.
7. **SOURCE SUBSYSTEM**: Stop replication for any subscriptions that replicate the source database.

8. **SOURCE SUBSYSTEM**: Take a nonconcurrent image copy of the source database.
9. **SOURCE SUBSYSTEM**: Start the source database by issuing the **/STA** command.
10. **TARGET SUBSYSTEM**: Take the target database offline by issuing the **/DBR** command.
11. **SOURCE AND TARGET SUBSYSTEM**: Send a copy of the DBD source member to the target site.
12. **TARGET SUBSYSTEM**: Run DBDGEN.
13. **TARGET SUBSYSTEM**: Point IMS update processing from the current library (ACBLIB) to an alternate library.
14. **TARGET SUBSYSTEM**: Perform ACBGEN.
15. **TARGET SUBSYSTEM**: Perform the online change process.
16. **TARGET SUBSYSTEM**: Follow the steps for reloading a target database with your data.
17. **TARGET SUBSYSTEM**: Restart replication for any subscriptions that replicate to the target database.

What to do next

You can now start or resume replication with a matching copy of your database.

Creating a nonconcurrent image copy of database data sets

IMS databases store their data in *database data sets*. Follow these steps to create a nonconcurrent image copy of the data sets for the source database.

Before you begin

You might have to install or reinstall the database structure first. Do this if you are performing an initial load of the target IMS database before starting replication for the first time, or if you reorganized your source IMS database and changed the segment structure.

About this task

Stop or quiesce the source database and wait for IMS to complete processing any unfinished transactions before you create your image copy. This approach ensures that Data Replication for IMS does not miss changes to the source database while you are making the copy.

If replication is active, use the point in time when you stopped or quiesced the source database to set the log position for the replication mappings that reference the database in all subscriptions. When you restart replication, the log reader service ignores all changes that occurred prior to the log position that you specify.

Tip: Do not set the log position to a time when the database was active, or you might miss changes.

If you are using standard IMS image copy utilities to create your image copies, the supplied sample job CECIGUSR generates job control language (JCL) that contains database recovery control (DBRC) commands that you will run later at the target site:

- The NOTIFY.IC command registers the image copy with DBRC.

- The GENJCL.USER command loads a standard IMS image copy to the database data sets.

For more information about administrative tasks for IMS databases, see IMS Database Administration.

Procedure

1. If you are using a standard IMS image copy and recovery utility, such as the Database Image Copy utility (DFSUDMP0) or the Database Image Copy 2 utility (DFSUDMT0), place the supplied sample member CECICNTY in the skeletal partitioned data set (PDS) for DBRC.

   The sample is in *USERHLQ*.USERSAMP(CECICNTY). The CECIGUSR job that you run in a later step references this member.
2. If you have not already done so, take the source database offline by issuing the **/DBR** command.

   Alternatively, you can quiesce the database.
3. Take a nonconcurrent image copy of each of the source database data sets by using a standard IMS image copy and recovery utility, a batch image copy utility, or a third-party utility of your choice.
4. Choose the method you will use to install any related primary or secondary index databases at the target IMS subsystem.

   Choose one of the following options:
   - Create image copies of the index databases and copy them to the target site when you transfer the database data sets.
   - Run an index builder utility at the target site after you load the target database. (This option is faster.)
5. If you decide to use image copies to install the index databases, create the image copy of the index databases at the same time that you create the image copy for the primary database.

   You can create an image copy of an index database any time after you stop or quiesce the primary database.
6. Customize the supplied sample JCL (CECIGUSR) with your own DBD and DD names, and then submit the job to extract the image copy information from the recovery (RECON) data sets.

   The job generates the JCL that you will run later at the target site. See the topic “Creating and populating data set replicas”.
7. Restart the database or end the quiesce.

What to do next

You can now transfer the image copies and the generated JCL to the target IMS subsystem and load the target database.

Creating and populating data set replicas

After you transfer the nonconcurrent image copies of the database data sets to the target IMS subsystem, register the data sets with the database recovery control facility (DBRC) and load the target databases.

Before you begin

You must first create an image copy of the database data sets for the source database. If you are using a standard IMS image copy and recovery utility, you

must also generate job control language (JCL) that contains the commands that register the data sets and load the target database.

About this task

You will use the image copies as a base from which to load the target databases. To register and load the database data sets, run the JCL that you generated after you created the image copies or issue the commands that are appropriate for type of image copy that you are using.

Tip: If you are using standard IMS image copy and recovery utilities, the image copy time in the **NOTIFY.IC** COMMAND must fall within a range of time when you have not yet allocated the database data set to the target IMS subsystem. If the UTC time of the source and apply systems differ significantly, the generated JCL might not work. For example, if you use a standard IMS image copy and recovery utility to generate the image copy, the UTC time that the utility embeds in the image copy might not fall within this range. In this case, you cannot use that image copy on the target site.

You must also install or rebuild index databases at the target site before you start replication for the primary database. You cannot replicate an IMS database that contains a primary or secondary index, or include this type of database in a replication mapping. Therefore, you cannot replicate any direct updates that applications make to an index database. In a case like this, the data in the target database does not match the source.

For more information about administrative tasks for IMS databases, see IMS Database Administration.

Procedure

1. Transfer the database data sets from the source site to the target site by using your preferred method.
2. Run the generated JCL or issue the appropriate command to build the database data sets.

Option	Description
If you are using a standard IMS image copy and recovery utility	Submit the JCL that you generated at the source site to run the required NOTIFY.IC and GENJCL.USER commands. The NOTIFY.IC command registers the database data sets with DBRC.
If you are using a nonstandard image copy utility (such as IDCAMS REPRO)	Issue the appropriate commands for the utility that you are using.

   If you are using image copies to install primary or secondary index databases, this step also installs those image copies.
3. If you are not using image copies to install index databases, run an index builder utility at the target site to rebuild the index databases.

   Do this after you install the primary database and before you start or restart replication.

What to do next

In all replication mappings that reference the database, set the log position to a point in time after you stopped or quiesced the source database and created image copies. You can now restart replication.

Releasing IMS RECON data sets for reorganization

When your IMS administrator reorganizes RECON data sets, use the RELEASE,RECON command to release any hold that the IMS log reader service has on the data sets.

About this task

Your IMS administrator reorganizes your IMS RECON data sets periodically as part of routine maintenance. As part of the process, the administrator might have to release a discarded data set from any IMS jobs that allocated it, including instances of the IMS log reader service. After release, the administrator can then delete and reallocate the data set.

Note: If you enable automatic RECON loss notification in your IMS system, you do not have to release the discarded data set.

Issue the **RELEASE,RECON** command to DBRC to release RECON data sets from the log reader service.

For information about reorganizing IMS RECON data sets, see "Maintaining the RECON Data Set" in the IMS documentation for the Database Recovery Control Facility (DBRC).

Procedure

1. Your IMS administrator issues the IMS command /RML to release the discarded RECON data set:

   ```
   /R nn,/RML DBRC='RECON STATUS'
   ```

   IMS releases the hold that any jobs have on the discarded RECON data set.
2. Release the discarded data set from the IMS log reader service by issuing the **RELEASE,RECON** command against the Classic data server that hosts the log reader service.

   ```
   /f Data-Server-Name,RELEASE,RECON
   ```

Results

After you issue the **RELEASE,RECON** command, the following console message appears:

```
CECZ0929I DBRC release RECON completed.
```

Monitoring a Data Replication for IMS environment

The monitoring service (MAA) provides metrics and subscription state and status change information.

About this task

The monitoring service must be configured on both the source and target servers. If the source server does not have a monitoring service, the capture service will not

initialize and an error will be issued to the console. If the monitoring service shuts down, the capture service will continue replicating but many errors will be written to the log when the capture service attempts to connect to the monitoring service to notify it of subscription state and status changes. If a monitoring service has not been configured, errors will also be logged on the target server.

When queried, the monitoring service checks both source and target subscription information. Should a server be running with both capture and apply, then both source and target information will be provided.

Monitoring subscription information by using MTO commands

The monitoring service issues a report that provides information about updates, transaction processing, and latency.

Before you begin

Add, configure, and start a monitoring service in the source server and target server.

About this task

You can issue MTO commands to report information about a subscription, about the subscription and its replication objects, or about a particular replication object. The report is displayed as a WTO message.

- Subscription name
- Messages, units of recovery, and bytes received
- Inserts, updates, and deletes processed
- Number of rollbacks
- Latency
- Thresholds

The latency statistics identify the time interval that elapsed between the completion of the last unit of recovery (UOR) at the source and the time the apply service wrote the change to the target database.

The threshold statistics identify apply latency threshold values for a subscription. The Classic data server uses these values to determine when to emit Event Integration Facility (EIF) events.

Procedure

- To obtain information about one or more subscriptions, issue the following command:

  ```
  DISPLAY,REPL,SUBSCR=subscription_name
  ```

  subscription_name
  Specify the name of the subscription as either a quoted or unquoted identifier. You can add an asterisk on the end as a wildcard. No other wild carding is supported.
- To obtain information about one or more subscriptions, including their replication mappings and replication object metrics, issue the following command:

  ```
  DISPLAY,REPL,SUBSCR=subscription_name,DETAIL
  ```

subscription_name

Specify the name of the subscription as either a quoted or unquoted identifier. You can add an asterisk on the end as a wildcard. No other wild carding is supported.

- To obtain status information for the subscriptions and replication mappings associated with one or more replication objects, issue the following command:

  `DISPLAY,REPL,MAPPING=`*`replication_object_name`*

  replication_object_name

  Specify the name of the replication object–the DBD name–as either a quoted or unquoted identifier. You can add an asterisk on the end as a wildcard. No other wild carding is supported.

- To obtain information about the threshold settings for one or more subscriptions, issue the following command:

  `DISPLAY,REPL,SUBSCR=`*`subscription_name`*`,THRESHOLDS`

  subscription_name

  Specify the name of the subscription as either a quoted or unquoted identifier. You can add an asterisk on the end as a wildcard. No other wild carding is supported.

Examples

Example 1: When you issue the **DISPLAY,REPL,SUBSCR** command with the **DETAIL** identifier, the output shows the current state and status for all subscriptions, including details about their replication mappings and objects:

```
DISPLAY,REPL,SUBSCR=*,DETAIL
```

```
CAC00200I DISPLAY,REPL,SUBSCR=*,DETAIL
CECM0053I SUBSCRIPTION METRICS DETAIL REPORT  255
SrcSysID Subscription Name
======== ============================================================
SUB0     SUB0
            Received         Sent         State: REPLICATE CONTINUOUS
           ============== ==============   Cache:   3%
 Bytes          11675551        11675551
 Rows              14830           14830
 Commits             141             141
 Inserts               0               0
 Updates           14830           14830
 Deletes               0               0

  Replication Object Name (Type)                              Status
     Inserts Sent     Updates Sent     Deletes Sent        Rows Sent
  ---------------  ---------------  ---------------  ---------------
  REPLOBJ0 (DBD)                                            ACTIVE
                0            14830                0            14830

SUB1     SUB1
            Received         Sent         State: REPLICATE CONTINUOUS
           ============== ==============   Cache:   1%
 Bytes           4802344         4802344
 Rows               6100            6100
 Commits              63              63
 Inserts               0               0
 Updates            6100            6100
 Deletes               0               0

  Replication Object Name (Type)                              Status
     Inserts Sent     Updates Sent     Deletes Sent        Rows Sent
  ---------------  ---------------  ---------------  ---------------
  REPLOBJ1 (DBD)                                            ACTIVE
```

```
               0            6100               0            6100

SUB2     SUB2
           Received         Sent         State: REPLICATE CONTINUOUS
         ============== ==============   Cache:  82%
 Bytes          35041817        30599243
 Rows              44530           38884
 Commits            4453            3888
 Inserts               0               0
 Updates           44530           38884
 Deletes               0               0

  Replication Object Name (Type)                              Status
     Inserts Sent     Updates Sent     Deletes Sent        Rows Sent
  ---------------  ---------------  ---------------  ---------------
  REPLOBJ2 (DBD)                                             ACTIVE
                0            38884               0            38884

Number of source subscriptions reported: 3
END OF REPORT
```

Example 2: When you issue the **DISPLAY,REPL,SUBSCR** command with the **THRESHOLDS** identifier, the output shows the threshold values for one or more subscriptions that contain threshold settings. You can use the report output to verify the threshold values that you specified by using the SET,REPL command.

The source server output will differ from the target server output. The report output for both source and the target servers shows the configured threshold values. The target server report also shows the number of events emitted for each threshold and the time associated with the most recently emitted event.

The following example shows output of the command DISPLAY,REPL,SUBSCR=SUB1,THRESHOLDS. In this example, the report displays the threshold values specified in the threshold set Thresholds1 for the subscription SUB1.

```
CECM0058I SUBSCRIPTION THRESHOLDS REPORT

SrcSysID Subscription Name
======== ============================================================
SUB1     SUB1
  Thresholds1              Latency State: Constrained
    Heartbeat        10000 (ms)              Meantime      10000 (ms)
    Maximum          60000 (ms)        5001  2011-09-21-10.23.31
    Reset            50000 (ms)        5000  2011-09-21-10.23.32
    Critical         40000 (ms)        6172  2011-09-21-10.23.33
    Constrained      30000 (ms)        7838  2011-09-21-10.23.34
    Discrete        100000 (ms)     8900123  2011-09-21-10.23.31

Number of target subscriptions reported: 1
```

The report output displays the following information.

SrcSysID
The source system ID for the subscription, SUB1.

Subscription Name
The name of the subscription, SUB1.

SetName
The name of the threshold set, Thresholds1. The report output for the threshold set displays:

- Threshold values that the user specified.

- A count of the number of events emitted for each threshold.

Latency State
The latency state of the target subscription associated with the GDPS/Active-Active workload. In this example, the latency state is Constrained.

Normal
A subscription initially enters the normal state when the subscription enters the Replicating Continuous state or when the Classic data server starts. After falling below the RESETTIME threshold, the subscription enters the latency state associated with the threshold currently exceeded.

Constrained
A subscription enters the Constrained state when the apply latency of the subscription exceeds the CONSTRAINEDTIME threshold.

Critical
A subscription enters the Critical state when the apply latency of the subscription exceeds the CRITICALTIME threshold.

Acute A subscription enters the Acute state when the apply latency of the subscription exceeds the MAXTIME threshold. The subscription remains in the Acute state until latency falls below the RESETTIME threshold. After falling below the RESETTIME threshold, the subscription enters the latency state associated with the threshold currently exceeded.

Time_of_last_event
The time of the last event emitted for the threshold of a target subscription.

Monitoring subscriptions with Classic Data Architect

You can monitor the replication metrics and latency of a subscription with Classic Data Architect, which provides numeric and visual data during the monitoring process.

Before you begin

For monitoring to take place, at least one subscription must be active or the capture cache must be currently in use.

About this task

You can set Classic Data Architect to monitor replication subscriptions at regular intervals, customizable in **Window** > **Preferences** > **Classic Data Architect** > **Replication Options** > **Subscription Metric Options**.

Classic Data Architect communicates with the monitoring service through the connection handler (the INIT service). Therefore, no additional communication strings or protocols need to be provided for the application to access the monitoring information.

Connections are authenticated through a System Authorization Facility (SAF) exit if you have configured one for the monitoring service using the SAFEXIT parameter.

Procedure

1. In the Subscriptions view, select the active subscription that you want to monitor.
2. Open each of the monitoring views if they are not already open. From the **Window** > **Show View** menu select each of: **Cache Meters, EIF Events, Latency,** and **Throughput**.
3. Right-click in the Latency or Throughput view and select **Start Metric Collection**.

Monitoring the event log

You can use the Event Log view to view all events in a single view, for a selected subscription or for all connected servers.

About this task

All source server, target server, and subscription events can be displayed in the view. Events can be sorted, filtered, and exported.

Procedure

1. Click on the subscription in the Subscriptions view, and the events related to the subscription will be shown in the Event Log view. If the Event Log view is not already in the list of displayed views then select **Window** > **Show View** > **Event Log**.
2. Keep the subscription highlighted if you want to see events for the subscription. If you want to see events for all connected servers, deselect the subscription before proceeding to the next step.
3. To obtain the most up-to-date events for the last hour, click the **Refresh current events** button. To retrieve events for another time period, click the **Retrieve Events** button and select one of the time frames, or select **Custom**.
   a. If you select **Custom**, the Retrieve Events dialog allows you to specify the date and time of the oldest event to retrieve and a maximum number of events to retrieve.
   b. The **Refresh current events** button is unavailable when custom event retrieval is specified. To use the refresh action again, you must reset the Retrieve Events setting to a predefined value (**Last Hour**, or **Last 6 Hours**, and so on). The latest events will then be retrieved, and subsequent refreshes are possible again.
4. The events in the event view are retrieved from the servers and stored in Classic Data Architect memory. If you are not connected to either a source or target server, the filtering options for the unavailable server are disabled and the messages for the disconnected server are not displayed. When viewing events for all connected servers, no subscription events are displayed.

Chapter 5. Troubleshooting and support

To isolate and resolve problems with your IBM software, you can use the troubleshooting and support information, which contains instructions for using the problem-determination resources that are provided with your IBM products.

Troubleshooting a problem

Troubleshooting is a systematic approach to solving a problem. The goal of troubleshooting is to determine why something does not work as expected and how to resolve the problem.

The first step in the troubleshooting process is to describe the problem completely. Problem descriptions help you and the IBM technical-support representative know where to start to find the cause of the problem. This step includes asking yourself basic questions:

- What are the symptoms of the problem?
- Where does the problem occur?
- When does the problem occur?
- Under which conditions does the problem occur?
- Can the problem be reproduced?

The answers to these questions typically lead to a good description of the problem, which can then lead you a problem resolution.

What are the symptoms of the problem?

When starting to describe a problem, the most obvious question is "What is the problem?" This question might seem straightforward; however, you can break it down into several more-focused questions that create a more descriptive picture of the problem. These questions can include:

- Who, or what, is reporting the problem?
- What are the error codes and messages?
- How does the system fail? For example, is it a loop, hang, crash, performance degradation, or incorrect result?

Where does the problem occur?

Determining where the problem originates is not always easy, but it is one of the most important steps in resolving a problem. Many layers of technology can exist between the reporting and failing components. Networks, disks, and drivers are only a few of the components to consider when you are investigating problems.

The following questions help you to focus on where the problem occurs to isolate the problem layer:

- Is the problem specific to one platform or operating system, or is it common across multiple platforms or operating systems?
- Is the current environment and configuration supported?

If one layer reports the problem, the problem does not necessarily originate in that layer. Part of identifying where a problem originates is understanding the environment in which it exists. Take some time to completely describe and document the problem environment, including the following items:

- Operating system
- Product version
- All corresponding software and versions, hardware information, and any maintenance that was applied

Confirm that you are running within an environment that is a supported configuration; many problems can be traced back to incompatible levels of software that are not intended to run together or have not been fully tested together.

When does the problem occur?

