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

Guide and Reference

IBM

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

Guide and Reference

IBM

Note

Before using this information and the product that it supports, read the information in “Notices” on page 183.

This edition applies to InfoSphere Data Replication for VSAM for z/OS (program number 5655-Y18) and to all subsequent releases and modifications until otherwise indicated in new editions.

Chapter 1. Overview of Data Replication for VSAM

You can use Data Replication for VSAM to produce copies of your VSAM data sets and maintain current data in near-real time, typically on geographically-dispersed CICS regions.

Purpose

IBM® InfoSphere® Data Replication for VSAM 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 VSAM 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 VSAM failures compared to hardware-based solutions

Business intelligence

Data Replication for VSAM 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 data sets.

Data backup

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

How Data Replication for VSAM works

The illustration shows how Data Replication for VSAM maintains synchronized replicas that undergo rapid rates of change:

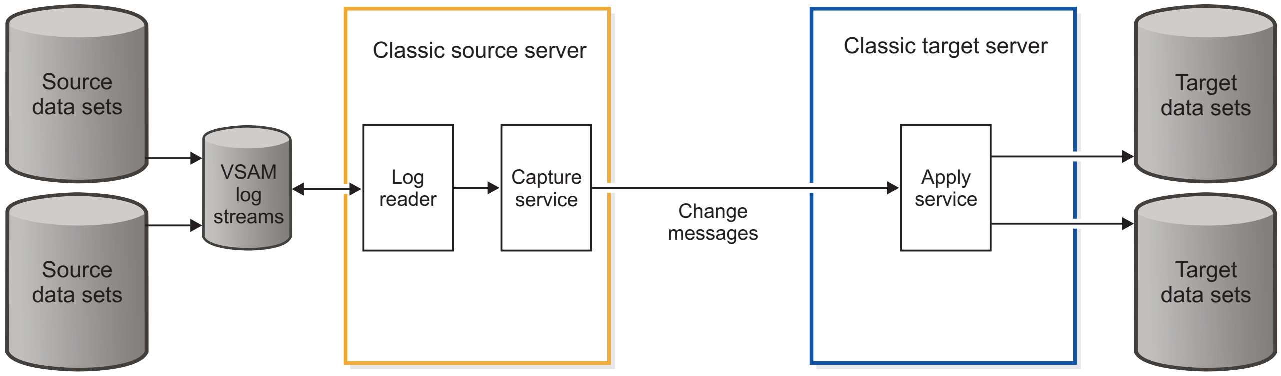

Figure 1. Classic data servers capture changes from VSAM log streams and send them immediately to target data sets

The synchronized replicas are maintained in CICS regions.

A Classic source server uses a log reader to read VSAM log streams and retrieve change records that are captured for source VSAM data sets. 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 data sets.

Data Replication for VSAM 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 VSAM with target data sets 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 VSAM can capture changes from one or more VSAM log streams. It supports one VSAM log stream per subscription.

Data Replication for VSAM has the following key components and capabilities:
- Subscriptions
- Transactional consistency
- Replicating historical changes
- Monitoring and reporting

Subscriptions

You organize the data sets that support a given application by mapping them to their target data sets within a *subscription*, a unique combination of source data sets, 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 VSAM can manage transaction processing across multiple logical partitions and data sets. 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 data sets to fall behind current processing and to go out of synchronization. Data Replication for VSAM 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 VSAM 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.

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

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

Migration considerations

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

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.

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 VSAM 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 VSAM

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

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

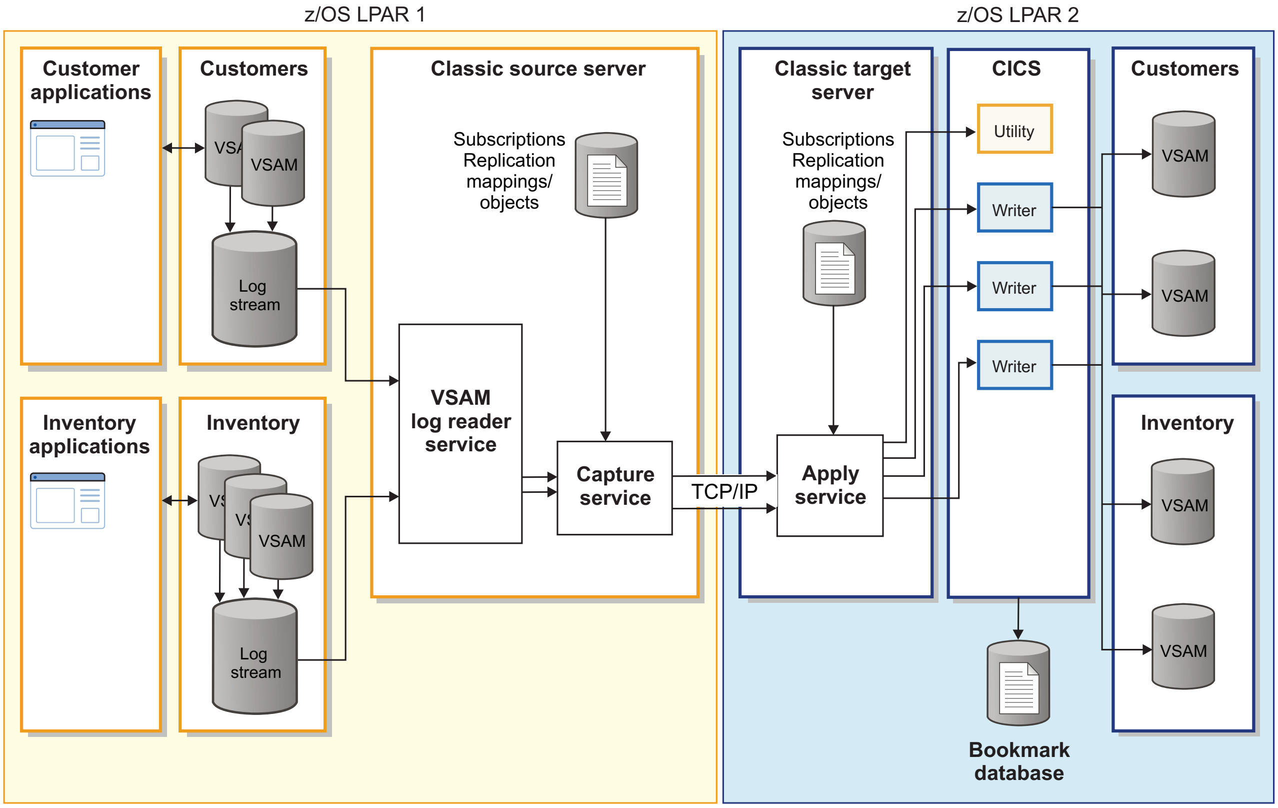

Figure 2. Architecture of Data Replication for VSAM

Geographically dispersed sites

In this example, the VSAM data sets 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 data sets 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 VSAM data sets and sends them to a *target server* that runs on a separate sysplex. The target server applies the VSAM file changes through a CICS® region at the target.

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

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

VSAM log reader service

The log reader service reads replication logs and sends change messages to the capture service that describe inserts, updates, deletes, and commits. A *replication log* is a system logger log stream. The log streams are shared with Forward Recovery.

Subscriptions

A *subscription* manages replication for data sets 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 data sets or to different records within the same data set.

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 VSAM data sets 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 VSAM data sets that you want to replicate by specifying a connection between a replication object and a subscription.

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 CICS region

The target data sets are defined to a CICS region. The CICS region is on the same z/OS logical partition (LPAR) as the target server. The target server and the source servers must be on separate z/OS LPARs that do not share the same Integrated Catalog Facility (ICF) catalog.

Recommendations:

- Devote a dedicated CICS region to apply processing so that you can optimize the region and improve performance for Data Replication for

VSAM. You can query the target data sets for business intelligence, but to maintain matching data between the source and target, do not actively update the target data sets. You should not acquire locks by a read for update.

- The files must be available to the CICS region. Files should be open before replication starts to prevent delays caused by opening the files.

Apply service

An apply service runs in the target server. This service schedules the VSAM source updates for each subscription. The updates are scheduled to a writer in CICS that applies the changes to the target CICS region.

For each replication object within a subscription, Data Replication for VSAM requires a VSAM data set, and a CICS file defined for that data set.

- The name and attributes of the target VSAM data set must match the name and attributes of the source data set.
- A CICS file name must be defined for each data set in the target CICS region with the file operations needed for replication.

The apply service typically queries CICS for the file names defined for a data set and chooses the first file name with the required definitions for the file operations. The apply service can also be configured to use a mapping file to define the CICS file names to use for the target VSAM data sets.