Develop a detailed timeline of events leading up to a failure, especially for those cases that are one-time occurrences. You can most easily develop a timeline by working backward: Start at the time an error was reported (as precisely as possible, even down to the millisecond), and work backward through the available logs and information. Typically, you need to look only as far as the first suspicious event that you find in a diagnostic log.

To develop a detailed timeline of events, answer these questions:

- Does the problem happen only at a certain time of day or night?
- How often does the problem happen?
- What sequence of events leads up to the time that the problem is reported?
- Does the problem happen after an environment change, such as upgrading or installing software or hardware?

Responding to these types of questions can give you a frame of reference in which to investigate the problem.

Under which conditions does the problem occur?

Knowing which systems and applications are running at the time that a problem occurs is an important part of troubleshooting. These questions about your environment can help you to identify the root cause of the problem:

- Does the problem always occur when the same task is being performed?
- Does a certain sequence of events need to occur for the problem to surface?
- Do any other applications fail at the same time?

Answering these types of questions can help you explain the environment in which the problem occurs and correlate any dependencies. Remember that just because multiple problems might have occurred around the same time, the problems are not necessarily related.

Can the problem be reproduced?

From a troubleshooting standpoint, the ideal problem is one that can be reproduced. Typically, when a problem can be reproduced you have a larger set of tools or procedures at your disposal to help you investigate. Consequently, problems that you can reproduce are often easier to debug and solve. However,

problems that you can reproduce can have a disadvantage: If the problem is of significant business impact, you do not want it to recur. If possible, re-create the problem in a test or development environment, which typically offers you more flexibility and control during your investigation.

- Can the problem be re-created on a test system?
- Are multiple users or applications encountering the same type of problem?
- Can the problem be re-created by running a single command, a set of commands, or a particular application?

Searching for messages

You can search for messages in the information center.

In the search box that is located in the **top-left toolbar** of this information center, enter the message number; for example, enter: 0x00670014.

Important: You need to enter the search string in the format of the full message number, in this example 0x00670014. Do not specify partial message numbers or wild cards (* or ?) in the search string.

Searching knowledge bases

You can often find solutions to problems by searching IBM knowledge bases. You can optimize your results by using available resources, support tools, and search methods.

About this task

You can find useful information by searching the information center, but sometimes you need to look beyond the information center to answer your questions or resolve problems.

Procedure

To search knowledge bases for information that you need, use one or more of the following approaches:

- Find the content that you need by using the IBM Support Portal.

  The IBM Support Portal is a unified, centralized view of all technical support tools and information for all IBM systems, software, and services. The IBM Support Portal lets you access the IBM electronic support portfolio from one place. You can tailor the pages to focus on the information and resources that you need for problem prevention and faster problem resolution.

  Familiarize yourself with the IBM Support Portal by viewing the demo videos about this tool. These videos introduce you to the IBM Support Portal, explore troubleshooting and other resources, and demonstrate how you can tailor the page by moving, adding, and deleting portlets.
- Search for content by using one of the following additional technical resources:
  - APARs (problem reports). You can locate APARs on the IBM Support site or by using an external search engine. To locate APARs on the IBM Support site:
    1. Select Information Management in the **Choose support type** box.
    2. Select a Classic product in the **Choose product** box.
    3. Enter an APAR number in the **Search** box.

Tip: To narrow the search for Classic products only, specify the component identifier "5697I8200" for Classic products.

- Information management forums. This page lists a variety of forums about specific IBM Information Management products.

- Search for content by using the IBM masthead search. You can use the IBM masthead search by typing your search string into the Search field at the top of any ibm.com® page.
- Search for content by using any external search engine. If you use an external search engine, your results are more likely to include information that is outside the ibm.com domain. However, sometimes you can find useful problem-solving information about IBM products in newsgroups, forums, and blogs that are not on ibm.com.

  Tip: Include "IBM" and the name of the product in your search if you are looking for information about an IBM product.

Getting fixes

A product fix might be available to resolve your problem.

About this task

Procedure

- To find and install fixes:
  1. Access downloads and fixes:
     - If you know a PTF number, go to Download specific fixes. You might know a PTF number from a technote or APAR description that you found by entering keywords for a search of the product support web site.
     - If you do not know a PTF number, go to the IBM Support Portal. From there, you can search for fixes for your product. If you have not visited the IBM Support Portal in the past, you can customize it so that you can view Support-related information for the specific products that you use.

       Alternatively, visit the Get zSeries related fixes web site.
  2. Follow the instructions at the eServer™ zSeries website to locate a fix that might solve your problem.
  3. When you find a fix that you are interested in, click the name of the fix to read its description. If you believe that the fix can resolve your problem, download the fix and apply it.
  4. Optional: Subscribe to receive weekly email notifications about fixes and other IBM Support information.
- To find a list of fixes for a product rollup, see the Release Notes for the rollup that you need.
- To find fixes for the Classic Data Architect, run the IBM Installation Manager. See Applying maintenance to the Classic Data Architect for instructions.

Contacting IBM Support

IBM Support provides assistance with product defects, answering FAQs, and performing rediscovery.

Before you begin

After trying to find your answer or solution by using other self-help options such as technotes, you can contact IBM Support. Before contacting IBM Support, your company must have an active IBM maintenance contract, and you must be authorized to submit problems to IBM. For information about the types of available support, see the Support portfolio topic in the *Software Support Handbook*.

Procedure

Complete the following steps to contact IBM Support with a problem:

1. Define the problem, gather background information, and determine the severity of the problem. For more information, see the Getting IBM support topic in the *Software Support Handbook*.
2. Gather diagnostic information.
3. Submit the problem to IBM Support in one of the following ways:
   - Online through the IBM Support Portal: You can open, update, and view all your Service Requests from the Service Request portlet on the Service Request page.
   - By phone: For the phone number to call in your country, see the Directory of worldwide contacts web page.

Results

If the problem that you submit is for a software defect or for missing or inaccurate documentation, IBM Support creates an Authorized Program Analysis Report (APAR). The APAR describes the problem in detail. Whenever possible, IBM Support provides a workaround that you can implement until the APAR is resolved and a fix is delivered. IBM publishes resolved APARs on the IBM Support website daily, so that other users who experience the same problem can benefit from the same resolution.

Exchanging information with IBM

To diagnose or identify a problem, you might need to provide IBM Support with data and information from your system. In other cases, IBM Support might provide you with tools or utilities to use for problem determination.

Collecting diagnostic information

You can use Classic Data Architect to collect diagnostic information from all connected servers. These diagnostics can be exported to a file for evaluation.

About this task

When you start collecting diagnostic information, all of the information is stored in Classic Data Architect cached memory. You can set the frequency at which diagnostic information is collected from the servers and the maximum number of results to be cached in memory. These options are available on the **Window** > **Preferences** > **Classic Data Architect** > **Diagnostic Metric Options** panel.

The information is also displayed in the Diagnostic Metric view. The metrics that have been collected from the selected server are displayed in this view, sorted by timestamp. Previous results (up to the maximum that you specify) and results for

non-selected servers are still available in cached memory and can be exported to a file.

Procedure

1. To display the Diagnostic Metric view, select **Window** > **Views** > **Diagnostic Metric view**.
2. To start collecting diagnostics, right-click in the Diagnostic Metric view and select **Start Diagnostic Metric Collection**.
3. Select a server to view the most up-to-date metrics for a particular server. When you select an object in the tree, the diagnostic information to the right is updated dynamically with the latest results collected.
4. To export diagnostic information to a file, right-click the Diagnostic Metric view and select **Export Diagnostic Metrics**. The metrics are exported as comma-separated values (CSV) to a file of your choosing. You can export the metrics currently displayed in the view or all metrics cached in memory. Metrics exported from the view are exported in the order that they are displayed. Metrics exported from memory are sorted by timestamp.
5. To clear all collected diagnostics, including those stored in memory, right-click the Diagnostic Metric view and select **Clear Diagnostic Metric History**.

Sending information to IBM Support

To reduce the time that it takes to resolve your problem, you can send trace and diagnostic information to IBM Support.

Procedure

To submit diagnostic information to IBM Support:

1. Open a problem management record (PMR).
2. Collect the diagnostic data that you might need, either manually or automatically, depending on the data. Diagnostic data helps reduce the time that is spent resolving your PMR. For example, having access to any relevant messages, error codes, log data, all data server output, trace output, or dump output, can speed the resolution process.
3. Compress the files by using one of the following methods, depending on the file type.
   - Use the AMATERSE program, which is a tool that is available for products that run in a z/OS environment. For more information about what z/OS versions support this program, search for either program name on ibm.com.
   - For UNIX files, use the tar and gzip programs to create compressed archive files.
     a. Run the tar program against the file.
     b. Run the gzip program against the file.

     For example: `tar -cvf - inputfile1 inputfile2 | gzip > file.tar.gz`
   - For Microsoft Windows files, create a ZIP file.
4. Transfer the files to IBM. You can use one of the following methods to transfer the files to IBM:
   - Service Request tool
   - Standard data upload methods: FTP, HTTP
   - Secure data upload methods: FTPS, SFTP, HTTPS
   - Email

If you are using a z/OS product and you use ServiceLink/IBMLink to submit PMRs, you can send diagnostic data to IBM Support in an email or by using FTP.

All of these data exchange methods are explained on the IBM Support site.

Receiving information from IBM Support

Occasionally an IBM technical-support representative might ask you to download diagnostic tools or other files. You can use FTP to download these files.

Before you begin

Ensure that your IBM technical-support representative provided you with the preferred server to use for downloading the files and the exact directory and file names to access.

Procedure

To download files from IBM Support:

1. Use FTP to connect to the site that your IBM technical-support representative provided and log in as `anonymous`. Use your email address as the password.
2. Change to the appropriate directory:
   a. Change to the `/fromibm` directory.
      ```
      cd fromibm
      ```
   b. Change to the directory that your IBM technical-support representative provided.
      ```
      cd nameofdirectory
      ```
3. Enable binary mode for your session.
   ```
   binary
   ```
4. Use the **get** command to download the file that your IBM technical-support representative specified.
   ```
   get filename.extension
   ```
5. End your FTP session.
   ```
   quit
   ```

Subscribing to Support updates

To stay informed of important information about the IBM products that you use, you can subscribe to updates.

About this task

By subscribing to receive updates, you can receive important technical information and updates for specific Support tools and resources. You can subscribe to updates by using one of two approaches:

RSS feeds and social media subscriptions

The following RSS feeds and social media subscriptions are available:

- RSS feeds for various Information Management communities. See the Information Integration section of Information Management community.
- RSS feed for developerWorks® resources, such as articles, tutorials, downloads, and forums. See developerWorks.

For general information about RSS, including steps for getting started and a list of RSS-enabled IBM web pages, visit the IBM Software Support RSS feeds site.

My Notifications

With My Notifications, you can subscribe to Support updates for any IBM product. You can specify that you want to receive daily or weekly email announcements. You can specify what type of information you want to receive (such as publications, hints and tips, product flashes (also known as alerts), downloads, and drivers). My Notifications enables you to customize and categorize the products about which you want to be informed and the delivery methods that best suit your needs.

Procedure

To subscribe to Support updates:

1. To subscribe to My Notifications, begin by going to the IBM Support Portal and clicking **My Notifications** in the **Notifications** portlet.
2. If you have already registered for My support, sign in and skip to the next step. If you have not registered, click **Register now**. Complete the registration form using your email address as your IBM ID and click **Submit**.
3. Click **Edit profile**.
4. Click **Add products** and choose a product category; for example, **Software**. A second list is displayed.
5. In the second list, select a product segment; for example, **Data & Information Management**. A third list is displayed.
6. In the third list, select a product subsegment, for example, **Databases**. A list of applicable products is displayed.
7. Select the products for which you want to receive updates.
8. Click **Add products**.
9. After selecting all products that are of interest to you, click **Subscribe to email** on the **Edit profile** tab.
10. Select **Please send these documents by weekly email**.
11. Update your email address as needed.
12. In the **Documents list**, select the product category; for example, **Software**.
13. Select the types of documents for which you want to receive information.
14. Click **Update**.

Results

Until you modify your RSS feeds and My Notifications preferences, you receive notifications of updates that you have requested. You can modify your preferences when needed (for example, if you stop using one product and begin using another product).

Related information

- IBM Software Support RSS feeds
- Subscribe to My Notifications support content updates
- My notifications for IBM technical support
- My notifications for IBM technical support overview

Chapter 6. Reference

Look up information about services and their configuration parameters, commands, utilities, and messages.

Services and configuration parameters

Services run in the address space of a Classic data server. After you complete the installation customization process, adjust the configuration of these services by using the Classic Data Architect.

Summary of services

The services required for the Classic data server are customized and running when you complete the installation customization process.

The following table summarizes these services.

Table 19. Summary of services

Service type	Task name	Service class
Administration service	CECPAA	PAA
Apply service	CECAPLY	APLY
Capture service	CECCAP	CAP
Client connection handler service	CACINIT	INIT
IMS database resource adapter (DRA) access service	CACDRA	DRA
IMS log reader service	CECLRS	LRSI
Logger service	CACLOG	LOG
Monitoring service	CECMAA	MAA
Region controller service	CACCNTL	CNTL
Remote operator command service	CACOPER	OPER

Administration service

The administration service is assigned to the PAA service class. The task name for the PAA service class is CECPAA.

The administration service manages the configuration and administration of replication. The management functions include setting up replication, error reporting during setup, and access to the system event log. The administration service also maintains the replication runtime environment.

The administration service runs in the address space of the source server and the target server. The administration service interfaces with the capture service. At run time, the administration service accesses the definitions of subscription, replication objects, and replication mappings and supplies those definitions to the capture service.

The administration service interfaces with the Classic Data Architect (CDA). CDA directs requests to the administration service to process subscriptions and replication mapping definitions and to start and stop replication activities for a subscription.

The following table lists the configuration parameters that apply to the PAA service class.

Table 20. Configuration parameters for the PAA service class

Parameter	Default value	Description
IDLETIMEOUT	5M	The amount of time that a service remains idle before it polls the local message queue for messages that are to be processed
INITIALTHREADS	1	Number of instances of this service that the region controller starts during initialization of the Classic data server
MAXTHREADS	1	Maximum number of instances of this service that the region controller is allowed to start
MAXUSERS	100	Maximum number of user connections
RESPONSETIMEOUT	3S	Maximum amount of time to wait for response before terminating a connection
SAFEXIT	None	The System Authorization Facility (SAF) system exit that performs authorization checks for connections to the Classic data server.
SEQUENCE	0	Sequence number assigned to services
TRACELEVEL	4	Trace level

Apply service

The apply service is assigned to the APLY service class. The task name for the APLY service class is CECAPLY.

Apply services receive messages from one or more capture services. The messages can contain replication control information and change data. The apply service makes the change data available for processing by one of the writer threads that applies the changes to the target data store.

Restriction: The Classic data server supports a single apply service. An attempt to add an additional apply service in the same address space will fail.

The following table lists the configuration parameters for the APLY service class.

Table 21. Configuration parameters for the APLY service class.

Parameter	Default value	Description
CICSBOOKMARKDB*	None	The name of the CICS® file that Data Replication for VSAM uses as a bookmark file.
CICSCONINTRVL*	15S	The frequency at which the apply service retries connecting to CICS when CICS is not available at target server startup.
CICSPORT*	0	The TCP port number that the z/OS CICS IP Socket Listener in the target CICS region monitors for connection requests from the apply service.

Table 21. Configuration parameters for the APLY service class. (continued)

Parameter	Default value	Description
CICSRECONINTRVL*	1M	The frequency at which the apply service retries establishing a CICS connection after losing the connection.
CICSTSQNAME*	CFCQ1	Defines the name of a control temporary storage queue that Data Replication for VSAM uses in the target CICS region.
CONFLICTRPTLVL	1	The informational messages issued during adaptive apply processing to report a conflict during a record insert, update, or delete operation.
DEPGRAPHHASHSZ	64 KB	Controls the size of the hash table that contains keys for each component used in dependency analysis.
DEPGRAPHKEYS	2	Identifies the number of components that are used in dependency analysis. For IMS, the first component identifies the area, database, or partition that changed. The second component identifies the root sequence field for the resource.
DEPGRAPHMEMORY	1000 MB	Controls the size of the 64-bit memory object that the target server obtains for dependency analysis.
DEPGRAPHUORLIMIT	200	Controls how the target server tracks UOR information for dependency analysis.
DONEUORLIMIT	100000	Identifies the maximum number of UORs that will be tracked for a subscription that uses completed UOR tracking.
DONEUORTRACKING	FALSE	Identifies whether to enable completed UOR tracking which allows a UOR that requires serial processing to be applied in parallel.
IMSBOOKMARKDB	None	The name of the IMS database that Data Replication for IMS uses as a bookmark database.
IDLETIMEOUT	300000ms (5M)	The amount of time that a service remains idle before it polls the local message queue for messages to process.
IMSV11CMPAT	FALSE	Identifies whether the version of IMS running at the target does not support use of the O command code. IMS version 11 does not contain this support. IMS version 12 and later contain this support.
INITIALTHREADS	1	The number of instances of this service that the region controller starts during initialization of the Classic data server.
LISTENURL	None	The protocol identifier and address for communication for the apply service.
MAXWRITERTHREADS	10	The maximum number of writer threads that the apply service can start.
RESPONSETIMEOUT	300000ms (5M)	The maximum amount of time to wait for a response before terminating a connection.
SEQUENCE	0	The sequence number assigned to services.

Table 21. Configuration parameters for the APLY service class. (continued)

Parameter	Default value	Description
STRICTVALIDATION	TRUE	Controls whether the source and target DBD attributes must be identical for replication to start for a subscription (TRUE) or replication is allowed to start if differences are detected.
TRACELEVEL	4	The trace level for the apply service.
USECICSMAPFIRST	FALSE	Specifies whether the apply service should use the CICS mapping file CECCFMAP to search for mappings between VSAM data sets and CICS files before searching the File Control Table (FCT).
USERSUBPOOLMAX	262144	The maximum size for the user pool.
WRITERTHREADS	3	The number of writer threads that are started when the apply service starts.

*The CICSBOOKMARKDB, CICSCONINTRVL, CICSPORT, CICSRECONINTRVL, CICSTSQNAME, and USECICSMAPFIRST configuration parameters do not apply to Data Replication for IMS.

Capture service

The capture service is assigned to the CAP service class. The task name for the CAP service class is CECCAP.

The capture service manages change data capture. Those operations include management of the log reader service, change streams, and communication with the target server.

The log reader service reports stream-related errors to the capture service that initiates end-of-replication for the affected subscriptions. The log reader service can also report other types of errors such as Classic data server or non-stream related errors that end replication for all active subscriptions and cause the capture service to stop.

The capture service also interfaces with the apply service to transmit change messages for processing.

Restriction: Only a single service of service class CAP can be configured in a single Classic data server. A single instance of this service runs in the address space of a Classic data server.

Connection handler service

The connection handler service is assigned to the INIT service class. The task name for the INIT service is CACINIT.

A connection handler listens for connection requests from client applications and routes the requests to the appropriate administration, monitoring, and operator tasks. The connection handler task can load the TCP/IP communication protocol.

Remote client applications use TCP/IP to communicate with a Classic data server.

The following table summarizes the configuration parameters that define connection handler services in the INIT service class.

Table 22. Configuration parameters for the INIT service class

Parameter	Default value	Description
COMMSTRING	TCP/0.0.0.0/ 9087	Client connection listen string
IDLETIMEOUT	5M	The amount of time that a service remains idle before it polls the local message queue for messages that are to be processed
INITIALTHREADS	1	Number of instances of this service that the region controller starts during initialization of the Classic data server
MAXTHREADS	1	Maximum number of instances of this service that the region controller is allowed to start
MAXUSERS	100	Maximum number of user connections
RESPONSETIMEOUT	5M	Maximum amount of time to wait for response before terminating a connection
SEQUENCE	0	Sequence number that is assigned to services.
TRACELEVEL	4	Trace level

Database resource adapter (DRA) service

The DRA service is assigned to the DRA service class. The task name for the DRA service class is CACDRA.

You need to enable the DRA interface to allow the target server to access IMS data for replication.

The DRA service interface initializes the DRA interface and connects to an IMS DBCTL region to access IMS data.

The following table lists the configuration parameters that define DRA services in the DRA service class.

Table 23. Configuration parameters for the DRA service class

Parameter	Default value	Description
CONNECTINTERVAL	15S	The interval time in seconds at which the data DRA service retries connecting to IMS when IMS is not available
DEFAULTPSBNAME	NOPSB	Default PSB name
DRATABLESUFFIX	None	DRA startup table suffix
DRAUSERID	None	Default DRA user ID that is used to connect to and register with a DBCTL subsystem. The DRAUSER is the name by which the Classic data server is known to DBCTL
IDLETIMEOUT	5M	The amount of time that a service remains idle before it polls the local message queue for messages that are to be processed
INITIALTHREADS	1	Number of instances of this service that the region controller starts during initialization of the Classic data server
MAXTHREADS	1	Maximum number of instances of this service that the region controller is allowed to start
MAXUSERS	10	Maximum number of user connections

Table 23. Configuration parameters for the DRA service class (continued)

Parameter	Default value	Description
RECONNECTWAIT	1M	The minimum amount of time in minutes that the DRA service waits after an IMS disconnect before attempting to reconnect to IMS
RESPONSETIMEOUT	5M	Maximum amount of time in minutes to wait for response before terminating a connection
SEQUENCE	0	Sequence number that is assigned to services
TRACELEVEL	4	Trace level

IMS log reader service

The IMS log reader service is assigned to the LRSI service class. The task name for the LRSI service class is CECLRS.

The IMS log reader service must communicate with the capture service. For communication between these services to occur successfully, the IMS log reader service and the capture service must be defined in the same Classic data server.

The IMS log reader service processes stream requests received from the capture service. Stream requests enable the IMS log reader service to return IMS change data from the necessary log streams.

Restriction: Only one instance of an IMS log reader service can be defined in a single Classic data server.

The following table lists the configuration parameters that apply to the LRSI service class.

Table 24. Configuration parameters for the LRSI service class.

Parameter	Default value	Description
INACTTHRESHOLD	30S	Interval time for issuing the inactivity message CECZ0400W for a given subsystem.
INITIALTHREADS	1	Number of instances of this service that the region controller starts during initialization of the Classic data server
NOTIFICATIONURL	None	TCP/IP address and port that the log reader interface uses for event notification. This parameter is defined as a 64-byte character string.
SEQUENCE	0	Sequence number assigned to services
SSIDEXCLUDELIST	None	Represents the IMS subsystem exclusion list for the log reader service.
TRACELEVEL	4	Trace level

Logger service

The logger service is assigned to the LOG service class. The task name for the LOG service class is CACLOG.

The logger service receives messages from all services in the data server and coordinates writing the messages to a common log. This service sends Event

Integration Facility (EIF) events to event servers in a GDPS/Active-Active environment. The logger also reports data server activities and is used for error diagnosis.

Restriction: A single logger task can run within a data server.

When the logger service is initialized, the configuration parameters in the LOG service class are defined with the parameter default values. You can modify the default values as needed.

The following table lists the configuration parameters that apply to the LOG service class.

Table 25. Configuration parameters for the LOG service class

Parameter	Default value	Description
CONSOLELEVEL	4	The amount of event messages that data server tasks record in the event log
DISPLAYLOG	FALSE	Display log
EIFEVENTSERVERS	None	The service list for URL values that identifies the event server to which Event Integration Facility (EIF) events will be sent
EVENTLOG	None	The name of the event message log that is defined to the logger service
IDLETIMEOUT	5M	The amount of time that a service remains idle before it polls the local message queue for messages that are to be processed
INITIALTHREADS	1	Number of instances of this service that the region controller starts during data server initialization
LOGBUFSIZE	65536	Log buffer size
LOGURL	None	Log URL that provides a method of moving logging storage outside of the address space MESSAGEPOOLSIZE storage and into a data space.
MAXTHREADS	1	Maximum number of instances of this service that the region controller is allowed to start
MAXUSERS	100	Maximum number of user connections
MSGLIST	None	Represents a message list that is maintained as a service list.
RESPONSETIMEOUT	5M	Maximum amount of time to wait for response before terminating a connection
SEQUENCE	0	Sequence number that is assigned to services
STREAMNAME	None	Stream name used for the diagnostic log
TRACELEVEL	1	Trace level

Monitoring service

The monitoring service is assigned to the MAA service class. The task name for the MAA service class is CECMAA.

The monitoring service runs in multiple address spaces and interfaces with both the source server and the target server. The purpose of this service is to manage all monitoring of the progress of replication for the source and target servers.

The monitoring service reports metrics about replication data and latency information. Other management functions include reporting, receiving display and report commands, and producing the reports needed from runtime information.

The following table lists the configuration parameters that apply to the MAA service class.

Table 26. Configuration parameters for the MAA service class.

Parameter	Default value	Description
IDLETIMEOUT	5M	The amount of time that a service remains idle before it polls the local message queue for messages that are to be processed
INITIALTHREADS	1	Number of instances of this service that the region controller starts during initialization of the Classic data server
MAXTHREADS	1	Maximum number of instances of this service that the region controller is allowed to start
MAXUSERS	100	Maximum number of user connections
NMICOMMSTRING	None	The communication path for the Network Management Interface (NMI) AF_UNIX domain socket.
RESPONSETIMEOUT	3S	Maximum amount of time to wait for response before terminating a connection
SAFEXIT	None	Name of the System Authorization Facility (SAF) system exit
SEQUENCE	0	Sequence number that is assigned to services
TRACELEVEL	4	Trace level

Operator service

The command operator service is assigned to the OPER service class. The task name for the OPER service class is CACOPER.

The operator service supports a command operator interface for distributed client applications.

The operator service also handles communications between the Classic data server and the configuration support in the Classic Data Architect. To use the configuration support in the Classic Data Architect, the operator service must be running on the Classic data server.

The following table summarizes the configuration parameters that define command operator services in the OPER service class.

Table 27. Configuration parameters for the OPER service class

Parameter	Default value	Description
IDLETIMEOUT	5M	The amount of time that a service remains idle before it polls the local message queue for messages that are to be processed

Table 27. Configuration parameters for the OPER service class (continued)

Parameter	Default value	Description
INITIALTHREADS	1	Number of instances of this service that the region controller starts during initialization of the Classic data server
MAXTHREADS	1	Maximum number of instances of this service that the region controller is allowed to start
MAXUSERS	100	Maximum number of user connections
RESPONSETIMEOUT	5M	Maximum amount of time to wait for response before terminating a connection
SAFEXIT	None	Name of the System Authorization Facility (SAF) system exit
SEQUENCE	0	Sequence number that is assigned to services
SMFEXIT	None	Name of the System Management Facility (SMF) accounting exit that reports clock time and CPU time for a user session
SQLSECURITY*	FALSE	Determines the level of access privilege verification that will be performed on operator commands
TRACELEVEL	4	Trace level

* The value of SQLSECURITY should always be set to FALSE for Data Replication for IMS and Data Replication for VSAM.

Region controller service

The region controller service is assigned to the CNTL service class. The task name for the CNTL service class is CACCNTL.

The region controller service monitors and controls the other services that run within the Classic data server.

The region controller directly or indirectly activates each service according to the configuration parameters that you define. The region controller starts, stops, and monitors the other tasks that run within the Classic data server.

The region controller also includes an IBM z/OS MTO (master terminal operator) interface that you can use to monitor and control an address space for a Classic data server.

The following table lists the configuration parameters for the CNTL service class.

Table 28. Configuration parameters for the CNTL service class

Parameter	Default value	Description
DBCSCODEPAGE*	0	Double-byte CCSID that the z/OS operating system uses where the Classic data server is running.
HOSTCODEPAGE	37	Host code page of the Classic data server
IDLETIMEOUT	5M	The amount of time that a service remains idle before it polls the local message queue for messages that are to be processed

Table 28. Configuration parameters for the CNTL service class (continued)

Parameter	Default value	Description
INITIALTHREADS	1	Number of instances of this service that the region controller starts during initialization of the Classic data server
MAXTHREADS	1	Maximum number of instances of this service that the region controller is allowed to start
MAXUSERS	100	Maximum number of user connections
RESPONSETIMEOUT	5M	Maximum amount of time to wait for response before terminating a connection
SEQUENCE	0	Sequence number that is assigned to services
TRACELEVEL	4	Trace level
* The DBCSCODEPAGE parameter does not apply to Data Replication for IMS.		

Configuration parameters for Classic data servers and services

Configuration parameters define settings for Classic data servers and for the services required for source and target data servers.

The global configuration parameters define server-wide settings. Standard configuration parameters define settings that are common to most services. All other configuration parameters are service-specific.

Global parameters for Classic data servers

Global parameters define configuration values that affect the entire Classic data server. Unlike other configuration parameters, global parameters are not related to a single service.

The following table summarizes the global configuration parameters and lists parameter default values.

Table 29. Global configuration parameters

Parameter	Default value	Description
DATACONVERRACT	0	Data conversion action
DATAVALIDATEACT	0	Data validation action
DECODEBUFSIZE	8192	Decode buffer size
FETCHBUFSIZE	32000	Size of the result set buffer returned to a client application
MESSAGEPOOLSIZE	16777216	Message pool size. This value is set during the installation and customization process.
REPORTLOGCOUNT	0	Log record limit for badly-formed data or conversion errors
STATICCATALOGS	0	Activates static catalog processing
TASKPARM	None	Specifies runtime options that are passed to subtasks through the IBM z/OS ATTACH macro

DATACONVERRACT:

The DATACONVERRACT parameter identifies the action for the Classic data server to take if a conversion error occurs when it converts numeric data between zoned and packed decimal formats and between binary and packed decimal formats.

Specification

Use: Global configuration parameter for the Classic data server.

Data type: INT

Default: 0 = FAIL

Valid values: 0 - 2

0 = FAIL
Specifies that the query ends with a -4908 return code that indicates non-valid mapped data.

1 = REPAIR
The Classic data server changes non-valid data to -99...99s, and the SQL statement ends successfully with a SQL_SUCCESS_WITH_INFO return code. One or more 002f0002 warning messages are also returned to the client application. A 002f0002 warning message is returned for each column each row that is changed to a value of -99..999s. The Classic data server does not write any log messages to indicate that conversion errors occurred.

Example: The Classic data server changes nonvalid data to the highest possible negative value for the column precision. A column with a precision of DECIMAL(3,0) changes to a value of -999, and a column with a precision of DEC(3,2) changes to a value of -9.99. If you sort the result set, rows with nonvalid data appear last.

2 = REPAIR REPORT
The Classic data server processes the conversion error as the REPAIR option describes. The Classic data server also writes the data conversion error message x002f0001 to the server log for each row that contains one or more conversion errors.

DATAVALIDATEACT:

The DATAVALIDATEACT parameter enables additional validation of data types that are not converted from source data to a different SQL data type. This parameter controls whether additional data type validation occurs and identifies the action for the Classic data server to take if a validation error occurs due to badly-formed data.

Specification

Use: Global configuration parameter for the Classic data server.

Data type: INT

Default: 0 = NO VALIDATION

Valid values: 0 - 3

0 = NO VALIDATION
The data is not checked. This is the default

1 = REPAIR
The Classic data server changes non-valid data as described in the table below, and the SQL statement ends successfully with a SQL_SUCCESS_WITH_INFO return code. One or more 002f0002 warning messages are also returned to the client application. A 002f0002 warning message is returned for each column and each row that is changed. The Classic data server does not write any log messages to indicate that conversion errors occurred.

2 = REPAIR REPORT
The Classic data server processes the validation error as the REPAIR option describes. The Classic data server also writes the data conversion error message x002f0001 to the server log for each row that contains one or more conversion errors.

3 = FAIL
The query ends with a -4908 return code that indicates non-valid mapped data.

Table 30. Data type validation

Data type value	SQL data type	Data validated	Repair value
P	DECIMAL	Packed decimal data	-9
V	VARCHAR	Field length	0: For a negative length value Maximum length: If the length is greater than the maximum length of the field
VB	VARBINARY	Field length	0: For a negative length value Maximum length: If the length is greater than the maximum length of the field
UP	DECIMAL	Packed positive number	-9
UF	INTEGER	Value between 0 and X'7FFFFFFF'	-9
UH	SMALLINT	Value between 0 and X'7FFF'	-9

DECODEBUFSIZE:

DECODEBUFSIZE defines the size of the DECODE buffer. This buffer is a staging area that decodes data from the network format into the host local data format.

Description

Data is taken from the FETCH buffer in pieces that are the size that is specified for the DECODE buffer. The data is converted until a single row of data is completely processed and returned to the application. For optimum use, set the DECODE buffer to a size that is at least equivalent to a single row of data.

The DECODEBUFSIZE and FETCHBUFSIZE parameters work together. If the DECODEBUFSIZE is omitted, its value is set to the value of FETCHBUFSIZE. If a value higher than the FETCHBUFSIZE is used, the value of DECODEBUFSIZE is set to the FETCHBUFSIZE. Thus, coordinate the settings of the DECODEBUFSIZE and FETCHBUFSIZE parameters.

Specifications

Use: Global configuration parameter for the Classic data server.

Data type: INT

Default: 8192

Valid values: 4096 - 64000

FETCHBUFSIZE:

The FETCHBUFSIZE parameter specifies the size of the result set buffer that is returned to a client application. You specify this parameter in the configuration file for the client application.

Description

When you set the fetch buffer size to 1, single rows of data are returned to the client application.

An appropriate FETCHBUFSIZE depends upon the average size of the result set rows that are sent to the client application and the optimum communication packet size. To improve performance, pack as many rows as possible into a fetch buffer. The default fetch buffer size is generally adequate for most queries.

If FETCHBUFSIZE is set smaller than a single result set row, the size of the actual fetch buffer that is transmitted is based on the result set row size. The size of a single result set row in the fetch buffer depends on the number of columns in the result set and the size of the data that is returned for each column.

The FETCHBUFSIZE and DECODEBUFSIZE parameters work together. If the DECODEBUFSIZE is omitted, its value is set to the value of FETCHBUFSIZE. If a value higher than the FETCHBUFSIZE is used, the value of DECODEBUFSIZE is set to the FETCHBUFSIZE.

You can use the following calculations to determine the size of a result set row in the buffer:

```
fetch buffer row size = (number of data bytes returned) x
(number of columns * 6)
```

Each fetch buffer has a fixed overhead. You can compute the overhead as follows:

```
fetch buffer overhead = 100 + (number of columns * 8)
```

If your applications routinely retrieve large result sets, contact your network administrator to determine the optimum communication packet size. Then, set the FETCHBUFSIZE to a size that accommodates large result sets.

Specifications

Use: Global configuration parameter for the Classic data server.

Data type: INT

Default: 32000

Valid values: 1 - 524288

MESSAGEPOOLSIZE:

The **MESSAGEPOOLSIZE** parameter specifies the size of the memory region in bytes for most memory allocation.

Description

Specify a region size that is at least 8 MB lower than the site limit, and use the greater of these values:

- 8 MB higher than the message pool
- 20% higher than the message pool

If the 8 MB gap between the region and the message pool is still not sufficient, increase this difference in increments of 8 MB.

Set the **MESSAGEPOOLSIZE** parameter to the greater of these values:

- 20% less than the region size
- 8 MB below the REGION value or 8 MB below any site limit imposed by exits.

If you increase the value of the **MESSAGEPOOLSIZE** parameter, set the region size higher to maintain the 8 MB gap.

Specification

Use: Global configuration parameter for the Classic data server.

Data type: INT

Default: 16777216

Valid values: 1048576 - upper limit not applicable

REPORTLOGCOUNT:

The REPORTLOGCOUNT parameter sets the maximum number of messages written to the log for badly-formed data or conversion errors.

Description

This parameter value prevents excessive logging to the log file for the Classic data server when a large amount of badly-formed data records are processed. The count controls the number of rows in the result set that generate the log messages for a given access to a table or view.

Specification

Use: Global configuration parameter for the Classic data server.

Data type: INT

Default: 0 = No logging limit

Valid values: 0 - 100000

STATICCATALOGS:

The STATICCATALOGS parameter activates static catalog processing for the system catalog data sets that are referenced by the Classic data server.

Description

With static catalog processing, the system catalog files are opened once for a query processor task. The system catalog files remain open until that Classic data server is shut down. In normal operating mode, the system catalogs are closed after the required table and column information is retrieved in order to process a query, for each query that is processed by the query processor.

Activate static catalog processing to substantially improve query performance in outer cursor and inner cursor situations when a large number of queries are issued serially.

Close the static catalog when the system is not updating catalogs information. Use this parameter when the Classic data server operates in production mode and the system catalogs are static.

Specification

Use: Global configuration parameter for the Classic data server.

Data type: INT

Default: 0

Valid values: 0 - 1

0 Close system catalog files and establish read locks for each query.

1 Close system catalog files when the Classic data server is shut down.

TASKPARM:

The TASKPARM parameter specifies IBM C runtime options that are passed to system child tasks through the z/OS ATTACH macro.

Description

One common use of this parameter is to pass TCP/IP information to the Communications Interface task. IBM Software Support can provide a current value.

Specifications

Use: Global configuration parameter for the Classic data server.

Data type: CHAR(64)

Default: None

Standard parameters for services

Standard service parameters are available in more than one service, but like service-specific parameters, you can define values separately for each service.

IDLETIMEOUT:

IDLETIMEOUT indicates idle time out.

Description

The IDLETIMEOUT value specifies the amount of time that a service remains idle before it polls the local message queue for messages that are to be processed.

Specification

Use: Configuration parameter for the Classic data server that applies to services that close connections after a specific time period.

Data type: TIME

Default values:

- 5M

Valid formats:

*n***MS** *n* milliseconds

*n***S** *n* seconds

*n***M** *n* minutes

*n***H** *n* hours

Setting the value to zero (0) indicates no time out. However, the setting the value to 0 is not recommended.