Bookmark database

The bookmark database is a VSAM data set. The target CICS region must contain a bookmark database for the apply service to store a current restart point for each subscription in the event of an outage.

Important: Data Replication for VSAM requires a separate CICS region per apply service. The CICS region must be in the same LPAR as the apply service.

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

1. Apply the changes to the target VSAM data sets using CICS file control requests.
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. Events that occur within the Utility and Writer programs in CICS are also written to the event log when a connection to the apply service exists.

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 VSAM 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 VSAM environment reads replication logs in source LPARs and preserves transactional consistency by writing change data to the target VSAM data sets in the correct order.

Apply processing in the target system 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 VSAM preserves transactional consistency between your source and target data sets by maintaining the sequence of changes to a 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 VSAM maintains transactional consistency by applying each source transaction as a transaction to your target VSAM data sets.

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.

Data Replication for VSAM also groups multiple changes into transactions for non-transactional changes. Changes to non-recoverable data sets and changes captured by CICS VSAM Recovery (CICS VR) are automatically grouped into transactions to provide more efficient use of system resources while minimizing latency.

For additional information about an exception to maintaining transactional consistency, see the topic *ESDS considerations*.

Transactions and latency: scenarios:

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

Data Replication for VSAM 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 VSAM 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 data sets*.

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 data set.
- The source transactions modify different data sets.

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

Transactions that modify data sets in different subscriptions

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

For this reason, Data Replication for VSAM cannot maintain consistency for transactions that modify data sets in multiple subscriptions. Design your subscriptions to include all data sets 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 VSAM applies changes to a target data set almost immediately after an application commits a unit of recovery (UOR) in the source data set. In some cases, however, the contents of a target data set 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 VSAM 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.

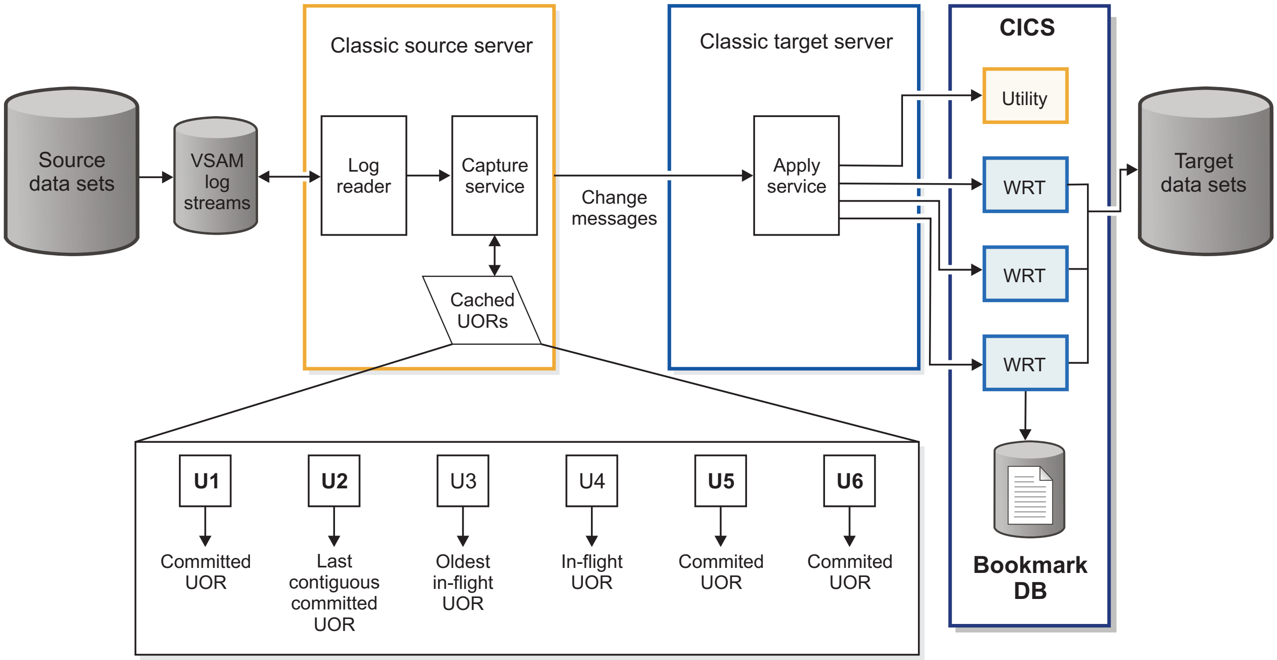

Figure 3. Historical changes available for replication processing after an outage

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 in the data set

  - Validation processing that detects mismatches between VSAM data sets
- An error or system outage occurs:
  - Link loss

    A link loss can be a lost TCP/IP connection between source and target servers or a lost z/OS IP socket connection to CICS.
  - A CICS system outage occurs and the subscription is stopped
  - Other internal errors

You stop and then subsequently restart a subscription

- The source or target data set 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.

Processing for other subscriptions continues. Use the following approach if you want to replicate most of the data sets in the subscription, or all the data sets but one:

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

Resource consumption (Data Replication for VSAM)

Data Replication for VSAM increases resource consumption on the source and target logical partitions (LPARs).

Examples of activities that increase resource consumption include logging and archiving, CPU processing, network traffic, and memory utilization. The following sections describe resource consumption issues in greater detail:

Log records for Data Replication for VSAM

Forward Recovery and Data Replication for VSAM share a log stream. This shared log stream is also referred to as the *replication log* in the product information for Data Replication for VSAM. The log is only common when the data set is defined for both forward recovery and replication logging. You do not need to use forward recovery to get a replication log.

Data Replication for VSAM relies upon replication log records. They provide more detailed information than a log record that you use for forward recovery.

You must specify the LOGREPLICATE attribute to activate replication logging for each VSAM data set. This attribute enables before images, after images, and commit records to be written to the log stream. These replication log records are generated in addition to any records generated by forward recovery logging.

The log reader retrieves changes from this log. Batch applications that use CICS VSAM Recovery (CICS VR) also write log records to the replication log, and you can share the log stream with an online CICS system.

You must manage the retention of the replication logs so that they exist for as long as you need them for processing historical changes after planned or unplanned outages. The longer retention period can increase storage utilization.

Processing time

CPU time increases measurably compared with processing the same workload without change capture. If the impact of adding Data Replication for VSAM is a concern, evaluate additional CPU usage in another environment before activating production. For example, you might want to evaluate the cost of replication logging in a test environment before altering production data sets to include LOGREPLICATE.

Network bandwidth

Data Replication for VSAM uses significant network bandwidth, and requires TCP/IP links between the source and target servers 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 VSAM must run in high-priority service classes to maintain low latency. Use a workload manager (WLM) service class 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 server 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 3 GB of memory. The memory required for the source server can grow as subscriptions increase because more subscription capture caches will be created.

Target servers use above-the-bar memory for transaction storage which is limited to 2 GB per subscription.

Target servers also use above-the-bar memory for dependency analysis. Each target server contains one area for dependency analysis.

Subscription processing

Subscriptions use two caches that you configure independently 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.

You can configure the caches by using the Subscription wizard in the Classic Data Architect (CDA).

Tasks overview

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

Installation tasks

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

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.

Database administrator (DBA)

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

The DBA role in configuring Data Replication for VSAM is to set up the data source as required. The DBA designs and defines the subscriptions. A DBA understands the data and how to organize the VSAM files into subscriptions.

Typically, the DBA loads target data sets before replication begins or continues. If the structure of a source data set changes in a way that affects replication, the DBA reinstalls the new structures at the target CICS region.

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 workload issues or subscription design.

Administration tasks

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 data sets, including administration and reorganization, such as starting and stopping replication and restructuring and cleanup operations.

Classic data servers

A Classic data server runs in its own address space.

A source server is a Classic data server that reads replication records generated by source VSAM data sets and sends them to a target server that runs on a separate z/OS image. The target server is a Classic data server that manages transaction dependencies and schedules file modifications that are ultimately completed by a remote writer transaction running in the target CICS region.

Classic data servers perform the following functions:

- Determine the type of data to access
- Maintain the environment

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 VSAM 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 VSAM source updates for each subscription and applies changes to the target data sets through the target CICS region.

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.

VSAM log reader service

The VSAM log reader service reads VSAM replication log records 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:
     - 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)
       - Added threads, such the log reader thread that reads logs for the replication log stream for each subscription.
     - The number of subscriptions, and related metadata:
       - Subscriptions
       - Replication mappings
       - Replication objects
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.

Synchronization of source and target VSAM data sets

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

Both the structure and data of your source and target data sets must match. To achieve this match when you start replication for the first time, you must both define the VSAM data sets and load the data. CICS file definitions must be defined on the target server with the access privileges needed for replication.