INITIALTHREADS:

INITIALTHREADS identifies minimum tasks.

Description

The value of INITIALTHREADS specifies the number of instances of this service that the region controller starts during initialization of the Classic data server. You can use the MTO START command to start an occurrence when needed, unless the service already has MAXTHREADS instances.

The default settings for this parameter are adequate for a default data server configuration. It is important to be cautious when you set the INITIALTHREADS parameter to a value that is greater than the default value. The region controller

might not be able to start as many threads as the number specified due to system resource restrictions.

Specification

Use: Configuration parameter for the Classic data server that applies to all services.

Data type: INT

Default value: 1: For all services

MAXTHREADS:

MAXTHREADS identifies the maximum number of instances for a service.

Description

The value of MAXTHREADS specifies the maximum number of instances of this service that the region controller can start.

The default settings for this parameter are adequate for a default data server configuration.

Specification

Use: Configuration parameter for the Classic data server.

Data type: INT

Default values:

- 1: For all services
- 10: For the query processor service only

Valid values: 0 and above.

1 If a service must be limited to a single instance.

0 The currently deployed number of threads remains the same. A new instance is not started.

MAXUSERS:

MAXUSERS identifies the maximum number of connections per task.

Description

The MAXUSERS value is the maximum number of connections that are allowed per instance of this service. Set this field to 1 to disable multi-tasking for all instances of this service.

Specification

Use: Configuration parameter for the Classic data server that is common to all services.

Data type: INT

Default value: Varies by service.

10 DRA

100 CNTL
INIT
LOG
MAA
OPER
PAA

Valid values: 1 and above

RESPONSETIMEOUT:

RESPONSETIMEOUT specifies the maximum amount of wait time for an expected response.

Specification

Use: Configuration parameter for the Classic data server.

Data type: TIME

Default value: 5M

Valid formats:

*n***MS** Number of milliseconds

*n***S** Number of seconds

*n***M** Number of minutes

If you set the RESPONSETIMEOUT value to zero (0), the response timeout function is disabled.

SEQUENCE:

Each service in the Classic data server is assigned a SEQUENCE number. The SEQUENCE parameter controls the order in which the region controller starts services in the Classic data server.

Description

The controller service starts first and is assigned a SEQUENCE value of 1. The logger service starts next and is assigned a SEQUENCE value of 2. You cannot modify the values of these core services.

When you add a service, that service is assigned the next available SEQUENCE value of 3 or higher. You can change the order in which services start by using SEQUENCE to reassign sequence numbers to services. You can modify these SEQUENCE values.

The non-core services can be assigned the same SEQUENCE number value. If this occurs, the services with the same SEQUENCE value will be started in alphabetical order based on service name.

SEQUENCE also affects termination processing for the Classic data server. Services stop in the reverse order that they start.

Specification

Use: Configuration parameter for the Classic data server that is common to all services.

Data type: INT

Default value: 0

Valid values:

- 1: Assigned to controller service
- 2: Assigned to logger service
- 3 - 999: Services are assigned the next available SEQUENCE value in the range of 3 - 999 when the service is added. You can modify these SEQUENCE values.

TRACELEVEL:

The TRACELEVEL parameter regulates the amount of information that tasks in the Classic data server record in the trace log.

Specification

Use: Configuration parameter that is common to all services.

Data type: INT

Default values:

- 4: For all services (exception: logger service)
- 1: For the logger service only. This value controls what other services send to the logger service and what the logger service writes to the log.

Valid values: 0 - 20:

20 No trace information generated

12 Identify non-recoverable error conditions

8 Identify all recoverable error conditions

4 Generate warning messages

3 Generate debugging information

2 Generate a detailed trace, but do not include binary buffers.

1 Generate function call information

0 Trace all

Important: Change this parameter only at the request of IBM Software Support. Settings lower than 4 cause response time degradation and higher CPU costs.

Service-specific parameters

Service-specific parameters pertain to a single service, so the values that you define affect only that service.

This section provides an explanation of each configuration parameter. “Summary of services” on page 159 summarizes the configuration parameters associated with each service.

APPLYBUFFERSIZE:

The APPLYBUFFERSIZE configuration parameter defines the size of the buffer that Data Replication for IMS uses to send changes from the source server to the target server.

Description

The default value for the APPLYBUFFERSIZE configuration parameter is 252K. In general the default value is sufficient. Use the default value unless otherwise directed by IBM Software Support.

Specifications

Use: Configuration parameter for the capture service.

Service class: CAP

Service task: CECCAP

Data type: INT

Default: 252K

COMMSTRING:

The COMMSTRING parameter specifies the protocol identifier and address for communication.

Description

The connection handler supports TCP/IPand XM protocols. The COMMSTRING value defines the protocol followed by the protocol specific information

- With a TCP/IP connection handler, you need the protocol identifier TCP, followed by the IP address of the machine that the Classic data server is running on, and the port number that is assigned to this server as a listen port. For example: TCP/*host-name*/*port-number*.

  The client connection string supports Internet Protocol Version 4 (IPv4) and Internet Protocol Version 6 (IPv6). For example:

  ```
  SET,CONFIG,SERVICE=INIT,COMMSTRING='TCP/0.0.0.0/9087'; (IPv4)
  SET,CONFIG,SERVICE=INIT,COMMSTRING='TCP/::/9087';         (IPv6)
  ```

  If you try to connect to the data server by using IPv6, you might need to provide a scope if you are using a link-local address. The scope is typically the network interface name that you specify following the IPv6 address. The format is `ipv6 address%scope`. For example:

  ```
  SET,CONFIG,SERVICE=INIT,COMMSTRING='TCP/fe80::xxxx:xxxx:xxxx:xxxx%INTF0001/9087';
  ```

Specification

Use: Configuration parameter for the connection handler service.

Service class: INIT

Service task: CACINIT

Data type: CHAR(64)

Default: TCP/0.0.0.0/9087

CONNECTINTERVAL:

The CONNECTINTERVAL parameter defines the frequency at which the DRA service retries connecting to IMS when IMS is not available.

Description

The default value for this parameter is 15S (seconds). This value indicates that the DRA service will retry failed connections to IMS every 15 seconds.

Specifications

Use: Configuration parameter for the DRA service.

Service class: DRA

Service task: CACDRA

Data type: TIME

Default: 15S

Valid values:

*n***MS** *n* milliseconds

*n***S** *n* seconds

*n***M** *n* minutes

*n***H** *n* hours

CONSOLELEVEL:

CONSOLELEVEL is a required parameter that controls when event messages are sent to the z/OS console.

Description

All event messages are written to the event log and the system trace. The value of the CONSOLELEVEL parameter controls when event messages are also routed to the z/OS console. When the trace level of an event message equals or exceeds the value specified for the CONSOLELEVEL parameter, that message is sent to the console.

Specifications

Use: Configuration parameter for the logger service.

Service class: LOG

Task name: CACLOG

Data type: INT

Default: 4

Valid values:

20 Generates no event messages to the console. Messages are written to the event log.

4 Generates the following event messages:

- Subscription group operation messages issued from the capture, apply, and administration services
- Stream activation and destruction messages issued from the capture and log reader services

Specifying a value less than 4 generates more console messages and increases the number of messages in the console buffers.

3 Generates the following event messages:

- Replication messages issued from the the capture, apply, and administration services
- Table, view, and DBMS object cache operation messages issued from the administration service
- SSID start and stop messages issued from the IMS log reader service

2 Generates the following event messages:

- Open and close messages issued from the IMS log reader service
- Roll-off cache maintenance messages issued from the capture service

0 Writes all event messages to the console.

DEPGRAPHHASHSZ:

The DEPGRAPHHASHSZ configuration parameter controls the size of the hash table that contains keys for each component used in dependency analysis.

Description

The default value for DEPGRAPHHASHSZ is 64 kilobytes (KB). Specifying a larger value might increase parallelism. However, increasing the value of this parameter consumes more memory.

Specifications

Use: Configuration parameter for the apply service.

Service class: APLY

Service task: CECAPLY

Data type: INT

Default: 64 KB

DEPGRAPHKEYS:

The DEPGRAPHKEYS configuration parameter identifies the level of granularity that you want dependency analysis to use for parallelism decisions.

Description

Dependency analysis supports a two-level hierarchy to identify the objects that were updated by a source UOR. You use the DEPGRAPHKEYS parameter to control whether only a single level or both levels are used by default.

The first part of the key identifies the resource that was updated. Data Replication for IMS uses the following names as resource identifiers:

- The area name that was updated for a DEDB database.
- The partition name that was updated for a HAL-DB database.
- The name of the DBD for other database types.

The second part of the key is the root segment sequence field that identifies the IMS database record that was updated.

If you experience frequent deadlock situations due to parallel apply attempting to concurrently process UORs that have keys that randomize to the same block, setting DEPGRAPHKEYS to 1 might improve performance. Using this option prevents deadlocks from occurring because updates to the same resource are applied in the order in which they occurred at the source site. In situations where you have DEDB databases with a large number of areas and your typical workloads are restricted to updating a single area, using this “area level” dependency analysis approach can provide better throughput than using the default full-key analysis that occurs when DEPGRAPHKEYS is set to 2 (the default).

Specifications

Use: Configuration parameter for the apply service.

Service class: APLY

Service task: CECAPLY

Data type: INT

Default: 2

Valid values: 1 - 2

DEPGRAPHMEMORY:

The DEPGRAPHMEMORY configuration parameter controls the size of the 64-bit memory object that the target server obtains for dependency analysis.

Description

The default value for DEPGRAPHMEMORY is 1000 megabytes (MB). In general the default value is sufficient. If you need to change the amount of memory used for dependency analysis, you can increase or decrease this value.

Specifications

Use: Configuration parameter for the apply service.

Service class: APLY

Service task: CECAPLY

Data type: INT

Default: 1000 MB

DEPGRAPHUORLIMIT:

The DEPGRAPHUORLIMIT configuration parameter controls how the target server tracks UOR information for dependency analysis.

Description

UORs with fewer changes than the specified value use two-component dependency analysis -- resource name and key value. UORs that exceed the specified value use single-component dependency analysis -- resource names only. If a UOR has more resource names than the value specified, the UOR will cause serialization for the subscription.

The default value for DEPGRAPHUORLIMIT is 200 changes. Specify the default value unless otherwise directed by IBM Software Support.

Specifications

Use: Configuration parameter for the apply service.

Service class: APLY

Service task: CECAPLY

Data type: INT

Default: 200

DISPLAYLOG:

The DISPLAYLOG parameter allows you to view log messages for the logger service.

Description

This parameter controls whether log records are mirrored to the data set that is specified in the SYSOUT DD statement. The default data set is the system output data set (SYSOUT).

Specifications

Use: Configuration parameter for the logger service.

Service class: LOG

Task name: CACLOG

Data type: Boolean

Default: FALSE

DONEUORLIMIT:

The DONEUORLIMIT configuration parameter allows you to control how many DONEUORS segments can exist for a subscription when completed UOR tracking is active.

Description

When completed UOR tracking is active, the DONEUORLIMIT parameter identifies the frequency at which serialization occurs for UORs that require completed UOR tracking for a subscription. This allows you to set an upper limit per subscription which allows you to more easily estimate the space requirements for the bookmark database.

For example, when the DONEUORLIMIT parameter is set to the default value of 100000 the following processing occurs. Upon receipt of the 100,000th UOR that requires completed UOR tracking from the source server, no dependency analysis information is provided for that UOR. This forces the UOR to be applied serially, which also establishes a 'contiguous committed UOR' condition. During commit processing for one of these kinds of UORs, the existing PAWINDOW segment instance for the subscription is deleted which clears existing DONEUORS tracking information for that subscription.

This kind of processing is performed at DONEUORLIMIT intervals. Using the example of the value set to 100,000 , one of these forced serialization points is established when the 100,000th UOR, 200,000th UOR, etc. that requires completed UOR tracking is received for a subscription.

Setting this parameter only takes effect when the DONEUORTRACKING configuration parameter is set to TRUE.

Specifications

Use: Configuration parameter for the apply service.

Service class: APLY

Service task: CECAPLY

Data type: INT

Default: 100000

Valid values: 1000 and above

DONEUORTRACKING:

The DONEUORTRACKING configuration parameter enables completed UOR tracking.

Description

Completed UOR tracking allows UORs that contain changes that require serialization to be processed in parallel. These kinds of UORs cannot be re-processed using adaptive apply techniques. Instead a list of these kinds of UORs that were applied at the target site are maintained in the bookmark database.

When DONEUORTRACKING is set to TRUE, completed UOR tracking is enabled. When it is set to FALSE, these kinds of UORs force serial apply processing for a subscription.

Specifications

Use: Configuration parameter for the apply service.

Service class: APLY

Service task: CECAPLY

Data type: Boolean

Default: FALSE

Valid values: TRUE or FALSE

DRATABLESUFFIX:

The DRATABLESUFFIX parameter identifies the suffix of the DRA startup table.

Description

Use DRATABLESUFFIX to specify the suffix of the load module name that you created for IMS DRA initialization.

Specification

Use: Configuration parameter for the IMS DRA access service.

Service class: DRA

Task name: CACDRA

Data type: CHAR(3)

Default value: None

DRAUSERID:

The DRAUSERID parameter identifies the DRA user ID.

Description

Use DRAUSERID to specify the default DRA user ID to use for connecting to and registering with DBCTL. The DRA user ID is the name by which the Classic data server is known to the IMS database manager subsystem, DBCTL.

Specification

Use: Configuration parameter for the IMS DRA access service.

Service class: DRA

Task name: CACDRA

Data type: CHAR(9)

Default value: None

EIFEVENTSERVERS:

The EIFEVENTSERVERS parameter is maintained as a service list that defines the destination of Event Integration Facility (EIF) events.

Description

The logger service sends EIF events to event servers, such as the Event/Automation Service that IBM Tivoli NetView® for z/OS provides. You use the EIFEVENTSERVERS parameter to define the URL of the event server that will receive the EIF events. The event server that you specify must be active and listening on the IP address and port number that you define. Changes made to this service list are effective immediately.

Specifications

Use: Configuration parameter for the logger service.

Service class: LOG

Service task: CACLOG

Data type: CHAR

Default: None

Valid values:

The value of the EIFEVENTSERVERS parameter consists of the protocol identifier SKT, followed by the IP address and port number of the event server. You specify the URL in the following format:

protocol/ip_address/port_number

protocol
: The logger service supports the SKT protocol. Specify SKT as the protocol value.

ip_address
The IP address of the event server. You can specify an Internet Protocol Version 4 (IPv4) address or an Internet Protocol Version 6 (IPv6) address. You can also specify a host name.

port_number
The port number that is assigned to the event server as a listen port.

The URL is not validated until it is used to connect to an event server.

Example

The following command sets the value of the EIFEVENTSERVERS parameter for a logger service with the name `LOG` in a service list entry. In this example, the parameter value is a URL that defines the protocol `SKT`, IP address `10.1.1.1`, and port number `9087`.

```
ADD,CONFIG,SERVICELIST=EIFEVENTSERVERS,SERVICE=LOG,VALUE=SKT/10.1.1.1/9087
```

EVENTLOG:

EVENTLOG is an optional parameter identifies the name of the event message log that is defined to the logger service.

Description

The logger service writes event messages to the log specified in the EVENTLOG parameter. This parameter identifies a z/OS system log stream. The event log file cannot be shared among multiple Classic data servers. You can use one event log file with one Classic data server.

If you do not specify the EVENTLOG parameter, event messages are not captured. Event messages will not be available for retrieval by the Classic Data Architect. In this case, the Classic data server formats event messages to SYSPRINT and incurs processing overhead at runtime.

Specifications

Use: Configuration parameter for the logger service.

Service class: LOG

Task name: CACLOG

Data type: CHAR (26)

Default: None

HOSTCODEPAGE:

The HOSTCODEPAGE parameter identifies the CCSID that the z/OS operating system uses where the data server is running.

Description

Data Replication for IMS must be performed between sites that operate with identical z/OS code pages.

Specification

Use: Configuration parameter for the region controller service.

Service class: CNTL

Service tasks: CACCNTL

Data type: INT

Default: 37

Valid values: 0 - 99999

IMSBOOKMARKDB:

The IMSBOOKMARKDB parameter specifies the name of the database defined to IMS to use as a bookmark database for replication.

Description

The IMSBOOKMARKDB value must match the name of the database that IMS uses as the bookmark database. All apply PSBs that this target server uses must have a PCB for this database name.

Specifications

Use: Configuration parameter for the apply service.

Service class: APLY

Service task: CECAPLY

Data type: CHAR(8)

Default: None

Valid values: Any valid IMS database name.

IMSV11CMPAT:

The IMSV11CMPAT configuration parameter controls activation of new apply processing that eliminates the use of database scanning logic. Instead this processing uses SSAs that contain O command codes to locate a target segment instances that does not have a unique sequence field.

Description

The default value for IMSV11CMPAT is FALSE. Set this parameter to TRUE if you are running IMS version 11 at the target site. Otherwise using the default is more efficient.

If you are capturing subset pointer updates or positioning information at the source (you included the SSPCMD or INPOS options on any EXIT definition in a source database), you must set IMSV11CMPAT to FALSE to prevent replication failures.

Specifications

Use: Configuration parameter for the apply service.

Service class: APLY

Service task: CECAPLY

Data type: Boolean

Default: FALSE

Valid values: TRUE or FALSE

INACTTHRESHOLD:

INACTTHRESHOLD is an optional parameter that defines the frequency for issuing the inactivity message CECZ0400W for a given subsystem.

Description

In an IMS data sharing environment, changes are captured from multiple DB/DC or DBCTL subsystems. The CECZ0400W message is issued in situations where the activity of one or more subsystems is not detected for an extended period of time.

The value that you specify for the INACTTHRESHOLD parameter controls how frequently the CECZ0400W message is issued. You can specify any value, in seconds, between the minimum value of 1 and the maximum value of 86400 (24 hours). The new value takes effect after the previous INACTTHRESHOLD value is reached.

Specifications

Use: Configuration parameter for the IMS log reader service.

Service class: LRSI

Task name: CECLRS

Data type: INT

Default: 30s

Valid values: 1s - 86400s

LISTENURL:

LISTENURL is a required parameter specifies the protocol identifier and address for communication for the apply service.

Description

The apply service supports the SKT (asynchronous TCP/IP sockets). The LISTENURL value defines the protocol that the protocol specific information follows.

For a connection handler that uses TCP/IP protocols, you need:

- The protocol identifier SKT, followed by the IP address of the machine that the Classic data server is running on
- The port number that is assigned to this Classic data server as a listen port.

Specifications

Use: Configuration parameter for the apply service.

Service class: APLY

Service task: CECAPLY

Data type: CHAR

Default: None

Example

`SKT/host-name/port-number`

LOGBUFSIZE:

The LOGBUFSIZE parameter defines the size of the log buffer.

Specifications

Use: Configuration parameter for the logger service.