In most situations, you do not have to reinstall or reload a target data set after replication begins. If errors occur or replication stops, your replication environment can catch up to current processing by reading the logs in the source LPAR when replication resumes.

1. Defining or redefining the target VSAM data sets

To define or redefine VSAM data sets, you close the file in the target CICS region, delete the cluster, define the cluster, and copy (REPRO) the source contents from a source quiesce point into the target cluster. You then start or restart replication from the quiesce point.

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

- You set up Data Replication for VSAM for the first time.
- You add a new data set to a subscription.
- You change the key, record length, or organization of the source data set.

  To change the definition of a data set, see the procedure in the topic *Changing the definition of a replicated VSAM data set (schema change).*

  If you only reclaim unused space or physically reorganize data in the source data set, you do not need to redefine the target data set. However, to improve access to the target data set for Data Replication for VSAM you might want to perform the same maintenance on the target data set.

2. Loading or reloading the data

You load the target data set by using the IDCAMS REPRO command to copy the source VSAM data set to a non-VSAM data set (flat file). You then transfer the non-VSAM copy to the target and use the IDCAMS REPRO command to load the non-VSAM data set copy into the target VSAM data set.

Follow these steps to ensure that source and target data sets match:

1. Allocate the source data sets.
2. Define a log stream for the source data sets. See the topic "Defining replication log streams".
3. Allocate data sets on the target server.
4. (Optional) Define a log stream for the target data sets.

   Perform this step if data sets are defined with LOG(ALL) or the target is set up for LOGREPLICATE. Otherwise, you do not need to perform this step. See the topic "Defining replication log streams".
5. Copy data from the source data sets to the target data sets.
6. (Optional) Ensure that the source and target data sets are equivalent. Source and target data sets must begin as exact replicas.
7. Define the bookmark data set. See the topic "Creating a bookmark database".
8. Set the log position. See the topic "Activating replication mappings".

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

- You encounter errors or inconsistencies in the target data set, such as missing records
- You restore a source data set to a prior version.
- A VSAM batch job abends and you do not run the CICS VR batch backout utility (DWWBACK). For detailed information, see *Recovering data sets after CICS VR batch job failures*.
- Mass updates occurred at the source that changed most or all records in the data set
- Replication has been inactive for a subscription, and reloading the data will take less time than replicating the historical changes.

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 VSAM

Installing Data Replication for VSAM 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 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 VSAM on the mainframe.	"Installing Data Replication for VSAM on the mainframe" on page 24
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 24
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 25
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 44

System requirements

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

Prerequisites

Replication from VSAM data sources

- IBM z/OS Version 1.13 or later
- CICS Transaction Server for z/OS (CICS TS) Version 5.1
- CICS VR (CICS VSAM Recovery) Version 5.1 is required to replicate changes made by batch jobs outside of CICS Transaction Server.
- APAR OA39035, PTF UA65303 on z/OS Version 1.13
- APAR PM86451, PTF UK97129 on CICS TS Version 5.1

- APAR PM95393 on CICS TS Version 5.1
- APAR PM95451 on CICS VR Version 5.1
- APAR PM95661 on CICS TS Version 5.1
- APAR PM96728 on CICS TS Version 5.1

Supported VSAM data set types

Data Replication for VSAM supports replication from online environments under the control of the Customer Information Control System (CICS) Transaction Server for z/OS. It also supports replication in non-CICS environments when the application that creates the changes is using batch logging through CICS VSAM Recovery (CICS VR).

Data Replication for VSAM supports the following types of VSAM data sets:

- Key-sequenced data sets (KSDS)
- Entry-sequenced data sets (ESDS)

  A logical delete exit must be active in source CICS TS and CICS VR environments and in the target CICS TS environment. For CICS TS, you must enable the XFCLDEL global user exit. For CICS VR, you must define the exit by using the DEFEXIT ESDSDELET command.
- Relative record data sets (RRDS)

  RRDS support is limited to fixed RRDS.

You can define VSAM data sets for all of the supported VSAM files types in basic and extended formats.

Other requirements

Consider these requirements when you implement Data Replication for VSAM with a VSAM data source:

- If you are capturing changes in a data sharing environment, you must define VSAM log streams in a coupling facility to hold the log records.

Restrictions

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

General restrictions

One type of data source per Classic data server

You cannot use a single Classic data server to replicate data from different types of data sources, for example IMS™ and VSAM, at the same time. To replicate both IMS and VSAM data from the same system, set up:

- One instance of a Classic data server to provide VSAM source processing for replication.
- Another instance of a Classic data server to provide IMS source processing for replication.

Limitations on combining source Classic data servers with target data servers

A Classic data server sourcing VSAM data can only replicate data to one or more Classic data servers targeting VSAM.

A Classic data server sourcing IMS data can replicate data to one or more Classic data servers targeting IMS or one or more IBM InfoSphere Data Replication servers targeting non-IMS databases, files, or message queues. However, it is recommended that any one instance of the Classic data server replicate to either IMS or to InfoSphere Data Replication servers.

Setup restrictions

You cannot set up the source and target servers to share the same Integrated Catalog Facility (ICF) catalog. You must ensure that the source server uses a different ICF catalog than the target server because for Data Replication for VSAM the source and target data set names are the same. Data Replication for VSAM must apply changes to a copy of the source data set rather than to the same physical resource.

Target servers require access to the VSAM catalog for the target data sets.

Subscription limitation

A maximum of 63 subscriptions can be actively replicating on a target server.

Restrictions for VSAM data sets

Unsupported types of data set organization

A source server cannot capture changes from the following types of data set organization for VSAM:

- Linear data sets (LDS)
- Variable-length relative-record data sets (VRRDS)

Unsupported features

You cannot capture changes from VSAM data sets under the following conditions:

- The files are under the control of Transactional VSAM Services (DFSMStvs)
- The files are not under CICS control and use Record Level Sharing (RLS). This restriction includes Batch RLS.

Data Replication for VSAM cannot replicate CICS SET FILE EMPTYREQ behavior or similar batch behavior when a data set is defined with the REUSE option and is opened with an ACB that specifies the RESET option. These data sets are normally replicated if these techniques are not used to reset the data set as empty.

Data Replication for VSAM will not detect a change in the definition of a VSAM data set that is being replicated. You need to follow the required steps to stop and restart replication to change the definition of a data set. See the topic *Changing the definition of a replicated VSAM data set (schema change)*.

Records created in a source data set by using the MASSINSERT option on the EXEC CICS WRITE command are replicated, but MASSINSERT is not used when applying the changes at the target server.

Only cross-region share options 1 and 2 are supported.

Installing Data Replication for VSAM on the mainframe

The IBM InfoSphere Data Replication for VSAM 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 VSAM 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.	Source	Obtaining ports for replication
Assign port numbers for communication for Classic data servers.	Target	
Obtain a port to communicate with a CICS region.	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

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.

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 VSAM are the source server or the target server. You can customize an installation environment for a source server and a target server on separate LPARs. 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.

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
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 (CECCUSPV) and the installation customization utility JCL (CECCUSV2). The SCACSAMP(CECCUSV1) JCL runs this utility. SCACSAMP(CECCUSV1) is the JCL that runs the user samples allocation utility. CECCUSV1 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(CECCUSPV) and generates the necessary JCL and configuration members in the *USERHLQ*.USERSAMP and *USERHLQ*.USERCONF partitioned data sets. The *USERHLQ*.USERSAMP(CECCUSV2) JCL runs this utility. *USERHLQ*.USERSAMP(CECCUSV2) 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(CECCUSPV).
USERHLQ.USERSAMP(CECVSSLS)	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(CECVTSLS)	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(CECVSCFG)	Generated JCL that allocates and initializes the configuration files for the source server.
USERHLQ.USERSAMP(CECVTCFG)	Generated JCL that allocates and initializes the configuration files for the target server.
USERHLQ.USERSAMP(CECVSSUB)	Generated JCL that allocates and initializes the VSAM data sets for the source server that store metadata for subscriptions and replication mappings.
USERHLQ.USERSAMP(CECVTSUB)	Generated JCL that allocates and initializes the VSAM data sets for the target server that store metadata for subscriptions and replication mappings.
USERHLQ.USERSAMP(CECVSMSC)	Generated JCL to start the source server.
USERHLQ.USERSAMP(CECVSMTG)	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(CECCUSPV) and the installation customization utility JCL *USERHLQ*.USERSAMP(CECCUSV2). All input parameters specified for the samples allocation utility are populated in the generated CECCUSPV and CECCUSV2 members.

You use the SCACSAMP(CECCUSV1) 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.V11R1M00
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 VSAM source server or target server environment.

VSAMR_SOURCE
Specify this value to install the components required for a VSAM source server.

VSAMR_TARGET
Specify this value to install the components required for a VSAM 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.V11R1M00
   CACMGTCL:
   CACSTGCL:
   CACUSHLQ:    USER.V11R1M00.VSAMSRC
   ISPFHLQ:     ISP
   ISPFLANG:    ENU
   SERVERROLE:  VSAMR_SOURCE

Member     Status
--------   --------
CECCUSV2   Processed successfully in DDN(USERSAMP)
CECCUSPV   Processed successfully in DDN(USERSAMP)


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


Return Status: 0
```

Figure 4. 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(CECCUSPV).
- 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(CECCUSV2) 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 VSAM source server or target server environment.

VSAMR_SOURCE

Specify this value to install the components required for a VSAM source server.

VSAMR_TARGET

Specify this value to install the components required for a VSAM 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 2013-09-20 06:40:23 by REXX370 3.48
Execution timestamp: 2013-09-20 06:40:23 MVS Product ID: z/OS 01.12.00 SMF ID: SY45 System ID:
------------------------------------------------------------------------------------------------

Effective Parameters:

 CACINHLQ:    CEC.V11R1M00
 CACUSHLQ:    USER.V11R1M00.VSAMSRC
 OVERWRITE:   No
 SERVERROLE:  VSAMR_SOURCE



Processing parameters file: USER.V11R1M00.VSAMSRC.USERSAMP  Member: CECCUSPV


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 VSAM for z/OS  Role: Common

Processing Members for Product: Data Replication for VSAM for z/OS  Role: VSAMR_SOURCE
Member     Status
--------   --------
CECVSCFG   Processed successfully in DDN(USERSAMP)
CECVSXFG   Processed successfully in DDN(USERCONF)
CECVSSLS   Processed successfully in DDN(USERSAMP)
CECVSSUB   Processed successfully in DDN(USERSAMP)
CECVSMSC   Processed successfully in DDN(USERSAMP)


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

Return Status: 0
```

Figure 5. 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(CACCUSPV), 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=VSAMR_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=VSAMR_TARGET.	
Target server parameters	Parameters that apply to the target server
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 CECCUSV1 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
&VSAM	VSAM 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.

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(CECCUSPV)

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 CECCUSV1 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 CECCUSV1 job.

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

Parameters	Default value	Description
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 CECCUSV1 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 CECCUSV1 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 CECCUSV1 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 CECCUSV1 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(CECCUSPV)

Parameters	Default value	Description
Common server parameters for change data capture		
VSCPHLQD	USER.V11R1M00.VSAMSRC	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 VSCPCFGD, VSCPCFGX, and VSCPSUBS.
VSCPCFGD	&CPHLQ..CACCFGD	Name of the configuration data file for the source server. This value is prefixed with the VSCPHLQD high-level qualifier.
VSCPCFGX	&CPHLQ..CACCFGX	Name of the configuration index file for the source server. This value is prefixed with the VSCPHLQD high-level qualifier.
VSCPSHST	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.

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

Parameters	Default value	Description
VSCPPORT	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.
VSCPSUBS	&CPHLQ..SUB	Data file name for source server subscriptions. This value is prefixed with the VSCPHLQD high-level qualifier.
VSCPRMDS	&CPHLQ..RM	Data file name for source server replication mapping. This value is prefixed with the VSCPHLQD high-level qualifier.
VSCPCDPG	""	Host code page of the system for the source server. This value defines the global HOSTCODEPAGE configuration parameter for the source server.
VSCCNMICS	""	Network management interface (NMI) communication string for the source server. Sets the NMICOMMSTRING configuration parameter for the monitoring service.
Security parameters for the source server		
VSCPSAFX	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 .
For more information about the following parameter settings for the SAFEXIT, see Security.		
VSCPADMV	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.
VSCPADMR	""	ADMPROF resource profile parameter on the administration service for the SAFEXIT. This value is valid when the exit is enabled and VALIDATE=Y.
VSCPADMC	""	ADMCLASS security class parameter on the administration service for the SAFEXIT. This value is valid when the exit is enabled and VALIDATE=Y.
VSCPOPRV	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.
VSCPOPRR	""	OPRPROF resource profile parameter on the operator service for the SAFEXIT. This value is valid when the exit is enabled and VALIDATE=Y.
VSCPOPRC	""	OPRCLASS security class parameter on the operator service for the SAFEXIT. This value is valid when the exit is enabled and VALIDATE=Y.
VSCPMONV	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.
VSCPMONR	""	MONPROF resource profile parameter on the monitor service for the SAFEXIT. This value is valid when the exit is enabled and VALIDATE=Y.

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

Parameters	Default value	Description
VSCPMONC	""	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		
VSCPLGST	VSAMSRC.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. For more information, see Defining replication log streams.
VSCPLGDS	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 VSCPLGST is specified.
VSCPLGRT	7	Retention period, in days, to retain the log records before they for they are eligible to be deleted. This value is valid when VSCPLGST is specified.
VSCPLGSC	STG1	Storage class (STG_DATACLAS) for the log stream. This value is valid when VSCPLGST is specified.
VSCPLGSR	CCL1	Coupling facility structure name (STRUCTNAME). This value is valid when VSCPLGST is specified and the coupling facility is chosen (LGSTRDASD=”N”).
VSCPEVST	VSAMSRC.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. Defining replication log streams.
VSCPEVDS	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 VSCPEVST is specified.
VSCPEGRT	14	Retention period, in days, to retain the log records before they for they are eligible to be deleted. This value is valid when VSCPEVST is specified.
VSCPEVSC	STG1	Storage class (STG_DATACLAS) for the log stream. This value is valid when VSCPEVST is specified.
VSCPEVSR	CCP1	Name of the coupling facility structure (STRUCTNAME). This value is valid when VSCPEVST 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(CECCUSPV)

Parameters	Default value	Description
Target server parameters section		
VSAPHLQD	USER.V11R1M00.VSAMTRG	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 VSAPCFGD, VSAPCFGX, and VSAPSUBS.
VSAPCFGD	&APHLQ..CACCFGD	Name of the configuration data file for the target server. This value is prefixed with the VSAPHLQD high-level qualifier.
VSAPCFGX	&APHLQ..CACCFGX	Name of the configuration index file for the target server. This value is prefixed with the VSAPHLQD high-level qualifier.
VSAPHST	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.
VSAPPORT	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.
VSAPSUBS	&APHLQ..SUB	Data file name for target server subscriptions. This value is prefixed with the VSAPHLQD high-level qualifier.
VSAPRMDS	&APHLQ..RM	Data file name for target server replication mapping. This value is prefixed with the VSAPHLQD high-level qualifier.
VSAPLPRT	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.
VSAPCPRT	5003	Port specified when creating the z/OS IP CICS Socket Listener in the CICS system that the apply server uses. This value defines the CICSPORT configuration parameter of the apply service for the target server.
VSUSSPFX	" "	Name of an optional prefix to the Data Replication for VSAM directory /usr/lpp/vsamrep/v11r1m00 that contains the USS definitions for the product. This prefix is specified during SMP/E installation. The maximum length allowed for the prefix value is 50 characters.
VSAPCDPG	" "	Host code page of the system for the target server. This value defines the global HOSTCODEPAGE configuration parameter for the target server.
VSANMICS	" "	Network management interface (NMI) communication string for the target server. Sets the NMICOMMSTRING configuration parameter for the monitoring service.
VSCINHLQ	CICSTS*nn*.CICS	High-level qualifier (HLQ) assigned to the CICS libraries that are used to define the necessary resources to CICS. The value *nn* identifies the version of CICS Transaction Server (CICS TS). This HLQ is applied to the data sets in the STEPLIB sections of the generated JCL. This is the value substituted for the '&CICS..' symbolic parameter in the other installation parameters (such as VSAPSTPL).

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

Parameters	Default value	Description
VSCSTEPL	&CICS..SDFHLOAD	Library concatenation, for example STEPLIB concatenation. You can specify this keyword multiple times. The order of the multiple VSAPSTPL keywords defines the order in which the files are included in the concatenation for the generated JCL members.
VSCDFHCS	&CICS..DFHCSD	DFHCSD data set to use when defining the Classic federation programs and transactions to CICS.
VSAPCBKD	CECBKMK	CICS (FCT) file name that contains the bookmark information. The file name cannot exceed 8 characters in length. The actual VSAM file name associated with this file will be comprised of the target server HLQ specified in VSAPHLQD, followed by this CICS file name. Using the default values, the file name is USER.V11R1M00.VSAMTRG.CECBKMK. This value defines the CICSBOOKMARKDB configuration parameter of the apply service for the target server.
VSAPBKPL	CECBKPL	CICS LSR pool name associated with the bookmark file. The pool name cannot exceed 8 characters in length.
VSAPPLNM	9	Identifier of the CICS LSR pool associated with the LSR pool name defined by VSAPBKPL. This parameter must be an integer value in the range 1 – 255.
Security parameters for the target server		
VSAPSAFX	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.		
VSAPADMV	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.
VSAPADMR	" "	ADMPROF resource profile parameter on the administration service for the SAFEXIT. This value is valid when the exit is enabled and VALIDATE=Y.
VSAPADMC	" "	ADMCLASS security class parameter on the administration service for the SAFEXIT. This value is valid when the exit is enabled and VALIDATE=Y.
VSAPOPRV	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.
VSAPOPRR	" "	OPRPROF resource profile parameter on the operator service for the SAFEXIT. This value is valid when the exit is enabled and VALIDATE=Y.
VSAPOPRC	" "	OPRCLASS security class parameter on the operator service for the SAFEXIT. This value is valid when the exit is enabled and VALIDATE=Y.
VSAPMONV	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.