Service class: LOG

Task name: CACLOG

Data type: INT

Default: 65536

Valid values: 4096 - 1024000

LOGURL:

The LOGURL parameter identifies the communication protocol for the logger service.

Description

You can use LOGURL to override the protocol defined for local queues. This parameter is typically used for XM queues.

Specifications

Use: Configuration parameter for the logger service.

Service class: LOG

Task name: CACLOG

Data type: CHAR(32)

Default: None

Example

The following sample command sets the value of the **LOGURL** parameter for a logger service with the name LOG.

```
F <Data-Server-Name>,SET,CONFIG,SERVICE=LOG,LOGURL=XM1/DSLG1/LOGQ1/256
```

MAXWRITERTHREADS:

The MAXWRITERTHREADS configuration parameter defines the maximum number of writer threads that the target server can start.

Description

For parallel processing, you can specify the default value for the WRITERTHREADS parameter and set the value of the MAXWRITERTHREADS parameter to the maximum amount of parallelism that you allow in your environment. The target server will add writer threads when the amount of parallel work increases up to the value specified for MAXWRITERTHREADS.

The value specified for MAXWRITERTHREADS should match the MAXTHRD value in the DRA start-up table that the target server is using. Each writer thread calls dependency analysis to get the next available unit of work. When work is available, one of the first things that the writer thread does is schedule the apply PSB which requires an available DRA thread. If there are no DRA threads available, the schedule request simply waits until a DRA thread becomes available. Therefore, if you have more writer threads than DRA threads you will not get the throughput that you are expecting.

The MAXWRITERTHREADS parameter also has a practical upper limit based on the number of subscriptions that exist and the maximum number of apply PSBs that can be scheduled for each subscription. For maximum throughput, you should set MAXWRITERTHREADS to the sum of all apply PSBs that can be scheduled.

Specifications

Use: Configuration parameter for the apply service.

Service class: APLY

Service task: CECAPLY

Data type: INT

Default: 10

Valid values: 1 - 255

MSGLIST:

The MSGLIST parameter is maintained as a service list. You maintain service list parameters by using service list commands.

Description

You specify a message list as *message-number/destination*:

Message-number
The message number must begin with prefix CEC and contain nine characters.

Destination
You can specify one of the following destinations:

- CONSOLE: z/OS console
- DIAGLOG: Diagnostic trace log
- EVENT: Event log
- SUPPRESS: No destination

The service list configuration commands create or update the destination (or suppression) for a particular message based on message IDs. You can change a message destination to another destination. The destination hierarchy is as follows:

- CONSOLE: The message is routed to the console, event log, and diagnostic log.
- EVENT: The message is routed to the event log and the diagnostic log.
- DIAGLOG: The message is routed to the diagnostic log only.
- SUPPRESS: The message does not appear in any destination.

Changes to this service list are effective immediately.

Specification

Use: Configuration parameter for the logger service.

Service class: LOG

Task name: CACLOG

Data type: CHAR

Default: None

NMICOMMSTRING:

The NMICOMMSTRING parameter specifies the communication path for the Network Management Interface (NMI) AF_UNIX domain socket.

Description

The value of the NMICOMMSTRING configuration parameter defines the fully-qualified path and file name of the AF_UNIX domain socket that is used for NMI client connections to the NMI. You specify this value in the following format:

```
NMICOMMSTRING='/var/sock/uniqueName'
```

where `'/var/sock/uniqueName'` is the path and unique name of the file. The maximum length that you can specify is 60 characters.

Recommendation: Use single quotes to preserve the case (uppercase or lowercase) of the characters in the AF_UNIX domain socket path. The case of the characters specified for this value must match the case of the Unix System Services file system path.

You can dynamically set or change the value of the NMICOMMSTRING configuration parameter by using the Classic Data Architect or the MTO SET,CONFIG,SERVICE command. The change takes affect when you stop and restart the monitoring service or the Classic data server.

Specification

Use: Configuration parameter for the monitoring service.

Service class: MAA

Service task: CECMAA

Data type: CHAR

Default: None

NOTIFICATIONURL:

NOTIFICATIONURL is a required parameter that identifies the TCP/IP address and port used for event notification.

Description

The TCP/IP address and port specified for the NOTIFICATIONURL parameter and for the configuration table module (CECE1OPT) for the notification exit must be the same.

Specifications

Use: Configuration parameter for the IMS log reader service.

Service class: LRSI

Task name: CECLRS

Data type: CHAR (64)

Default: None

RECONNECTWAIT:

The RECONNECTWAIT parameter defines the minimum amount of time that the DRA service waits after an IMS disconnect before attempting to reconnect to IMS.

Description

This wait time ensures that the DRA service does not try to reconnect to IMS while IMS is still in the process of stopping and prevents abends in the DBCTL interface.

The RECONNECTWAIT parameter is enforced when the console message CAC00137W is issued to ensure that IMS has enough time to stop before the DRA service attempts to reestablish a connection.

The default value for this parameter is 1M (minute). This value indicates that the DRA service will start trying to connect to IMS one minute after a disconnect is received from IMS and message CAC00137W is issued.

Specifications

Use: Configuration parameter for the DRA service.

Service class: DRA

Service task: CACDRA

Data type: TIME

Default: 1M

Valid values:

*n***MS** *n* milliseconds

*n***S** *n* seconds

*n***M** *n* minutes

*n***H** *n* hours

SAFEXIT:

The SAFEXIT parameter specifies the System Authorization Facility (SAF) system exit that performs authorization checks for the administration, monitoring, and console connections to the Classic data server.

Description

The values that you specify for the SAFEXIT parameter control the actions that a user can perform when connected to a Classic data server for the following types of connections:

- Administration connections from the Classic Data Architect. You use administration connections to view and edit subscription definitions, start and stop replication, and view system events.
- Monitoring connections established by the Classic Data Architect separately from administration connections. These connections are authenticated at the z/OS host. Monitoring connections also include Network Management Interface (NMI) connections to the z/OS data server. You use NMI connections to retrieve metrics data and for access to subscription states or statuses.
- Console connections from the Classic Data Architect that allow remote operators to issue console commands to the Classic data server.

You specify the optional parameters for the SAF exit in the following format:

```
CACSX04{,optional-parameters...}
```

If you do not specify any optional parameters, the SAF exit load module CACSX04 activates user ID and password authentication when a user connects to a Classic

data server. The optional parameters provide additional security checking that the administration service, monitoring service, and operator service perform.

All connections

The following optional parameter for validation of IP addresses applies to all connections:

NETACCESS=Y/N
Indicates whether the exit should validate the IP address of the connected client to authenticate access to the Classic data server.

Set the value to Y when the IP address of the connected client is known and the SERVAUTH parameter of the RACROUTE REQUEST=VERIFY invocation is supplied. The RACROUTE operation is successful when the associated user ID has at least READ-level access rights to the network security zone resource. If the security system indicates that it cannot make a decision in response to the request because a corresponding network security zone resource profile does not exist, the SAF exit regards the response as Access Denied.

A value of N indicates that the SERVAUTH parameter is omitted from the RACROUTE REQUEST=VERIFY invocation. This is the default.

Administration connections

The following optional parameters for the administration service activate security checking for client connections:

VALIDATE=*Y/N*
Indicates whether the SAF exit should perform resource class checking for each connected user. The default value for VALIDATE is Y. If Y is specified, resource access checking occurs when users make requests to either query data from the Classic data server or update information about replication.

- For query requests, READ level access is checked.
- For operations that change subscription information or the state of subscriptions (for example, starting or stopping subscriptions), CONTROL access is checked.

If N is specified, resource class checking is not performed. This is the default.

ADMCLASS=*administrator-class-name*
Indicates the name of the security class that contains a profile that requires access authentication.

This parameter is valid if VALIDATE=Y on the administration service for the SAF exit. If this parameter is not specified, SERVAUTH is the default security class.

ADMPROF=*administrator-profile-name*
Indicates the name of the resource profile that requires access authentication.

This parameter is valid if VALIDATE=Y on the administration service for the SAF exit. If this parameter is not specified, CEC.ADMIN is the default profile name.

Monitoring connections

The following optional parameters for the monitoring service activate security checking for client connections from the Classic Data Architect and NMI clients:

VALIDATE=*Y/N*
Indicates whether the SAF exit should perform resource class checking for each connected user. If Y is specified, resource access checking occurs when users make requests to retrieve metrics information. READ level access is checked. If N is specified, resource class checking is not performed. This is the default.

MONCLASS=*monitor-class-name*
Indicates the name of the security class that contains a profile that requires access authentication.

This parameter is valid if VALIDATE=Y on the monitoring service for the SAF exit. If this parameter is not specified, SERVAUTH is the default class name.

MONPROF=*monitor-profile-name*
Indicates the name of the resource profile that requires access authentication.

This parameter is valid if VALIDATE=Y on the monitor service for the SAF exit. If this parameter is not specified, CEC.MONITOR is the default profile name.

Console connections

The following optional parameters for the operator service activate security checking for client connections from the Classic Data Architect for issuing console commands:

VALIDATE=*Y/N*
Indicates whether the SAF exit should perform resource class checking for each connected user. If Y is specified, CONTROL level access is checked when users issue console commands through the remote operator. If N is specified, resource class checking is not performed. This is the default.

OPERCLASS=*operator-class-name*
Indicates the name of the security class that contains a profile that requires access authentication.

This parameter is valid if VALIDATE=Y on the operator service for the SAF exit. If this parameter is not specified, SERVAUTH is the default class name.

OPERPROF=*operator-profile-name*
Indicates the name of the resource profile that requires access authentication.

This parameter is valid if VALIDATE=Y on the operator service for the SAF exit. If this parameter is not specified, CEC.OPER is the default profile name.

Specification

Use: Configuration parameter for the administration, monitoring, and operator services.

Service classes: PAA, MAA, OPER

Service tasks: CECPAA, CECMAA, CACOPER

Data type: CHAR

Default: None

SMFEXIT:

SMFEXIT reports clock time and CPU time for an individual user session.

Description

You can supply the following values for the SMFEXIT parameter:

RECTYPE=*nnn*
This is a required parameter that defines the SMF user record type. This parameter contains a numeric value between 128 and 255.

SYSID=*xxxx*
This is a required parameter that contains the primary JES subsystem ID. SYSID can be a maximum of four characters.

Specification

Use: Configuration parameter for the operator service.

Service class: OPER

Data type: CHAR(64)

Default value: None

SSIDEXCLUDELIST:

The SSIDEXCLUDELIST parameter is an optional parameter that represents the IMS subsystem exclusion list for the log reader service.

Description

SSIDEXCLUDELIST is maintained as a service list. You can specify the name of an IMS subsystem ID (SSID) to exclude from ordering decisions.

By default, all subsystems that exist in the RECON that the Classic data server references are eligible to be captured. The actual state of each subsystem and the starting position of a change stream determine whether a subsystem change needs to be captured.

Specification

Use: Configuration parameter for the IMS log reader service.

Service class: LRSI

Task name: CECLRS

Data type: CHAR

Default: None

STREAMNAME:

The STREAMNAME parameter identifies the name of the log stream that is defined in the z/OS system logger.

Description

The logger service writes log records to the z/OS log stream specified in the STREAMNAME parameter. The log stream is used for the diagnostic log that runs in the Classic data server.

Specifications

Use: Configuration parameter for the logger service.

Service class: LOG

Task name: CACLOG

Data type: CHAR(26)

Default: None

STRICTVALIDATION:

The STRICTVALIDATION configuration parameter controls whether replication is allowed to start when differences in the source and target DBD attributes are detected.

Description

Setting the value of the STRICTVALIDATION parameter to the default value of TRUE means that source and target DBD definitions must be identical for replication to occur for a subscription. When STRICTVALIDATION is set to FALSE, differences detected in DBD attributes will result in a warning message and replication is allowed to start for the subscription.

For more detailed information, see:

- DBD validation for a list of the attributes that are validated.
- Synchronization of source and target IMS databases for an overview of the types of differences that can exist between a source and target database that will not cause replication failures.

Deactivation of strict validation is intended to only be enabled for brief periods of time when you know that the source and target database structures are different.

Specifications

Use: Configuration parameter for the apply service.

Service class: APLY

Service task: CECAPLY

Data type: Boolean

Default: TRUE

Valid values: TRUE or FALSE

UORGROUPCOUNT:

The UORGROUPCOUNT parameter identifies the number of messages that the capture service should group into a common unit of recovery (UOR) during replication.

Description

The source server, by default, respects the transaction boundaries of the source database managements system (DBMS) and replicates transactions within those boundaries. This means that each transaction is individually managed at the target server.

For small transactions, you might achieve higher throughput and lower CPU by grouping multiple small transactions into a single larger transaction. You can use the UORGROUPCOUNT parameter to instruct the source server to combine smaller transactions together before they are sent to the target server.

The UORGROUPCOUNT parameter represents the maximum number of updates to group together. The source server combines transactions until the number of updates will exceed the UORGROUPCOUNT value if the next transaction is grouped. At this point, the group is sent to the target server and a new group is started. A group is also completed and a new group is started when the following situations occur:

- The source server runs out of committed work in the capture cache
- The group is serial and the next UOR is parallel, or the group is parallel and the next UOR is serial

Grouping transactions with the UORGROUPCOUNT parameter does not cause an original transaction from the source DBMS to be divided into multiple groups. An original transaction is always contained in a single group when transaction grouping is active.

Specifications

Use: Configuration parameter for the capture service.

Service class: CAP

Task name: CECCAP

Data type: INT

Default: 1

The default value indicates that transaction grouping is inactive because each transaction meets or exceeds the UORGROUPCOUNT value and causes the UOR to be sent to the target.

USERSUBPOOLMAX:

The USERSUBPOOLMAX parameter determines the maximum size of a user sub pool.

Description

A user sub pool can grow to 256 times the USERSUBPOOLMAX value, resulting in the maximum user sub pool size in bytes. For example, if you set USERSUBPOOLMAX to the default value of 8192, the memory requirements for all of your queries for the current connection cannot exceed 2MB.

Recommendation: Increase the value of USERSUBPOOLMAX in high-volume environments to ensure that there is enough storage for the apply service.

Configure this parameter carefully to avoid using too much data server storage.

Specification

Use: Configuration parameter for the apply service.

Data type: INT

Default value: 8192

WRITERTHREADS:

The WRITERTHREADS configuration parameter defines the number of writer threads to start when the target server is started.

Description

The Classic data server uses the value of the WRITERTHREADS configuration parameter to determine the number of writer threads to start when the target server is started.

The default value is 3. In most cases the default value is sufficient. The value of WRITERTHREADS must be less than or equal to the value of the MAXWRITERTHREADS configuration parameter.

For parallel processing, you can specify the default value for the WRITERTHREADS parameter and set the value of the MAXWRITERTHREADS parameter to the maximum amount of parallelism that you will allow in your environment. The target server will add writer threads when the amount of parallel work increases up to the value specified for MAXWRITERTHREADS. You can also consider increasing the value of WRITERTHREADS. Prestarting the writer threads to handle parallelism can help improve performance.

Specifications

Use: Configuration parameter for the apply service.

Service class: APLY

Service task: CECAPLY

Data type: INT

Default: 3

Valid values: 1 - 255

Command reference

Look up syntax and explanations for commands that help you to manage subscriptions, Classic data servers, and configurations.

Subscription management commands

You can use the commands for replication management to start and stop replication, modify subscriptions, and gather metric data about subscriptions running on source and target servers.

The administration service supports the operational and administration commands for replication management. The monitoring service supports the commands that report metric data for subscriptions and replication mappings.

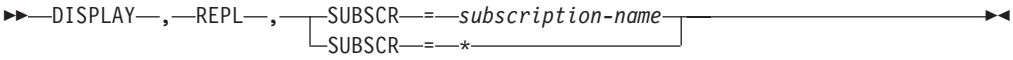
You can issue the subscription management commands by using the master terminal operator (MTO) interface. You can also use the Classic Data Architect to manage subscriptions and obtain subscription metrics.

DISPLAY,REPL command

You can use the DISPLAY,REPL command to query a source server or a target server for metrics for subscriptions, replication mappings, and threshold values. The commands generate reports that are issued to the console as WTO messages.

Displaying subscription summary

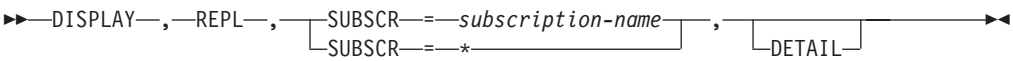
This command displays subscription metrics for the specified subscription or for all subscriptions.



The report output provides latency values and the state of the subscription found on the target server. A warning message displays on the report if no subscriptions are found.

Displaying subscription detail

This command displays replication mapping and replication object metrics for the specified subscription or for all subscriptions.



The report output provides the same subscription metrics as the subscription summary report with the addition of replication mapping information.

A warning message displays on the report if no subscriptions are found.

Displaying replication mappings

This command displays metrics for all subscriptions associated with a replication object.

The report provides information about the subscriptions that contain replication mappings for the specified replication object. It includes the status of subscriptions and replication mappings.

Displaying subscription workload names and thresholds

This command displays workload names and thresholds for the specified subscription or for all subscriptions.

The report output provides a summary of the subscription workload names and threshold values for the specified subscriptions. Only named workloads are displayed.

You can use the report output to verify the workload names and threshold values that you specified by using the SET,REPL command. You can also use the report output to identify the threshold values set for the GDPS/Active-Active workloads associated with the subscription.

Parameters

subscription name
: The name of the subscription to display. You can specify the subscription name as an identifier with or without quotation marks.

asterisk (*)
: An asterisk (*) displays information for all subscriptions. You can also specify an * as a wildcard character at the end of a partial subscription name. For example, specifying SUBSCR=ABC* will display all the subscription names that begin with ABC.

DETAIL
: Displays detailed information about subscriptions and replication mappings.

MAPPING
: Displays information about subscriptions that contain replication mappings.

replication object
: The name of the replication object to display. You can specify the replication object name as an identifier with or without quotation marks.

THRESHOLDS
: Displays information about subscriptions that contain workload names and threshold settings.

  You can associate up to three threshold sets with a subscription. A threshold set contains both a workload name and threshold values.

  The workload name assigned with the threshold set is included in the EIF event that the Classic data server emits when the associated threshold values are exceeded.

- The first workload is assigned the same name as the subscription name when any of its threshold values are set to a non-zero value. The first workload name is cleared when all of its threshold values are set to zero values.
- Threshold sets 2 and 3 are named or cleared based on the settings that you define by using the SET,REPL,SUBSCRIPTIONS,THRESHOLDS MTO command. The Classic data server uses these threshold values to determine when to emit EIF events.

If you do not participate in GDPS/Active-Active, only the threshold set THRESHOLDS is needed. For GDPS/Active-Active, you can display threshold sets that reflect the threshold values for the GDPS/Active-Active workloads.