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

Parameters	Default value	Description
VSAPMONR	" "	MONPROF resource profile parameter on the monitor service for the SAFEXIT. This value is valid when the exit is enabled and VALIDATE=Y.
VSAPMONC	" "	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		
VSAPLGST	VSAMTRG.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. Defining replication log streams.
VSAPLGDS	Y	Identifies whether the z/OS log stream should use DASD or the coupling facility: • Y: DASD • N: Coupling facility
VSAPLGRT	7	Retention period, in days, to retain the log records before they for they are eligible to be deleted.
VSAPLGSC	STG1	Storage class (STG_DATACLAS) for the log stream.
VSAPLGSR	CAL1	Coupling facility structure name (STRUCTNAME). This value is valid when VSAPLGST is specified and the coupling facility is chosen (LGSTRDASD="N").
VSAPEVST	VSAMTRG.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. Defining replication log streams.
VSAPEVDS	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 VSAPEVST is specified.
VSAPEGRT	14	Retention period, in days, to retain the log records before they for they are eligible to be deleted. This value is valid when VSAPEVST is specified.
VSAPEVSC	STG1	Storage class (STG_DATACLAS) for the log stream. This value is valid when VSAPEVST is specified.
VSAPEVSR	CAE1	Name of the coupling facility structure (STRUCTNAME). This value is valid when VSAPEVST is specified and the coupling facility is chosen (LGSTRDASD="N").

Installing Classic data servers

The installation customization steps that you follow depend on the type of installation that you need to customize.

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

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

Installing VSAM source servers

You can follow the installation customization process to install and customize a VSAM 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 VSAM, SCACSAMP(CECCUSV1). 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 VSAM.

   CACUSHLQ=USER.V11R1M00
   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=VSAMR_SOURCE
: The value specified for the SERVERROLE keyword identifies the type of Classic data server to install and customize. Specify the value VSAMR_SOURCE to generate the JCL and components required for a source server environment for Data Replication for VSAM.

2. Submit SCACSAMP(CECCUSV1) 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 VSAM, CECCUSPV.
3. Edit the Data Replication for VSAM customization parameters file *USERHLQ*.USERSAMP(CECCUSPV) 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(CECCUSV2) 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(CECVSSLS) JCL to define the log stream and logs.
   c. Verify that all job steps result in a return code = 0.
6. Submit the generated *USERHLQ*.USERSAMP(CECVSCFG) JCL to allocate and initialize the source server configuration. This job populates the configuration with the service definitions needed for a source server environment. Verify that all job steps result in a return code = 0.
7. Submit the generated *USERHLQ*.USERSAMP(CECVSSUB) JCL to allocate the subscription and replication mapping data sets for the source server.
8. Submit the generated *USERHLQ*.USERSAMP(CECVSMSC) 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
   - Logger service
   - Operator service
   - Monitor service
   - Region controller service
   - VSAM log reader 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. See Setting up a source server for change capture to complete the setup required to prepare the source server environment.

Installing VSAM target servers

You can follow the installation customization process to install and customize an VSAM 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 VSAM, SCACSAMP(CECCUSV1). 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 VSAM.

   CACUSHLQ=USER.V11R1M00

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

2. Submit SCACSAMP(CECCUSV1) 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 VSAM, CECCUSPV.
3. Edit the Data Replication for VSAM customization parameters file *USERHLQ*.USERSAMP(CECCUSPV) 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(CECCUSV2) 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(CECVTSLS) JCL to define the log stream and logs.
   c. Verify that all job steps result in a return code = 0.
6. Submit the generated *USERHLQ*.USERSAMP(CECVTCFG) JCL to allocate and initialize the target server configuration. This job populates the configuration with the service definitions needed for a target server environment. Verify that all job steps result in a return code = 0.
7. Submit the generated *USERHLQ*.USERSAMP(CECVTSUB) JCL to allocate the subscription and replication mapping data sets for the target server along with the VSAM file that will contain the bookmark data.
8. Ensure that the Data Replication for VSAM CICS environment is set up and is currently running. Submit the generated *USERHLQ*.USERSAMP(CECCRDEF) JCL to define the CICS objects required for Data Replication for VSAM. The defined objects include programs, transactions, and files.
9. Submit the generated *USERHLQ*.USERSAMP(CECVSMTG) 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
- 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 Setting up a target server for apply processing to complete the setup required to prepare the target server environment.

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 45.

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.

Method	Description
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`.

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**.

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`.

Chapter 3. Configuring Data Replication for VSAM

Configuring Data Replication for VSAM requires setup tasks that include gathering information, preparing the VSAM data sets at the source and target, deploying and securing Classic data servers and the target CICS region, and creating subscriptions.

Gathering information and securing your environment

Data Replication for VSAM requires initial setup tasks, such as obtaining library authorizations and port assignments and securing your data servers.

Obtaining library authorizations for the APF

Data Replication for VSAM 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.

Obtaining ports for replication

Data Replication for VSAM 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 VSAM.

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. 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.

   c. A port for the z/OS IP connection between the apply server and the CICS region.

      You must define this port number for the CICSPORT configuration parameter for the APLY service on the target server. You must also use this port number when setting up the z/OS IP CICS socket (CICS listener) for the target CICS region.

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
Operator service	SERVAUTH	CEC.OPER	OPRCLASS	OPRPROF

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.

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 VSAM 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

VSCPCFGD="&CPHLQ..CACCFGD"
VSCPCFGX="&CPHLQ..CACCFGX"
	Source	Configuration data sets

VSCPSUBS="&CPHLQ..SUB"	Source	Subscription data sets
VSCPRMDS="&CPHLQ..RM"	Source	Replication mapping data sets

VSAPCFGD="&APHLQ..CACCFGD"
VSAPCFGX="&APHLQ..CACCFGX"
	Target	Configuration data sets

VSAPSUBS="&APHLQ..SUB"	Target	Subscription data sets
VSAPRMDS="&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 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.

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

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.

Preparing the Data Replication for VSAM environment

Preparing the Data Replication for VSAM environment requires setting up access to VSAM data sets and setup tasks for the source server, the target server, and the target CICS region.

Setting up access to VSAM data sets for replication

To replicate VSAM data sets, you must *augment* your source files for change data capture and define replication log streams to store the log records.

Different environments perform replication logging, depending on the type of file access:

CICS Transaction Server (Customer Information Control System)

The CICS address space performs replication logging for CICS access.

CICS VR (CICS VSAM Recovery)

The CICS VR address space performs replication logging for batch job access.

Augmenting VSAM data sets for change data capture

To perform change data capture from a VSAM data set, augment the file by using the IDCAMS utility to set the **LOGREPLICATE** parameter and set the value for the **LOGSTREAMID** parameter in the integrated catalog facility (ICF).

About this task

For Classic change data capture, use IDCAMS **DEFINE CLUSTER** or **ALTER** commands to define replication options in the ICF rather than in CICS resource definitions. For more information about the IDCAMS utility, see the IBM documentation for z/OS basic skills.

Set these parameters to activate replication logging for a VSAM data set and specify its related log stream.

LOGREPLICATE|NOLOGREPLICATE

A LOGREPLICATE value activates replication logging for the VSAM data set. A NOLOGREPLICATE value disables replication logging. Classic change data capture requires **LOGREPLICATE**.

LOGSTREAMID(*log-name*)

Identifies the name of the log stream that stores the log records that describe changes to the VSAM data set. The log streams that you specify on the LOGSTREAMID parameter are the same as the replication log streams that you define.

Note: Both replication logging and forward recovery logging use the **LOGSTREAMID** parameter, and both operations share the same log stream. For information about recovery options for CICS VSAM data sets, see the optional DEFINE CLUSTER parameters.

Procedure

1. Run the IDCAMS utility to specify **LOGREPLICATE** in the ICF catalog entry for the VSAM data set.
2. Specify the name of the log stream in the **LOGSTREAMID** parameter.