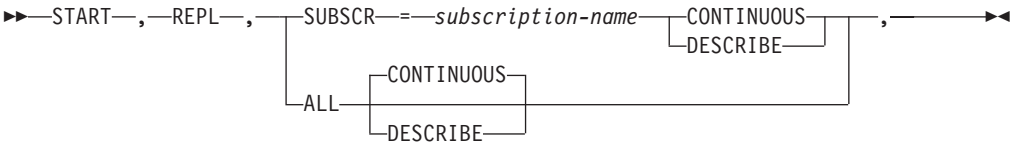
Primarily, a Classic data server emits EIF events to an event server. The event server is a server that consumes EIF events for GDPS/Active-Active services or an event server of your own design.

START,REPL command

You can use the START,REPL command to start replication.

Starting replication

This command starts replication activity for the specified subscription or for all subscriptions.



Parameters

subscription name
The name of the subscription that contains the DBDs that you want to replicate. You can specify the subscription name as an identifier with or without quotation marks.

ALL
Starts all subscriptions that are available for processing. Any subscriptions that are locked for other processing (for example, for replication processing or administrative updates) are not started. Continuous replication is the default.

CONTINUOUS
Starts continuous replication for the subscription. Replication continues until a command to stop replication is received.

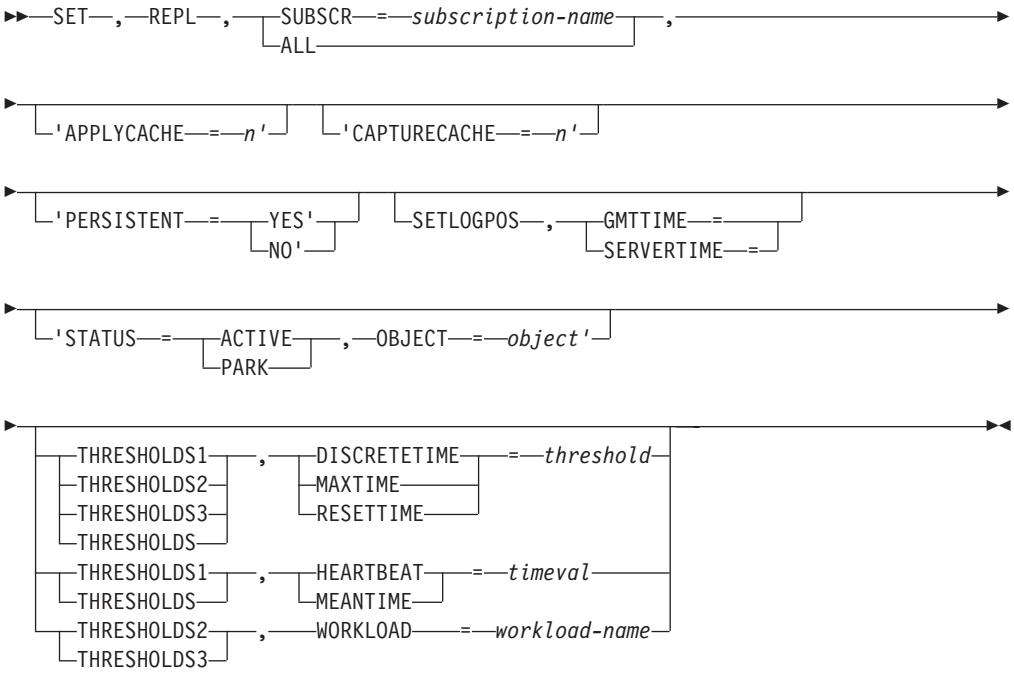
DESCRIBE
Activates the Describe process. This process validates metadata at the source server and transfers metadata about active subscriptions and replication mappings to the target server.

SET,REPL command

You can use the SET,REPL command to modify existing subscriptions.

Modifying subscription information

You can use the following SET command to modify the attributes of an existing subscription or of all subscriptions.



Parameters

subscription name
The name of the subscription to modify. You can specify the subscription name as an identifier with or without quotation marks.

ALL
Modifies the specified attributes for all subscriptions. The ALL parameter does not apply to setting thresholds or workload names. Therefore you cannot specify both the ALL parameter and the THRESHOLDS*n* or THRESHOLDS parameter, or both the ALL parameter and WORKLOAD names.

APPLYCACHE
The size of the apply cache. A value of 64 to 2048 MB is allowed.

CAPTURECACHE
The size of the capture cache. A value of 64 to 2048 MB is allowed.

PERSISTENT
Automatically restarts replication for the specified subscription. Specify this keyword only for subscriptions that use continuous replication. Valid values are YES or NO.

SETLOGPOS
Updates the bookmark information stored at the target server.

Specify the time in the format YYYY-MM-DD-hh.mm.ss.thmiju:

- YYYY-MM-DD is a calendar date
- hh.mm.ss.thmiju is a time of day

Valid values: You can specify a value to get the current time or specify GMTTIME or SERVERTIME.

GMTTIME
The Greenwich Mean Time (GMT) time. You must specify this value up to the ss (seconds). For example:

```
SET,REPL,SUBSCR=subscription name,SETLOGPOS,
GMTTIME=2011-04-14-18.01.00.123456
```

SERVERTIME
The server time. Specify the local time of the server time zone. You must specify this value up to the ss (seconds) value. For example:

```
SET,REPL,SUBSCR=subscription name,SETLOGPOS,
SERVERTIME=2011-04-14-18.01.00.123456
```

Important: If you modify a bookmark, you might need to refresh the entire subscription and restart replication for the subscription to recreate a valid bookmark. You should only change a bookmark under the direction of IBM Software Support.

"STATUS= <*status*>,OBJECT=<*object*>"
The status of the subscription. Valid values:

ACTIVE
For any subscription, you can set which mappings you want to be set as active for replication. If a mapping is not set as active, it will not replicate when replication is started for the subscription.

PARK For any subscription, you can park replication mappings. If a mapping is set as parked, it will not replicate when replication is started for the subscription.

The STATUS and OBJECT keywords must be enclosed in quotation marks.

THRESHOLDS*n*|THRESHOLDS,<*keyword*>=<*value*>
Sets the apply latency threshold values and workload names of the threshold sets for a subscription. The Classic data server uses these values to determine when to emit Event Integration Facility (EIF) events to an event server.

You can define up to three threshold sets for a subscription in the format `THRESHOLDS*n*,*keyword=value*`.

THRESHOLDS*n*
The threshold set identifier for the subscription:

- THRESHOLDS1
- THRESHOLDS2
- THRESHOLDS3
- THRESHOLDS

The suffix `n` designates 1, 2, or 3 to correspond to the three sets of apply latency thresholds for a subscription. If you do not specify a threshold set identifier, the command is issued against threshold set 1.

keyword
The threshold keyword:

- DISCRETETIME
- MAXTIME
- RESETTIME
- HEARTBEAT

- MEANTIME
- WORKLOAD

value The threshold value (*threshold*) or time value (*timeval*) in milliseconds. Valid values are 0 - 600000. You must name workloads for threshold sets 2 and 3 before setting any values for these thresholds.

The following guideline applies to setting threshold values:

- The RESETTIME value must be less than 80% of the MAXTIME value.
- The MAXTIME value must be less than the DISCRETETIME value.

DISCRETETIME=*threshold*
The discrete level of latency in milliseconds. This threshold defines the discrete threshold at which you want an event to notify you about the latency of a subscription. You can define this value to alert you to take a particular action when the event occurs.

The value of DISCRETETIME controls when the Classic data server sends the EIF event AA_replication_discrete_latency_exceeded.

- The event is sent when the apply latency of an individual unit of work for a subscription exceeds the DISCRETETIME threshold
- The event is suppressed to one per mean time interval to prevent overloading the event server that receives this event.

The DISCRETETIME threshold is independent of other threshold values. You must configure this threshold to a value higher than the value of the MAXTIME threshold time.

MAXTIME=*threshold*
The maximum level of latency in milliseconds. The value of MAXTIME controls when the Classic data server sends the EIF event AA_replication_max_latency_exceeded. The minimum non-zero configurable value of MAXTIME is 250 milliseconds.

- The event is sent when the averaged or non-averaged apply latency for a subscription exceeds the MAXTIME threshold.
- The event is not sent if MAXTIME <= RESETTIME .

RESETTIME=*threshold*
The reset level of latency in milliseconds. This threshold defines a reset point after a maximum threshold is reached. The RESETTIME threshold must be less than 80% of the MAXTIME threshold. The minimum non-zero configurable value of RESETTIME is 100 milliseconds.

The value of RESETTIME controls when the Classic data server sends the EIF event AA_replication_reset_latency_met.

- The event is sent when the averaged or non-averaged apply latency for a subscription is less than the RESETTIME threshold after the CONSTRAINEDTIME, CRITICALTIME, or MAXTIME threshold is exceeded.

HEARTBEAT=*timeval*
The latency heartbeat time in milliseconds. The heartbeat defines the heartbeat interval at which a heartbeat event reports the average latency and subscription state for a subscription.

You can only define this threshold time value in the THRESHOLDS1 threshold set. The THRESHOLDS2 and THRESHOLDS3 threshold sets

use the same value when their MAXTIME or RESETTIME threshold value are not zero. The minimum non-zero configurable value of HEARTBEAT is 10000 milliseconds (10 seconds).

The value of HEARTBEAT controls when the Classic data server sends the EIF event AA_replication_averaged_latency_heartbeat.

- If the value of HEARTBEAT is greater than 0 and the subscription state is REPLICATION CONTINUOUS, this event is issued on the specified interval.
- If the value of HEARTBEAT is greater than 0, that value is used as the heartbeat interval for all threshold sets.

MEANTIME=*timeval*

The latency mean time in milliseconds. The minimum non-zero value of MEANTIME is 3000 milliseconds (3 seconds).

You can only define this threshold time value in the THRESHOLDS1 threshold set. The THRESHOLDS2 and THRESHOLDS3 threshold sets use the same value when their MAXTIME or RESETTIME threshold value are not zero.

The mean time defines the time period during which end-to-end apply latency values are collected and averaged. Based on that averaging, EIF events are emitted when the averaged apply latency value transitions to the specified MAXTIME and RESETTIME thresholds. Averaging occurs when the value of MEANTIME is greater than 0.

If the MEANTTIME value is set to 0, apply latencies are not averaged. Events will be emitted for each instance when an apply latency threshold is breached. If the value of MEANTIME is greater than 0, that value is used as the latency mean time for all threshold sets.

WORKLOAD=*workload-name*

The workload name for the threshold set.

Each threshold set must have a workload name.

- The workload name of the first threshold set is the same as the subscription name. This workload is named when any of its threshold values are greater than zero. The workload name will be cleared when all its threshold values are zeros.
- You must define the workload name for the second and third threshold sets in the format `THRESHOLDS`*`n`*`,WORKLOAD=`*`workload-name`*, where *n* designates the threshold set (2 or 3).

When the Classic data server participates in GDPS/AA, threshold set 1 corresponds with the active/standby workload, threshold set 2 corresponds with the first active/query workload, and threshold set 3 corresponds with the second active/query workload.

The following rules apply to threshold sets:

- The length of the names is from 1 to 63 characters.
- The first character must be alphabetic.
- Characters 2 through 63 can be any alpha-numeric character and can contain the underscore character.
- Imbedded blanks are not allowed.
- Workload names are saved in upper case.

The configuration of workload names is performed against both the Classic source server and the Classic target server.

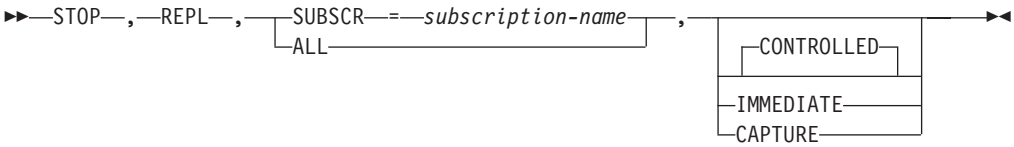
Use the DISPLAY,REPL,SUBSCR=*subscription_name*,THRESHOLDS MTO command to verify the state of the workload name.

STOP,REPL command

You can use the STOP,REPL command to stop replication activity for a single subscription or for all subscriptions.

Stopping replication

This command ends replication activity for the specified subscription.



Parameters

subscription name
The name of the subscription to stop. You can specify the subscription name as a quoted or an unquoted identifier.

ALL
Stops all subscriptions. The capture service examines the state of each subscription and processes the request if all subscriptions are in an appropriate state.

CONTROLLED
Controls how replication stops by allowing all transactions that are in process to finish before replication stops. Following a CONTROLLED stop, the subscription is placed in an inactive state. The capture service continues to read log data for the subscription in anticipation of a subsequent START command for the subscription. This is the default.

IMMEDIATE
Stops all transactions immediately, rolling back any that are not finished. Following an IMMEDIATE stop, the subscription is placed in the inactive state. The capture service continues to read log data for the subscription in anticipation of a subsequent START command for the subscription.

CAPTURE
Stops data capture and discards the capture cache for one or more subscriptions.

You can only stop data capture for a subscription that is in an inactive state. When you issue the STOP,REPL command with the CAPTURE keyword, the subscription remains in an inactive state and the capture service stops reading data for the subscription.

Examples

The CAPTURE keyword is useful when you change the definition of a subscription and the cached data will no longer match the new definition. The CAPTURE keyword is also useful in situations when you want to reduce the amount of storage that the capture cache uses.

- To stop data capture and discard any currently cached data for all inactive subscriptions:

```
STOP,REPL,ALL,CAPTURE
```

- To put subscription *sub1* into an inactive state and then stop data capture and discard any currently cached associated with the *sub1* subscription, issue the following sequence of commands:

  ```
  STOP,REPL,SUBSCR=sub1,IMMEDIATE
  STOP,REPL,SUBSCR=sub1,CAPTURE
  ```

- To put all eligible subscriptions into an inactive state and then stop data capture and discard any currently cached associated with the subscriptions, issue the following sequence of commands:

  ```
  STOP,REPL,ALL,IMMEDIATE
  STOP,REPL,ALL,CAPTURE
  ```

Classic data server administration commands

Use these commands to start and stop Classic data servers and services, list connected users, cancel user sessions, and view or print log messages.

Starting a data server

When you start a Classic data server, you start all of the services defined in the configuration file for the Classic data server.

About this task

You can perform either of the steps described in the following procedure to start a Classic data server. All services start if the value of the INITIALTHREADS configuration parameter is greater than 0.

Procedure

- Issue a console command to start the JCL procedure for the Classic data server:

  ```
  S procname
  ```

  where *procname* is the 1-8 character PROCLIB member name to be started. When you issue commands from the SDSF product, prefix all operator commands with the forward slash (/) character.
- Submit a batch job.

STOP command

Stopping a Classic data server stops all of the services that are running within it.

About this task

The purpose of the STOP,ALL command is to shutdown the data server. The data server stops after the services running in the data server complete their required processing.

If the shutdown process does not complete after issuing a STOP,ALL command, you can issue the STOP,ALL,IMMEDIATE command. For example, the shutdown process might not complete if a service encounters a problem and cannot complete its quiesce processing. In this case, you can issue the STOP,ALL,IMMEDIATE command to bypass service quiesce processing and stop the data server.

Procedure

- To stop a Classic data server, issue the following command in an MTO interface:

  ```
  F name,STOP,ALL
  ```

  name The name of the started task or batch job for the Classic data server.

- To stop a Classic data server immediately, issue the following command in an MTO interface:

  ```
  F name,STOP,ALL,IMMEDIATE
  ```

  name The name of the started task or batch job for the Classic data server.

START,SERVICE command

You can start an instance of a service that is defined in the configuration for the Classic data server.

About this task

You can use this command when you want to start a service without stopping and restarting the Classic data server. The service instance starts if the number of instances already active is less than the value of the MAXTHREADS configuration parameter.

Procedure

Issue the following command in an MTO interface, where *name_of_job* is the name of the started task for the Classic data server:

```
F name_of_job,START,SERVICE=name_of_service
```

STOP,SERVICE command

You can stop an instance of a non-critical service that is defined in the configuration for the Classic data server.

About this task

Important: You should not issue the STOP,SERVICE command regularly. When a data server is configured, you typically do not need to start and stop the services that run in the data server individually.

The STOP command cancels any user activity in a service and disconnects all active users from that service.

Restriction: You cannot stop a critical service. The following services are critical to the operations of a Classic data server:

- Administration service
- Apply service
- Capture service
- IMS log reader service
- Logger service

If you attempt to stop a critical service by issuing a STOP,SERVICE command or a STOP,TASKID command, a warning message is issued.

Procedure

- To stop a service by means of its task ID, issue this command:

  ```
  F server_name,STOP,TASKID=task_ID
  ```

  server_name
  The name of the task or batch job started by the Classic data server.
- To stop a service by means of its name, issue this command:

  ```
  F server_name,STOP,SERVICE=name_of_service
  ```

server_name
The name of the task or batch job started by the Classic data server.

DISPLAY,ALL command

The **DISPLAY,ALL** command outputs a formatted list of the current usage information about a data server.

Procedure

To display current usage information about services, users, configurations, and the memory pool, issue the DISPLAY,ALL command:

```
F name,DISPLAY,ALL
```

name The name of the task or batch job started by the data server.

DISPLAY,MEMORY command

The **DISPLAY,MEMORY** command outputs a formatted list of usage information about data server memory.

Procedure

To display the current use of the memory pool in the data server, issue the DISPLAY,MEMORY command:

```
F name,DISPLAY,MEMORY
```

name The name of the task or batch job started by the data server.

The following information is displayed about overall data server memory usage:

TOTAL MEMORY
The total size in kilobytes of the message pool that was allocated.

USED The amount of memory that is currently being used out of the message pool. This value is expressed in kilobytes followed by the percentage of the current message pool that is being used.

MAX USED
The maximum amount of the message pool that was ever used. This value is expressed in kilobytes followed by the percentage of the message pool that was ever used.

DISPLAY,SERVICES command

The **DISPLAY,SERVICES** command outputs a formatted list of information about the services running in a data server.

Procedure

To display a list of all running services in the data server, issue this command:

```
F name,DISPLAY,SERVICES
```

name The name of the task or batch job started by the data server.

When information is requested about the services that are active within a data server, a WTO display message is generated for each service that is active. For each service, the following information is displayed:

SERVICE
Service name.

TASKID
TCB address (in decimal notation) of the service instance that is displayed.

TASKNAME
Same as TYPE.

STATUS
One of the values that is displayed in the table below.

USER The user ID that is currently being serviced. Generally, this value is blank.

The following table lists the most common statuses:

Table 31. States and descriptions

Status	Description
QUIESCE	Unused.
READY	Idle and waiting for requests.
RECEIVING	Receiving a request.
RESPONDING	Sending a response.
STOP	Processing a STOP,ALL request.

Classic data server configuration commands

After a Classic data server is created and running as a result of the installation customization process, you can modify the configuration for the Classic data server as needed by using a set of operator commands.

You use configuration commands for the following actions:

- Updating and displaying configuration data for a Classic data server.
  The commands include ADD, SET, DELETE, and DISPLAY.
- Importing and exporting configuration data for a Classic data server.
  The commands include EXPORT and IMPORT.