Example

The following example shows the job control language (JCL) for an IDCAMS ALTER command that sets the **LOGREPLICATE** parameter and adds the logstream CICSD.CICSD.DFJ02 to an existing VSAM data set USER1.SAMPLE.VSAMFILE.

```
//STEP1 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
  ALTER -
  USER1.SAMPLE.VSAMFILE -
  LOGSTREAMID(CICSD.CICSD.DFJ02) -
  LOGREPLICATE
/*
```

Defining replication log streams

You need to define a replication log stream for the VSAM data sets that you activate for replication logging.

About this task

A replication log is treated the same as a CICS general log.

When you define a replication log stream, you need to determine the type of log stream to define and consider how the options that you specify affect replication logging.

Procedure

Define a replication log stream by using the Administrative Data Utility (IXCMIAPU).
The replication log stream name that you specify for the NAME parameter must match the LOGSTREAMID parameter that you define when augmenting VSAM data sets for change data capture.
See the z/OS product documentation for detailed information about using this utility.

Example

The following example shows sample job control language (JCL) for defining a replication log stream.

```
//DEFLOG EXEC PGM=IXCMIAPU
//SYSPRINT DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//SYSIN DD *

DATA TYPE(LOGR) REPORT(NO)

DEFINE LOGSTREAM NAME(STREAM1.VSMLOG1)
 DASDONLY(YES)
 LS_SIZE(1024)
 STG_DATACLAS(STGCLS)
 STG_SIZE(512)
 RETPD(14) AUTODELETE(YES)
/*
```

Types of log streams

You must define log streams in data sharing and non-data sharing environments.

The type of log stream that you define depends on whether you access the log stream from one LPAR or from multiple LPARs.

- If you only access a log stream from one LPAR, you can use a DASD-only log stream or you can use a coupling facility log stream.
- If you access a log stream from more than one LPAR, you need a data sharing environment and a coupling facility log stream.

In a data sharing environment, you must define log streams in a coupling facility to enable logging operations to access the same log stream across all participating logical partitions (LPARs) in the sysplex. Multiple LPARs can generate inserts, updates, and deletes to the same VSAM data set. The target CICS region can write changes from all LPARs in the correct order and maintain transactional consistency.

For example: CICS updates all source files from one LPAR (LPAR1). The source server runs on a different LPAR (LPAR2). In this case, you need a coupling facility log stream.

A data sharing environment also requires that all the systems that access the source files share an ICF catalog.

For DASD-only log streams, all applications and the source server must run on the same image. If you have to apply maintenance, you might incur a replication outage.

For additional guidance about setting up log streams in a coupling facility, which type of log stream to define, and sizing the log streams, see the CICS product documentation about defining the logger environment.

Replication logging considerations

When you define a replication log stream, in addition to determining the type of log stream to define you need to consider factors that affect recovery and the volume of data that is generated.

Retention periods

When you define a replication log stream, you need to consider how long you want to keep data in the log.

The RETPD and AUTODELETE parameters that you set when you define a log stream determine how long data is retained and when data can be deleted. Manage the values that you set for these parameters to ensure that the required log replication data is available when replication is started.

You must retain log data long enough that you can recover change data if your replication configuration goes into recovery mode. For example, if you set the replication start time to 10 days ago and set the retention period to 5 days ago, the necessary data is not available and your replication configuration is in recovery.

Log volumes

When you activate replication logging, the types and volume of records written to the log is based on the recovery and replication options that you define. The number of updates and the number of files assigned to a given log stream also impact the volume of data that replication logging generates.

For more information about log record types and the effects of recovery and replication options, see the CICS product documentation.

Reference

The following table provides resources for additional information about defining replication log streams.

Table 11. Replication log stream considerations

Information resource	Reference
Replication logs	Replication logs
Administering restart and recovery, logging concepts	Administering restart and recovery

Table 11. Replication log stream considerations (continued)

Information resource	Reference
Defining the logger environment for CICS, log stream types and sizings	Defining the logger environment for CICS
Replication logging, recovery and replication options	Replication logging
Administrative Data Utility (IXCMIAPU) for defining replication log streams	Administrative Data Utility

ESDS considerations

Entry-sequenced data sets (ESDS) contain unique characteristics that impact replication processing.

The key issues that affect replication are ESDS inserts and deletes.

- ESDS records are assigned a relative byte address (RBA) when the record is written and that RBA becomes the record key. When a record that was written to the source ESDS is replicated, the RBA of record in the target data set must match the RBA of the record in the source data set.
- ESDS records cannot be physically deleted. However, ESDS records that an application flags as inactive are considered logically deleted and the records can be reused.

The methods that Data Replication for VSAM uses to maintain RBA consistency affect the following aspects of replication processing.

Serialization of inserts

Parallel apply processing supports units of recovery (UORs) with ESDS changes that include ESDS inserts. All UORs are eligible for parallel apply processing unless the subscription is enabled for serial apply processing.

When necessary, the system serializes the inserts for an individual ESDS.

Replication of inserts in recoverable ESDSs

Data Replication for VSAM maintains transactional consistency and replicates UORs that affect the same resource chronologically. However, when the resource is a recoverable ESDS, Data Replication for VSAM cannot maintain transactional consistency. ESDS inserts must be replicated in write order due to the method that CICS Transaction Server (CICS TS) uses to hold ESDS write locks.

To minimize lock duration, CICS TS does not hold ESDS write locks until the end of a transaction. This approach allows multiple UORs to insert into the same ESDS. The commit sequence of those UORs might not be the same as the insert sequence. For example:

- UOR A insert RBAs 0 and 160
- UOR B inserts RBAs 80 and 240

If UOR A commits first and an attempt is made to replicate UOR A to the target server, replication will fail. After completing the ESDS change to insert RBA 0, the attempt to insert RBA 160 will fail because the before image in the change record, with RBA 80, does not match the before image of the target data set record with

RBA 0. A similar condition occurs if UOR B commits first. When the RBAs are out of sequence an RBA pre-write check error occurs.

To maintain the RBA values in the target ESDS, Data Replication for VSAM must separate ESDS inserts from the UOR and replicate the ESDS inserts in write order. This replication method breaks transactional consistency because there is no coordination between the apply of the UOR that contains the insert and the UOR that contains other changes from the original source UOR.

Verification checks

When an ESDS insert record is processed, Data Replication for VSAM performs verification checks to ensure that the RBA in the target data set is accurate. A verification check occurs before and after the record is written.

The pre-write check ensures that the next available RBA matches the RBA in the log record before the record is written. The post-write check ensures that the RBA assigned after the record is written matches the log record.

If the RBA of the source and target data sets do not match, an insert error occurs. If the post-write check fails, this error means that the target data set is corrupted and you are in a recovery situation. The following error messages are issued to indicate an RBA mismatch on an insert:

- Message CECW0037E indicates that the expected RBA of the record in the target data set does not match the RBA of the record inserted in the source data set. In this case the target data set is not updated.
- Message CECW0038E indicates that the assigned RBA of the record written to the target data set does not match the RBA of the record written to the source data set. In this case the record was written to the target data set.

Backout processing

When a source unit of work (UOW) is backed out, any ESDS records that were inserted had an ESDS deletion exit invoked that updates the delete indicator in the record. To ensure that the target ESDS data set is in sync with the source data set, the insert followed by the update for the logical delete indicator that occurs during backout processing is written to the replication log. This allows Data Replication for VSAM to make the corresponding changes to the target data set. See the topic "Defining exits for ESDS records".

Defining exits for ESDS records

Data Replication for VSAM requires a logical delete exit to be active in source CICS Transaction Server (CICS TS) and CICS VSAM Recovery (CICS VR) environments and in the target CICS TS environment.

About this task

CICS TS provides the global user exit (GLUE) point XFCLDEL for ESDS deletions. CICS VR allows you to define the ESDS deletion exit by using the DEFEXIT ESDSDELET command. CICS invokes these exits when a write to a VSAM ESDS is backed out. The exit updates the record to flag a logical delete.

Procedure

- For CICS TS, enable the XFCLDEL global user exit. See the CICS product documentation about the sample global user exit program..

- For CICS VR, define the exit by using the DEFEXIT ESDSDELET command. See the CICS product documentation about the DEFEXIT command.

Preparing a target CICS region for replication

To set up your replication environment on the target site, prepare the CICS environment, install the z/OS IP CICS socket interface, define a bookmark database, and install CICS resource definitions.

Before you begin

Complete the following tasks before you begin these steps:

1. Install a CICS region on the target site.

   Ensure that the *CACINHLQ*.SCACLOAD library is defined in the STEPLIB and DFHRPL DD statements for the CICS region.