You can issue the configuration-related commands by using the master terminal operator (MTO) interface. You can also use the Classic Data Architect to update your configuration for a Classic data server. With both the Classic Data Architect and the MTO interface, you make configuration updates against a running Classic data server.

The configuration migration and maintenance utility, CACCFGUT, also supports the EXPORT and IMPORT commands and provides a REPORT command. You can use the CACCFGUT utility to issue the commands offline, when the Classic data server is not running. For example, the utility supports the EXPORT and IMPORT commands that enable you to restore a configuration environment to a previous point in time.

Commands for updating and displaying configurations for Classic data servers

You use the ADD, SET, DELETE, and DISPLAY configuration commands to update configuration information for services, service lists, and global configuration parameters.

When you add a service with the ADD command, you add the service to the configuration for the Classic data server. Adding the service does not automatically start the service. A service starts automatically during the next startup of the

Classic data server if the value of the INITIALTHREADS configuration parameter is set to a value of 1 or greater. Otherwise, if you do not restart the Classic data server, you must issue a START,SERVICE command to start the service.

After a service is added, you cannot update the name of the service. To change a service name, you must delete the existing service and then add a new service with the new service name.

Basic validation occurs when you modify configuration parameters with the SET command. These validations are limited to general parameter data type and numeric range checks. The individual services validate specific configuration parameters to verify parameter content and relationships during service startup.

In addition to services configurations, you can add and modify service lists. A service list is a type of parameter that represents list values.

You use the ADD, SET, and DELETE commands to maintain these parameters.

To update a service list entry, you must delete the existing entry and then add a new entry.

ADD, DELETE, and DISPLAY service list commands:

You can use the ADD and DELETE commands to add and remove service list entries. You can use the DISPLAY command to display service lists.

Description

A service list is a type of parameter that represents list values. A single service list can contain an unlimited number of entries.

A service list is always associated with a specific service.

- The EIFEVENTSERVERS parameter associated with the logger service defines the destinations for Event Integration Facility (EIF) events. Changes made to this service list are effective immediately.
- The MSGLIST parameter associated with the logger service defines the destination for a message or suppresses a message. Changes made to this service list are effective immediately.
- The SSIDEXCLUDELIST parameter associated with the log reader service represents an exclusion list. Changes made to this service list are effective when the Classic data server or the log reader service is restarted.

You use the ADD and DELETE commands to maintain these parameters. To update a service list entry, you must delete the existing entry, and then add a new entry. You cannot use the SET command to update a service list entry.

Adding service list entries

You can add a service list entry with the following ADD command:

```
ADD,CONFIG,SERVICELIST=listname,SERVICE=servicename,VALUE=value
```

Example

```
ADD,CONFIG,SERVICELIST=SSIDEXCLUDELIST,SERVICE=LRSI,VALUE=ID1
```

See the information about the specific service list parameter for any operational issues involved with adding service list entries.

Adding service list entries for message destinations

You can add a message filter and a message destination to a message list with the following ADD command:

```
ADD,CONFIG,SERVICELIST=MSGLIST,SERVICE=servicename,VALUE=<msg>/<dest>
```

Examples

To suppress a message:

```
ADD,CONFIG,SERVICELIST=MSGLIST,SERVICE=LOG,VALUE=CECN0002I/SUPPRESS
```

To change the destination of a message:

```
ADD,CONFIG,SERVICELIST=MSGLIST,SERVICE=LOG,VALUE=CECN0002I/DIAGLOG
```

Adding service list entries for servers that receive EIF events

You can add a service list entry to define the server destination of Event Integration Facility (EIF) events.

Examples

To add an event server, issue one of the following ADD commands:

Example 1:

```
ADD,CONFIG,SERVICELIST=EIFEVENTSERVERS,SERVICE=LOG,
VALUE="SKT/event.server.host.name.com/6498"
```

where *event.server.host.name.com* and 6498 are the host name and the port number of the server to which you want the Classic data server to send EIF events.

Example 2:

```
ADD,CONFIG,SERVICELIST=EIFEVENTSERVERS,SERVICE=LOG,
VALUE="SKT/11.1.1.1/6498"
```

where the IP address 11.1.1.1 and the port number 6498 are the IP address and the port number of the server to which you want the Classic data server to send EIF events.

You must specify the value SKT as the protocol, followed by the IP address or host name, and the port number. You must delineate these values with a forward slash.

Deleting service list entries

You can delete entries from a service list with the following DELETE command:

```
DELETE,CONFIG,SERVICELIST=listname,SERVICE=servicename,VALUE=value
```

Example

```
DELETE,CONFIG,SERVICELIST=SSIDEXCLUDELIST,SERVICE=LRSI,VALUE=ID1
```

When you delete an entry from a service list, you remove the entry from the configuration. See the information about the specific service list parameter for any operational issues involved with deleting service list entries.

Deleting service list entries for message destinations

You can remove a message filter with the following DELETE command:

```
DELETE,CONFIG,SERVICELIST=MSGLIST,SERVICE=servicename,VALUE=<msg>/<dest>
```

Deleting a message from the MSGLIST service list resets the destination of the message to its default destination.

Deleting service list entries for servers that receive EIF events

You can delete a service list entry that identifies the server destination of Event Integration Facility (EIF) events.

Examples

To delete an event server, issue one of the following DELETE commands:

Example 1:

```
DELETE,CONFIG,SERVICELIST=EIFEVENTSERVERS,SERVICE=servicename,
VALUE="SKT/event.server.host.name.com/6498"
```

where *event.server.host.name.com* is the host name of the server to which you do not want the Classic data server to send EIF events.

Example 2:

```
DELETE,CONFIG,SERVICELIST=EIFEVENTSERVERS,SERVICE=servicename,
VALUE="SKT/11.1.1.1/6498"
```

where the IP address 11.1.1.1 is the IP address of the server to which you do not want the Classic data server to send EIF events.

The Classic data server disconnects associated client connections when the command is issued.

Displaying service list entries

Service list information is displayed when you display configuration information for the related service. For example, to display exclusion list information related to the log reader service, issue the following command:

```
DISPLAY,CONFIG,SERVICE=LRSI
```

Parameters

listname
The service list name. You can specify one of the following service list names:

- EIFEVENTSERVERS
- MSGLIST
- SSIDEXCLUDELIST

servicename
The service name associated with the service list name.

value
List value.

For EIFEVENTSERVERS, value is a URL string that defines the server destination to which the Classic data server sends EIF events. This string is in the format *protocol/ip_address/port_number*.

For MSGLIST, value is a string that defines the message ID and the destination. The string is in the format *<msg>/<dest>*, where:

- *msg* is the message ID. The message number must begin with prefix CEC and contain nine characters.
- *dest* is one of the following destinations:
  - CONSOLE: z/OS console
  - DIAGLOG: Diagnostic trace log
  - EVENT: Event log
  - SUPPRESS: No destination

For SSIDEXCLUDELIST, value is the name of the IMS subsystem ID.

ADD configuration command:

You can use the ADD command to add new service definitions and service lists to a configuration.

Adding services

Adding a new service typically occurs during the installation customization process. Otherwise, you can add a service by using the ADD configuration command.

When you add a service, you specify the service name and select the service class for the new service. The service class is predefined.

The following table lists the service classes and the service type that each service class defines.

Table 32. Service classes.

Service class	Service type
APLY	Apply service
CAP	Capture service
CNTL	Region controller service
DRA	IBM IMS database resource adapter (DRA) access service
GLOB	Global service
INIT	Client connection handler service
LOG	Logger service
LRSI	IMS log reader service
MAA	Monitoring service
OPER	Remote operator command service
PAA	Administration service

You can add services with the following ADD command:

```
ADD,CONFIG,SERVICE=servicename,SERVICECLASS=serviceclass
```

Example

To add a new capture service named CS to the configuration, issue the following command:

```
ADD,CONFIG,SERVICE=CS,SERVICECLASS=CAP
```

In this example, the configuration parameters for the TESTV10 service are created with the default values for the LOG service class.

Parameters

serviceclass
The name of the service class that the service is associated with.

servicename
When adding a service, specify the name of the service to create. When adding a user-specific configuration, specify the name of the service to override. The maximum length allowed for the name is 64 bytes.

DELETE configuration command:

You can use the DELETE command to remove service definitions, service list entries, and user-specific definitions from a configuration.

Deleting services

To delete a service, you must specify the name of the service to delete. If a service is running, you must stop the service before you delete the service. When a service configuration is deleted, any associated user-specific configuration is also deleted.

You cannot delete the core default services — the controller service and the logger service. You also cannot delete the GLOBAL service name which represents global parameters.

You can remove a non-core service with the following DELETE command:

```
DELETE,CONFIG,SERVICE=servicename
```

Example

To delete the service TESTV10 from the configuration, issue the following command:

```
DELETE,CONFIG,SERVICE=TESTV10
```

Deleting user-specific configurations

To delete a user-specific configuration, you must stop the associated service. You specify the user logon ID that is used to connect to the Classic data server.

DELETE command format:

```
DELETE,CONFIG,USER=userid,SERVICE=servicename
```

Example

To delete the user configuration for user ID TESTUSER related to service CACLOG from the configuration, issue the following command:

```
DELETE,CONFIG,USER=TESTUSER,SERVICE=CACLOG
```

Parameters

servicename
When deleting a service, specify the name of the service to delete. When deleting a user-specific configuration, specify the name of the related service.

userid
The user ID of the user to delete.

DISPLAY configuration command:

You can use the DISPLAY command to display configuration information about services, service lists, and global configuration parameters.

Displaying all configuration information

You can display all configuration information for a Classic data server with the following DISPLAY command:

```
DISPLAY,CONFIG,ALL
```

During startup of the Classic data server, the contents of the configuration file are written to the first entry in the diagnostic log for the Classic data server.

Displaying service information

You can display configuration information about a specific service with the following DISPLAY command:

```
DISPLAY,CONFIG,SERVICE=servicename
```

Example

To display all configuration parameters in the service named CAPTURE, issue the following command:

```
DISPLAY,CONFIG,SERVICE=CAPTURE
```

Displaying user-specific configuration information

You can display user-specific configuration information about a specific user configuration, a specific user, or all users with the following DISPLAY commands.

- Display a user-specific configuration for the specified service:

  ```
  DISPLAY,CONFIG,USER=userid,SERVICE=servicename
  ```

  Example

  To display all parameters in user configuration for USER1 and service CACSAMP, issue the following command:

  ```
  DISPLAY,CONFIG,USER=USER1,SERVICE=CACSAMP
  ```

- Display all user-specific configurations across all services that contain the specified user ID:

  ```
  DISPLAY,CONFIG,USER=userid,SERVICE=ALL
  ```

  Example

  To display all parameters in all user configurations for USER1, issue the following command:

  ```
  DISPLAY,CONFIG,USER=USER1,SERVICE=ALL
  ```

- Display all user-specific configurations across all services:

  ```
  DISPLAY,CONFIG,USER=ALL
  ```

Parameters

servicename
The name of the service to display. To display global parameters, specify GLOBAL as the service name.

userid
The user ID of the user-specific configuration.

SET configuration command:

You can use the SET command to modify parameter values defined for an existing service, a user-specific configuration, and global parameters.

Modifying service information

You can use the following SET commands to modify parameters in a single service, modify all services within a given service class, and modify all services across all service classes.

You can also use the SET command to reset the values of a configuration parameter value to the parameter default value.

- For a specific service, use the following SET command:

  `SET,CONFIG,SERVICE=servicename,parm=value`

  Example

  To set the TRACELEVEL parameter in service CAPTURE to 8, issue the following command:

  `SET,CONFIG,SERVICE=CAPTURE,TRACELEVEL=8`

- To reset a parameter value to the default for specific service, use the following SET command:

  `SET,CONFIG,SERVICE=servicename,parm=DEFAULT`

  Example

  To set the TRACELEVEL parameter in service CAPTURE to the default value (4), issue the following command:

  `SET,CONFIG,SERVICE=CAPTURE,TRACELEVEL=DEFAULT`

- For services within a given service class, use the following SET command:

  `SET,CONFIG,SERVICECLASS=serviceclass,parm=value`

  Example

  To set the TRACELEVEL parameter in all monitoring services to 3, issue the following command:

  `SET,CONFIG,SERVICECLASS=MAA,TRACELEVEL=3`

- For all services, use the following SET command:

  `SET,CONFIG,SERVICE=ALL,parm=value`

  Example

  To set the value of the TRACELEVEL parameter in all services to 2, issue the following command:

  `SET,CONFIG,SERVICE=ALL,TRACELEVEL=2`

Modifying user-specific configuration information

You can set the value of a specific parameter in a user-specific configuration with the following SET command.

`SET,CONFIG,USER=userid,SERVICE=servicename,parm=value`

Example: Set the value of the TRACELEVEL parameter in the user-specific configuration for USER1 for TEST95 to 2.

```
SET,CONFIG,USER=USER1,SERVICE=TEST95,TRACELEVEL=2
```

You can also use the DEFAULT keyword on the SET command to return a parameter to its default value.

Parameters

parm
The name of the configuration parameter to modify.

serviceclass
The service class of the service to modify.

servicename
The name of the service that contains the configuration parameter to modify. For global parameters, specify GLOBAL as the service name. The maximum length allowed for the name is 64 bytes.

userid
The name of the user logon ID. The maximum length allowed for the user ID is 8 bytes.

value
The new parameter value.

Usage notes

The following rules apply to specifying lowercase and uppercase character values for the SET command:

- Character strings that contain embedded spaces or special characters must be enclosed in either single or double quotation marks.
  - For double quotation marks (“), the character string is set to uppercase.
  - For single quotation marks ('), values specified in lowercase are saved as lowercase.

Commands for importing and exporting configurations for a Classic data server

An export does not affect an existing configuration for a Classic data server. The EXPORT command creates a snapshot of the configuration in a command file that contains ADD and SET commands. The command file can be used as input to the IMPORT command.

You can use the IMPORT and EXPORT commands to perform the following functions:

- Back up and restore a current configuration environment. The EXPORT command creates a command file that consists of ADD and SET commands based on the current environment that can later be imported into a Classic data server with newly initialized configuration files to complete the restore process.

  Important: Frequently back up configuration files by copying them or by using the EXPORT command.
- Apply updates to the current configuration environment. You can use the IMPORT command to apply updates to the current configuration environment in multiple updates or single updates. You can use the EXPORT command to create a command file or create a command file manually.

- Save different versions of a configuration environment. You can use exported configuration output to create a clone of the existing configuration for a Classic data server. For example, the EXPORT command is useful in a test environment where you can rebuild the configuration required for a specific test scenario by importing the saved configuration. The EXPORT and IMPORT commands can also be an effective mechanism for cloning Classic data servers.

Example

You can use the EXPORT and IMPORT process to restore a specific configuration environment to a previous point in time. By using the EXPORT command, you can create a command file that is based on the configuration environment of a running Classic data server. You can then use this command file to update a different configuration file by using the IMPORT command.

For example, you can use the EXPORT command to generate a command file, and then IMPORT that command file on a server that is running with a newly initialized configuration file.

To build a new configuration environment that is identical to an existing configuration environment, export the source configuration to the desired target file. Then run the configuration migration and maintenance utility CACCFGUT to import the original source configuration to the new target environment.

You can also issue the IMPORT command while a Classic data server is running to update parameter settings in the current configuration environment.

EXPORT configuration command:

The EXPORT command is useful for multiple purposes, such as backing up configuration information and cloning a configuration for a Classic data server.

Description

You can use the EXPORT and IMPORT process to restore a specific configuration environment to a previous point in time. By using the EXPORT command, you can create a command file that is based on the configuration environment of a running Classic data server. You can then use this command file to update a different configuration file by using the IMPORT command.

The target of the EXPORT command is a PDS member or a sequential file. If the file or member that you specify in the EXPORT command already exists, it is rewritten. If the file or member does not exist, it is created.

If you predefine the EXPORT target:

- The minimum record length is 80 bytes.
- The format can be either fixed or variable length records.

The EXPORT command does not support GDGs.

The owner ID associated with the Classic data server job must have authorization with an external security manager (ESM), such as the Resource Access Control Facility (RACF), to create or access the EXPORT target data set (sequential file or PDS member).

When a PDS member is specified as the target of the EXPORT command, an attempt to run the EXPORT command fails if another user or job accesses the PDS member at the same time. To avoid this situation, you create a PDS member to use for the IMPORT and EXPORT process only.

Exporting configuration information

EXPORT command format:

```
EXPORT,CONFIG,FILENAME=DSN:dsname | DSN:dsname(member) | DDN:ddname
 | DDN:ddname(member)
```

Example

The following example shows sample file contents of an EXPORT file:

```
--SET,CONFIG,SERVICE=GLOBAL,MESSAGEPOOLSIZE=67108864;
--SET,CONFIG,SERVICE=GLOBAL,DATACONVERRACT=1;
--SET,CONFIG,SERVICE=GLOBAL,FETCHBUFSIZE=32000;
--SET,CONFIG,SERVICE=GLOBAL,DECODEBUFSIZE=8192;
--SET,CONFIG,SERVICE=GLOBAL,STATICCATALOGS=0;
--SET,CONFIG,SERVICE=GLOBAL,TASKPARM='';
--SET,CONFIG,SERVICE=GLOBAL,DATAVALIDATEACT=0;
--SET,CONFIG,SERVICE=GLOBAL,REPORTLOGCOUNT=0;
ADD,CONFIG,SERVICE=CNTL,SERVICECLASS=CNTL;
--SET,CONFIG,SERVICE=CNTL,INITIALTHREADS=1;
--SET,CONFIG,SERVICE=CNTL,MAXTHREADS=1;
--SET,CONFIG,SERVICE=CNTL,MAXUSERS=100;
--SET,CONFIG,SERVICE=CNTL,TRACELEVEL=4;
--SET,CONFIG,SERVICE=CNTL,RESPONSETIMEOUT=5M;
--SET,CONFIG,SERVICE=CNTL,IDLETIMEOUT=5M;
SET,CONFIG,SERVICE=CNTL,SEQUENCE=1;
```

In the example, the export operation produces a command file of all overridden parameters as active SET commands. SET commands are also generated for all parameters that use default values. These particular SET commands are generated as comments. The prefix "– –" in the first two columns of the command identifies comments.

Parameters

ddname
: The DD statement defined in the JCL that starts the Classic data server. The DD name points to the target of the EXPORT command.

dsname
: The name of the EXPORT data set.

Usage notes

The following rules apply to the format of command files:

- Commands must end with a semicolon.
- A comment begins with two dashes ("– –") in columns 1 and 2.
- A single command can span multiple lines. You can break a line after a comma or a space. You cannot break a keyword or value across multiple lines. For example:

  ```
  SET,CONFIG,SERVICE=CAPTURE_SERVICE_NAME/CRAR/QRAR/2048/8,TEMPFILESPACE=
  'HIPERSPACE,INIT=2048M,MAX=2048M,EXTEND=0M';
  ```

Example

This EXPORT command creates the file USER.TEST.FILE if the file does not already exist.

```
EXPORT,CONFIG,FILENAME=DSN:USER.TEST.FILE
```

IMPORT configuration command:

You can use the IMPORT command to apply multiple updates or single updates to the configuration file for a running Classic data server. You can also use the IMPORT command to perform recovery operations to restore configuration information.

Description

The updates that the IMPORT command processes must reside in an existing IBM z/OS PDS member or sequential file. You can use either of the following methods to build the input file:

- Manually create the file and populate it with a defined set of commands.

  When you manually create a command file, you can specify any valid format of the ADD, SET, or DELETE configuration commands.
- Issue the EXPORT command to generate an IMPORT command file.

All commands in the file are processed whether or not a single command fails. Any errors encountered during the IMPORT process are displayed on the operator console. For example, unknown commands or incorrect command syntax can cause errors. Attempting to add a service that already exists can also cause an error. A status message is displayed when the process is complete.

You can use the IMPORT command at any time to change configuration parameter settings for existing services on a running Classic data server.

Importing configuration command files

IMPORT command format:

```
IMPORT,CONFIG,FILENAME=DSN:dsname | DSN:dsname(member) | DDN:ddname
 | DDN:ddname(member)
```

The data that you import must be in the command file format as described for the EXPORT command.

Example:

The following sample shows the contents of a sample IMPORT command file:

```
ADD,CONFIG,SERVICE=IMSLRS,SERVICECLASS=LRSI;
SET,CONFIG,SERVICE=IMSLRS,TRACELEVEL=3;
ADD,CONFIG,SERVICELIST=SSIDEXCLUDELIST,SERVICE=LRSI,VALUE="IMS1";
ADD,CONFIG,SERVICELIST=SSIDEXCLUDELIST,SERVICE=LRSI,VALUE="IMS2";
```

When this command file is imported, these changes occur:

- A new service named IMSLRS is added
- The value of TRACELEVEL is set to 3 for the IMSLRS service
- The value IMS1 is added to the exclusion list SSIDEXCLUDELIST for service IMSLRS

- The value IMS2 is added to the exclusion list SSIDEXCLUDELIST for service IMSLRS

Parameters

ddname
The DD statement defined in the JCL that starts the Classic data server. The DD name points to the source IMPORT command file.

dsname
The name of the IMPORT data set.

Example

This IMPORT command imports the file USER.PDS.FILE.

```
IMPORT,CONFIG,FILENAME=DSN:USER.PDS.FILE(USER1)
```

IMS-specific commands

IMS-specific commands apply to IMS replication.

RELEASE,RECON command

You can use the RELEASE,RECON command to release the existing holds (enqueues) on the current RECON data set.

Description

The RELEASE,RECON command is useful for releasing enqueues when your IMS administrator reorganizes RECON data sets.

This command is associated with the IMS log reader service. When the log reader interface (LRI) completes command processing, it passes a return code to the IMS log reader service. The IMS log reader service then logs the result of the command.

Syntax

This command releases existing holds:

```
'RELEASE,RECON'
```

Syntax

This command releases existing holds:

```
data-server-name,RELEASE,RECON
```

Parameters

data-server name
The name of the Classic data server that hosts the IMS log reader service.

Utilities reference

To work with configurations from a previous version, monitor data server configurations, or for backup and recovery purposes, use the configuration migration and maintenance utility.

The configuration migration and maintenance utility

The configuration migration and maintenance utility, CACCFGUT, runs as an IBM z/OS batch job that manages configurations for your Classic data servers. You can use the utility for backup and recovery, monitoring, and maintenance.

Features of the configuration migration and maintenance utility

You can use the configuration migration and maintenance utility to manage the configurations of your Classic data servers. The utility can monitor, back up, and recover your configurations.

The utility supports the following configuration-related Master Terminal Operator (MTO) commands:

- EXPORT
- IMPORT,CONFIG

The utility also provides a REPORT command.

To run the configuration migration and maintenance utility, enter one or more of the configuration commands on the SYSIN DD statement.

Backups

You can use the utility to run the EXPORT command at any time to create backups of configuration files.

Recovery

You can use the EXPORT and IMPORT commands to restore a configuration environment to a previous point in time. By using the EXPORT command, you can create a command file that is based on the configuration environment of a running Classic data server. You can then use this command file to update a different configuration file by using the IMPORT command.

To restore the full configuration, run the utility offline against empty configuration data sets when the Classic data server is not running. To avoid conflicts with default services, run the utility against empty configuration data sets.

Monitoring

You can use the utility to run the REPORT command at any time to monitor the configuration of the Classic data server.

Viewing log messages with the log print utility (CACPRTLG)

With the log print utility (CACPRTLG), you can format and display messages that are written to a log. You can summarize the log messages or filter them. You can also format and print event messages.

About this task

Perform the steps in the following procedure to view log messages:

Procedure

1. Configure CACPRTLG. See “Parameters for configuring the log print utility (CACPRTLG)” on page 225.

2. Create filters for the output. See "Filters for modifying output from the log print utility (CACPRTLG)" on page 226.
3. Run CACPRTLG. There are two ways to run this utility:
   - Run CACPRTLG as a step in the same job used to run the Classic data server.
   - Run CACPRTLG as a separate job from the Classic data server job or started task.

Parameters for configuring the log print utility (CACPRTLG)

You supply values to the **PARM** parameter of the CACPRTLG EXEC statement to determine which information CACPRTLG displays and where CACPRTLG extracts the information from.

Specify the **PARM** parameter in the JCL for the Classic data server. See the sample JCL for CACPRTLG in the sample members for the Classic data servers found in *USERHLQ*.SCACSAMP, such as CECIMSSC or CECIMSTG.

See also the sample JCL for CACPRTLG in the sample member *USERHLQ*.SCACSAMP(CACPRTLS), which shows how to run CACPRTLG against a system logger stream and print a log stream separately.

Recommendation: Use log streams for the diagnostic log or the event log (CACLOG) so that you can print the log while the data server is running. By using log streams, you do not need to set the logger service parameter DISPLAYLOG=TRUE to see logged information and can avoid the processing overhead costs related to formatting and displaying the logs.

The following list shows the possible values for the PARM parameter.

SUMMARY=N|Y

N Displays all of the messages that are in the log if you configured the logger service to write to the CACLOG DD statement or system log streams.

Y Displays a report about the contents of the log if you configured the logger service to write to the CACLOG DD statement or system log streams.

STREAM=log_stream
The *log_stream* value must be a valid log stream that contains data that was written by the logger service. If you use the STREAM keyword, remove the CACLOG DD statement from the JCL for the log print utility.

PURGE
Marks for deletion all of the log messages that are in the log stream and that are older than the value of the STARTTIME filter criterion for the log print utility.

PURGEALL
Marks for deletion all of the log messages that are in the log stream.

EVENTS=eventlog_stream_name
The name of the event log stream that was specified for the EVENTLOG configuration parameter for the logger service.

LOCALE=locale
The message locale to use when translating the event messages. If you do not

specify the LOCALE parameter, the default value EN_US is used to translate event messages using the US English message catalog. Valid values:

EN_US
US English message catalog.

JA_JP Japanese message catalog.

KO_KR
Korean message catalog.

ZH_CH
Traditional Chinese message catalog.

ZH_TW
Simplified Chinese message catalog.

Filters for modifying output from the log print utility (CACPRTLG)

You can use SYSIN control cards to filter and display only a subset of the log messages. With these control cards, you can display messages for a specific time-frame, a specific task, a range of return codes, or any combination of the elements that are listed in the log summary report.

The format of the SYSIN filtering is exactly the same as the format of the summary report. So, you can run a summary report, find the criteria that would be relevant for you to filter on, then submit a SYSIN control card with those criteria. You can find sample JCL to run a summary report in member CACPRTLS in the SCACSAMP data set.

The following list presents the available filtering criteria. Although the criteria are presented in uppercase, you can specify them in mixed case because the log print utility will fold the characters into uppercase. All filter criteria must be followed by an equal sign and a value.

STARTTIME='*YYYY***/***MM***/***DD HH***:***MM***:***SS***:***thmi***'**
Specifies the beginning of the duration of time that you want log information from. When you request the log information for a particular Classic data server, you might find it helpful to review the JES output for the Classic data server job to obtain the start time.

- *t* is tenths of a second
- *h* is hundredths of a second
- *m* is milliseconds
- *i* is ten-thousandths of a second

STOPTIME='*YYYY***/***MM***/***DD HH***:***MM***:***SS***:***thmi***
Specifies the end of the duration of time that you want log information from.

- *t* is tenths of a second
- *h* is hundredths of a second
- *m* is milliseconds
- *i* is ten-thousandths of a second

MINRC
Specifies a numeric value that represents the lowest return code that you want to be reported.

FILTERS
Specifies tracing filters to use in the report. Use only in conjunction with IBM support.

EXFILTERS
Specifies tracing filters not to use in the report. Use only in conjunction with IBM support.

MAXRC
Specifies a numeric value that represents the highest return code that you want to be reported.

TASKS
Specifies a task number (service) to filter the log information by. Although this criterion is helpful if you are diagnosing a problem with a specific task, generally you should not use this criterion. If this criteria is used with multiple values, each separate line must start with the TASKS keyword, an equal sign, and the comma-delimited list of task numbers enclosed within parenthesis.

NODES
Specifies a specific node (address space or Classic data server) for which the log print utility should return information. This value is a comma-delimited list enclosed with parentheses. Each line of node filters must be preceded by the NODES keyword and an equal sign.

SPCRC
Specifies a list of specific return code values for which the log print utility should return log records. Use only in conjunction with IBM support.

Glossary

This glossary contains terms and definitions for Data Replication for IMS.

A B C D E F I J L M P Q R S T U

A

administration service

In Data Replication for IMS, a service that facilitates runtime operations. In addition to maintaining the runtime environment and metadata, an administration service can coordinate configuration, route z/OS operator commands, and access the system event log.

See service.

apply PSB

In Data Replication for IMS, a user-created program specification block (PSB) on a target server that contains PCBs that reference each target database. The PSB enables the apply service to write change data to target databases.

See also program communication block (PCB), program specification blocks (PSBs).

apply service

In Data Replication for IMS, a service on a target server that processes change data and writes the changes to target databases or file systems.

See service. See also capture service, change data, source server, target database, target server.

B

bookmark

In Data Replication for IMS, a set of values in a database on the target server that supports transactional data delivery by specifying a restart position for a subscription.

See also subscription.

C

call level interface (CLI)

An API for database access that provides a standard set of functions to process SQL statements, XQuery expressions, and related services at run time.

capture point

See log position.

capture service

In Data Replication for IMS, a service running in a source server that processes change data from a data source and then forwards the changes to an apply service on a target server.

See service. See also apply service, change data, data source, source server, and target server.

change capture

See change data capture.

change data

Modifications to records in a data source or asynchronous change messages that describe those modifications, which middleware software then transfers to replicas of the database or file system, or to applications for additional processing.

See also data source.

change data capture

The process of monitoring a data source for change data and then transporting the changes to a replica of the data source or to an application.

See also change data, data source.

CLI

See call level interface.

connection handler service

In Data Replication for IMS, a service that listens for connection requests from client applications. The connection handler then routes these requests to the appropriate service, such as a query processor or administration service. A connection handler can use different communication protocols, such as TCP/IP or z/OS cross-memory services.

See service.

D

data server

A server that provides services for the secure and efficient management of information.

See also service.

data source

In information management, a database system, file system, table, or directory that makes data available for replication or for transfer to applications for additional processing.

See also change data capture, target database.

diagnostic log

A z/OS log stream or data set that the logger service in a Classic data server uses to store informational, error, console, event, and trace messages.

See event log, job log, log, log stream.

E

event log

A z/OS log stream that the logger service in a Classic data server uses to store and retrieve console messages and event messages.

See diagnostic log, job log, log, log stream.

F

FIFO

See first-in first-out.

filtering

A method of selecting data or information for processing that employs user-specified criteria. Examples include filtering that a log reader service performs to select data for capture, and views that expose only those rows that pass the filtering criteria.

See also log reader service.

first-in first-out (FIFO)

A queuing technique in which the next item to be retrieved is the item that has been in the queue for the longest time.

See also queue.

I

Data Replication for IMS

Replication from a source IMS database to a target IMS database.

J

job log

A z/OS log related to a running server or utility.

See log, diagnostic log. event log.

journal control field

In change data capture, a field containing metadata information that log entries provide about a change to a data source. You can replicate the data in journal control fields and assign them to target columns. Examples of information in journal control fields include the type of update and the operating system user that performed the change.

See also change data, data source.

L

latency

In replication, the time that elapses between changing a data source and applying the change to a target.

See also data source, throughput.

load

In replication, a process that transfers the contents of a data source to a target database before replication starts. Some sources refer to this process as an *initial load*.

See also replication.

log

1. A file used to record changes made in a system.
2. A collection of records that sequentially describes the events that occur in a system.

See diagnostic log. event log, job log.

log position

In Data Replication for IMS, a user-defined property of a replication mapping that specifies a precise time for log reading to begin when replication resumes for that mapping.

See also log sequence number.

log reader service

In Data Replication for IMS, a service that processes current, past, or archived log records and then transfers selected change data to a capture service for further processing. A log reader service might also rebuild transactions from log records while processing changes that occurred in the past.

See service. See also log reading, filtering.

log reading

In Data Replication for IMS, a process that reads IMS logs to detect changes to the data, and then transmits selected changes to a target server for further processing.

See also change data capture, log reader service.

log record identifier

See log sequence number.

log sequence number (LSN)

An ordered identifier for a log record that enables programs to process logs accurately.

See also log position.

log stream

A sequence of data blocks that provides storage for IBM z/OS operations. z/OS identifies each log stream by its own unique identifier, or log stream name.

loopback protection

See recursion.

LSN

See log sequence number.

M

metadata

In Data Replication for IMS, an internal set of files that maintains the entities, attributes, and characteristics of your configuration.

monitoring service

In Data Replication for IMS, a service that connects with clients and manages reporting activities. Metrics reported by a monitoring service can include latency, statistics, status or state changes, and other information about ongoing replication.

See service.

P

parallel apply

A process on a target server that can improve performance by applying change data concurrently to different databases or to different records within the same database.

See also apply service, subscription.

PCB

See program communication block.

persistent subscription

In Data Replication for IMS, a subscription that begins to replicate automatically when it starts. A persistent subscription restarts without user intervention when you restart the engine after shutdown or after the subscription stops with a recoverable error. If you stop a persistent subscription manually, you must restart it manually.

program communication block (PCB)

In DL/I and IMS, a control block that contains pointers to IMS databases.

program specification block (PSB)

In DL/I and IMS, a control block that describes the databases and logical message destinations that are used by an application program. A PSB consists of one or more program communication blocks (PCBs).

See also apply PSB.

PSB

See program specification block.

Q

queue

A data structure for processing work in which the first element added to the queue is the first element processed.

See also first-in first-out.

R

recursion

An activity in which replication processing recaptures and resends change data in a continuous loop. Recursion is an undesirable consequence of replicating change data from one data source to another in both directions.

See change data.

recursion protection

A feature of replication software that prevents recursion.

replication

The process of maintaining a defined set of data in more than one location. Replication involves copying designated changes for one location (a source) to another (a target) and synchronizing the data in both locations.

See also data source.

replication mapping

In a subscription, a user-defined relationship for replication between a source replication object and a target replication object that specifies any required attributes.

See table mapping. See also replication object, subscription.

replication object

In Data Replication for IMS, a component of a subscription that represents a database or a data source, such as a file, table, or database description (DBD).

See also replication mapping.

S

service

1. A program that performs a primary function within a server or related software. A service accepts a message as input, processes the message, then routes the result to one or more output queues for additional processing.
2. In Data Replication for IMS, a z/OS task control block (TCB) that runs in a Classic data server to perform or manage specific functions. These functions might include administration, monitoring, communication, logging, change data capture, or apply processing.

See service class. See also data server.

service class

In Data Replication for IMS, a grouping of services that identifies the service type by using an identifier. Examples of service types include capture services, administration services, and monitoring services.

See service.

source data server

See source server.

source database

A database from which replication software transfers changes to a target database.

See also data source, target database.

source server

In Data Replication for IMS, a Classic data server that processes change data and forwards the data to a target server.

See also data server, source database, target server.

stored procedure

A block of procedural constructs and embedded SQL statements that is stored in a database and that can be called by name.

subscription

In Data Replication for IMS, a group of replication mappings that helps to manage replication, typically for a specific business application.

See also replication, replication mapping, replication object,.

sync point

A point during the processing of a transaction at which protected resources are consistent.

See also bookmark, transaction.

synchronizing

A process that produces a copy of a database or file system such that the structure and data match exactly.

See load.

system logger stream

See log stream.

T

table mapping

In a metadata catalog for a Classic data server, a user-defined projection of non-relational data objects into a relational representation.

See replication object.

target data server

See target server.

target database

A database to which replication software transfers change data from a source database.

See change data, source database.

target server

In Data Replication for IMS, a Classic data server that receives change data from a source server and manages the process of applying the data to a target database.

throughput

In replication, a measure of the rate at which the software retrieves, sends, and applies change data to a target database or file system.

See also latency.

transaction

1. In a non-relational database management system, a process in which all data modifications are either committed together as a unit or rolled back as a unit.
2. In a relational database management system, an atomic series of SQL statements that make up a logical unit of work. All of the data modifications made are either committed together as a unit or rolled back as a unit.

See unit of recovery, unit of work.

U

unit of recovery (UOR)

A sequence of operations within a unit of work between sync points.

See sync point, transaction.

unit of work (UOW)

A recoverable sequence of operations performed by an application between two points of consistency. A unit of work begins when a transaction starts or at a user-requested syncpoint. It ends either at a user-requested syncpoint or at the end of a transaction.

See sync point, transaction, unit of recovery.

UOR

See unit of recovery.

UOW

See unit of work.

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:
INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
J46A/G4
555 Bailey Avenue
San Jose, CA 95141-1003
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating

platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS," without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_. All rights reserved.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at “Copyright and trademark information” at www.ibm.com/legal/copytrade.shtml.

The following terms are trademarks or registered trademarks of other companies, and have been used at least once in this information:

- Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.
- Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
- Java™ and all Java-based trademarks are trademarks of Sun Microsystems, Inc., in the United States, other countries, or both.
- Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.
- UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, or service names may be trademarks or service marks of others.

Privacy policy considerations

IBM Software products, including software as a service solutions, (“Software Offerings”) may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering’s use of cookies is set forth below.

This Software Offering does not use cookies or other technologies to collect personally identifiable information.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies

and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, See IBM's Privacy Policy at http://www.ibm.com/privacy and IBM's Online Privacy Statement at http://www.ibm.com/privacy/details the section entitled "Cookies, Web Beacons and Other Technologies" and the "IBM Software Products and Software-as-a-Service Privacy Statement" at http://www.ibm.com/software/info/product-privacy.

Printed in USA