   Recommendation: Dedicate this CICS region to apply processing for Data Replication for VSAM.
2. Ensure that the definitions of all VSAM data sets that will be replicated match on the source and the target sites. Also ensure that each VSAM data set has a CICS file name that is defined in the target CICS region.

About this task

A bookmark database is a VSAM data set that stores information about the subscription restart positions so that the VSAM 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 VSAM sends only change data to the database replica, you must begin replication with matching copies of your source and target data sets. Before you start replication, synchronize the structure and contents of your source and target data sets by copying the data sets to the target system.

To prepare a target site for Data Replication for VSAM :

Procedure

1. Create and install a bookmark database and define it to the CICS region. See “Creating a bookmark database” on page 61.
2. Install and configure the z/OS IP CICS socket interface. See Setting up access to CICS for Data Replication for VSAM.
3. Define and install the CICS writer and utility tasks. The CICS writer and utility tasks reference programs that you install with the Data Replication for VSAM product. You must install Data Replication for VSAM before you perform this step. See Setting up CICS resource definitions for Data Replication for VSAM.
4. Set the CICS system initialization table (SIT) parameters MXT and EDSALIM to a sufficient value to ensure that the Classic data server can start the correct number of writer threads. Typically, the CICS Transaction Server for z/OS (CICS TS), version 5.1 default value are sufficient. See WRITERTHREADS.
5. Copy the VSAM data sets that you intend to replicate to the target system. You need to copy the data sets immediately before replication starts, during a quiesce period of operations at the source. To set the log position correctly, you need to specify this quiesce period when you activate the replication mapping.

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 VSAM uses this database to store restart positions.

About this task

A bookmark database is a VSAM data set that stores a restart position for each subscription. The restart position specifies where log reading resumes in the VSAM logs on the source server if an error occurs or if replication stops.

Use any unique and valid CICS file name for the bookmark database. This name must match the name specified for the **CICSBOOKMARKDB** configuration parameter for the apply service.

To create a bookmark database, ensure that you complete the installation customization process to set up a VSAM target server. When you set up a target server, the steps required to create a bookmark database are provided in the following JCL:

Procedure

- *USERHLQ*.USERSAMP(CECVTCFG): Allocates and initializes the target server configuration. This JCL defines the CICSBOOKMARKDB configuration parameter to specify the name of the CICS file.
- *USERHLQ*.USERSAMP(CECVTSUB): Allocates the subscription and replication mapping data sets for the target server along with the VSAM file that will contain the bookmark data. This JCL defines:
  - A VSAM cluster. The cluster is a key sequenced data set.
  - A local shared resources (LSR) pool to CICS. The LSR pool attributes influence the performance of access requests.
- *USERHLQ*.USERSAMP(CECCRDEF): Defines the CICS objects required for Data Replication for VSAM. This JCL defines the new CICS file to the target CICS region. The CICS file relates the name of the file in CICS with VSAM data set name.

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 12. 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
```

Starting a source server

When you start a source server, you start an address space, components, and services.

About this task

Starting the source server starts the address space that will connect to and read system logger log streams that contain changes captured from your source VSAM files. When you start a source server, you are also starting the components that allow transaction management and provide administration and monitoring activities. In addition, you start those services that are configured to start in the configuration file for the server.

Procedure

Start a source server by using one of the following methods:

- Issue a console command to start the JCL procedure for the source 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.

  Recommendation: Run Data Replication for VSAM as a started task in production environments, or in environments where performance is a concern, to ensure that it is given priority as a higher service class.

Setting up a target server for apply processing

Setting up a target server for apply processing requires preparation tasks to set up access to CICS, set up CICS resource definitions, configure the apply service, and start the target server.

Before you begin

The CICS region that controls your VSAM data sets on the target side must run in the same logical partition (LPAR) as your target server.

Ensure that the target server meets requirements for universal character set (UCS) support. See Meeting requirements for universal character set (UCS) support.

Setting up access to CICS for Data Replication for VSAM

The target CICS region uses a z/OS IP CICS socket (or *CICS listener*) to communicate with the apply service in the target server.

About this task

In the Data Replication for VSAM environment, there is one CICS target region per target server, with a CICS listener started.

The apply service that runs in the target server establishes multiple connections with CICS by using the z/OS IP CICS socket when the server starts.

- A connection to the CICS replication utility
- A connection to each instance of a CICS replication writer that is started

As changes arrive at the target server, the apply service schedules independent work for a started CICS replication writer by sending each change on a specific connection.

If you are setting up multiple replication environments, you must set up one z/OS IP CICS socket per target CICS region.

To ensure that your CICS listener is active, you can issue the following TSO command on the apply server: `DISPLAY TCPIP,`*`procname`*`,NETSTAT,CONN`. You can also issue this command from the system console on the target server to display the listeners.

For information about setting up a CICS listener, see the section about setting up and configuring CICS TCP/IP in the z/OS IP CICS Sockets Guide.

Procedure

1. Using the information in the z/OS IP CICS Sockets Guide, set up a z/OS IP CICS socket for the use of your target server.

2. Keep the port number of the z/OS IP CICS socket for reference.

   You will use the port number later, when you set up CICS resource definitions for Data Replication for VSAM and configure your target server.

Setting up CICS resource definitions for Data Replication for VSAM

Define and install CICS resource definitions to create the CICS transactions and programs that perform utility functions and write changes to your target VSAM data sets. The CICS transactions and programs are installed using a CICS bundle, CECVSAMR.

About this task

The communication process between your target server and CICS consists of these components:

- A target server, and the apply service that runs in the server
- A CICS listener
- A CICS utility transaction for replication
- A CICS replication utility
- CICS writer transactions for replication
- A CICS replication writer

The apply service establishes several IP connections to CICS transactions and programs. The first connection is to the CICS replication utility (CECCUTIL), which performs replication processing functions on behalf of the apply service. Connections to the CICS writer transactions (CECCWRTR) write VSAM source updates to their associated VSAM target data sets.

You run the sample member CECCRDEF during the installation customization process for the target server to define the CICS definitions required for Data Replication for VSAM.

Procedure

1. Run the sample member CECCRDEF in the USERSAMP data set. CECCRDEF defines the following programs and transactions:
   - The CICS bundle CECVSAMR that contains the definitions for the following programs and transactions:
     - The utility program CECCUTIL, and the utility transaction CFC3.
     - The writer program CECWRTR, and the writer transaction CFC1 (no logging).
     - The VHS program (DLL) CECVHS.
     - The work queue program (DLL) CECWQUE.
     - The BTREE program (DLL) CECBTREE.

     The value that you define for the MAXWRITERTHREADS configuration parameter for the apply service impacts the settings of the MXT (maximum task) parameter which is used in the target CICS system. In addition, since other non-writer tasks are necessary (the replication utility program and transactions needed for online use), ensure that you use the following formula to define the correct number of writer tasks:

     ```
     MAXWRITERTHREADS <= MXT - (number of non-writer tasks needed)
     ```
   - A transient data (TD) queue CFC1.

- An LSR pool for the bookmark database.
- The file definition for the bookmark database.

The resource definitions use IBM reserved names that should not be modified.

2. Ask your CICS administrator to install the VSAM programs and transaction definitions in the CICS target system. All resource definitions belong to GROUP(CECVSAM).

   To ensure that the definitions are installed when the CICS region is restarted, include the group in a list that is part of the GRPLIST SIT parameter for the target CICS region.

Mapping CICS files to VSAM data sets

The CICS mapping file CECCFMAP associates the name of a VSAM data set with a specific CICS file name that Data Replication for VSAM should use when applying change data.

The mapping file is required if you use remote files; it is optional for local files.

During the installation customization process, the following JCL creates, defines, and loads remote file mappings:

CECCCRFM
Creates the CECCFMAP data set that contains the association between any data sets that will be accessed remotely and the file name on the remote system. The data set name is matched to the data set name defined in the replication mapping for a subscription. You can also define local files as described below. Data Replication for VSAM ensures that the CICS file name that you specify in CECCFMAP is defined to CICS.

CECCDRFM
Defines the CECCFMAP data set to CICS.

CECCLRFM
Loads the CECCFMAP data set.

If you are replicating to remote CICS files, you must ensure that a file mapping is defined in your CICS file definitions. A mapping between a remote CICS file name and its associated data set name defined in the replication mapping for each subscription must exist in CECCFMAP.

If multiple local CICS files are defined to the same data set, you can define the specific local file name that Data Replication for VSAMs uses in CECCFMAP. When you add local files to CECCFMAP, you must set the value of the USECICSMAPFIRST configuration parameter to TRUE. The USECICSMAPFIRST configuration parameter informs the apply service that you want to locate local CICS file and VSAM data set associations in CECCFMAP, in addition to CICS remote files. Specifying that you want CECCFMAP searched first for both file types, rather than the File Control Table (FCT), can improve performance due to less resource consumption but provides less validation.

If USECICSMAPFIRST is set to FALSE, CECCFMAP is only searched for data sets that are not found in existing CICS file definitions.

Important: When you specify a data set name in CECCFMAP, the apply service confirms that the file that it points to exists in CICS and uses the file without further checking. If the file specified in CECCFMAP is not correct, an error does not occur until log records are sent to the target after subscription processing starts. Ensure that you specify file names correctly in CECCFMAP to avoid unexpected replication errors.

Recommendation: Do not use nonshared resource (NSR) buffers for target file definitions for Data Replication for VSAM. Using NSR degrades perfromance.

Restriction: Data Replication for VSAM must read the VSAM catalog to gather information for all VSAM data sets that are being replicated. For remote files, the VSAM catalog must be shared between all CICS File Owning Regions (FORs) and the target server. Based on z/OS requirements, shared files must exist in the same sysplex.

Starting a target server for Data Replication for VSAM

When you start your target server, you can verify the server configuration by checking error messages in the logs.

About this task

Starting the target server starts an address space that receives committed change data from one or more source servers. The change data is applied to target VSAM data sets through an associated CICS region.

Procedure

Start a Classic data server by using one of the following methods:

- Issue a console command to start the JCL procedure for the Classic 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.

  Recommendation: Run Data Replication for VSAM as a started task in production environments, or in environments where performance is a concern, to ensure that it is given priority as a higher service class.

What to do next

To verify that your target server is connecting with CICS properly, check for the following messages in the logs:

Job log

The following messages indicate successful connections to CICS:

- **CECW0004I** The target server successfully started CICS transaction CFC3 on port *xxxx*.

  This message represents the connection to CICS replication utility (CECCUTIL).

- **CECW0004I** The target server successfully started CICS transaction CFC1 on port *xxxx*.

  This message represents the connection to an instance of a CICS writer transaction (CECCWRTR). The message is repeated for each CECCWRTR instance that is started.

By default number, three writer transactions start when the target server is initialized. You can modify the number of writer transactions that start by specifying the WRITERTHREADS configuration parameter for the apply service.

In both messages, '*xxxx*' is the CICSPORT number.

The following message indicates that the target server failed to connect to CICS:

CECA0065E The target server was unable to connect to CICS by using port *port-number*. The specific return code is *specific-return-code* with reason codes *RC1* and *RC2*.

The apply service retries the connection request based upon the interval defined in CICSCONINTRVL configuration parameter.

Event log

After a connection is successfully established to the target server, the CICS writer program and the CICS utility program issue trace messages that are sent to the apply service. The apply service sends the trace messages to the event log or the diagnostic log.

CICS log

If an error occurs with the initial z/OS IP CICS connection between the apply service and CICS, trace messages will appear in the event log or the diagnostic log or in the CICS trace.

If the z/OS IP CICS connection is lost, any utility or writer messages are sent to the CICS transient data queue (TDQueue) output for the CICS region. The messages will appear in the MSGUSR DD statement

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 VSAM 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 13. 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

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

Table 13. Console messages and corresponding EIF events. (continued)

Console Message	**Corresponding Event Class**

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

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” on page 73.

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 14. 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

Table 14. Event classes (continued)

Event class name	Description	Attributes
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
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

Table 14. Event classes (continued)

Event class name	Description	Attributes
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
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

Table 14. Event classes (continued)

Event class name	Description	Attributes
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
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

Table 14. Event classes (continued)

Event class name	Description	Attributes
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 15. 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*

Table 15. Event attributes (continued)

Attribute name	Description	Value
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 VSAM, 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 VSAM, 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.	*VSAM*
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*
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*

Table 15. Event attributes (continued)

Attribute name	Description	Value
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*
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*

Table 15. Event attributes (continued)

Attribute name	Description	Value
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.
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 data sets 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 information and metadata about Data Replication for VSAM. You then select source data sets 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.

Optimal topologies for change data capture from VSAM data sets

Plan your deployment of VSAM data sets, log streams, and subscriptions to optimize both ease of setup and performance.

Performance and reliability

Performance might decrease as the volume of records that a log reader must process increases. These are some of the factors that can impede performance:

- Multiple subscriptions are assigned to the same log stream.
- A single subscription replicates many files or applications. Many files in the same log stream it might decrease the rate of writing changes from the source CICS regions.

- Changes exist in a log stream for files that are not in a subscription.

Tip: Examples of changes from files that are not in a subscription can include changes for replication logging or forward recovery logging.

- To minimize the number of records that the VSAM log reader must process, do not activate replication logging for a VSAM data set by setting **LOGREPLICATE** unless you define a subscription for the file.
- When you activate forward recovery logging for a file that is not being replicated, a log stream is reused for forward recovery logging that is also being used for replication logging. In this case, a VSAM log reader has to read and skip over all of the forward recovery log entries for this file since they are not important to the replication function.

Reliability is one of the major reasons for using subscriptions to replicate your data. If a subscription stops because it encounters an error or you stop it manually for maintenance, the remaining subscriptions continue to replicate independently.

Define subscriptions along application boundaries

As a minimum requirement, all source VSAM data sets within a single subscription must write their log records to the same log stream. When you perform any of the following actions, ensure that each underlying source file specifies the same value for **LOGSTREAMID**:

- Add a replication mapping to a subscription
- Start replication for a subscription
- Manually initiate Describe processing

You can use the Validate Replication Mappings function of the Classic Data Architect (CDA) to verify that all source VSAM data sets are defined to use the same log stream.

You can initiate Describe processing by issuing the following command:

```
F proc-name,START,REPL,SUBSCR=subscription-name,DESCRIBE
```

Check the event log for errors when you perform these actions to detect log stream mismatches among your replication mappings or VSAM data sets that are not augmented correctly. If you do not resolve these errors, you cannot start replication for the subscription.You can use the Validate Replication Mappings function of the CDA to verify that your VSAM data sets are augmented correctly.

To optimize performance and reliability, design a topology that sends all changes for a given application, such as Inventory or Orders, to a single log stream and subscription. This approach improves performance and enhances reliability. The first example in the next section illustrates the optimal configuration.

Example topologies

The illustration shows how to optimize the reliability and performance of change data capture by defining log streams and subscriptions along application boundaries. This approach defines a separate log stream for the files in each application group:

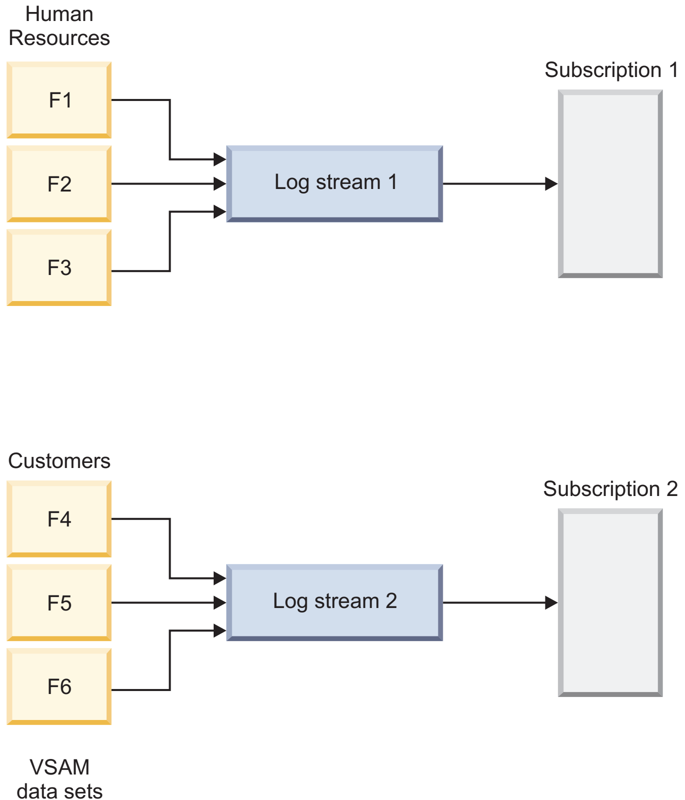

Figure 6. Optimal topology that devotes a dedicated log stream to each application

In a different topology that also satisfies the requirement for a single log stream per subscription, all files write their changes to a single log stream. While this approach is easier to set up, you incur a performance penalty because log readers for both subscriptions need to read all the records.

If many or all of your applications currently write to a single log stream, you can keep this configuration. However, you must also accept the performance penalty.

Recommendation: Avoid this environment by splitting a single log stream into multiple log streams before starting Data Replication for VSAM.

The illustration shows a topology where all files write their changes to a single log stream:

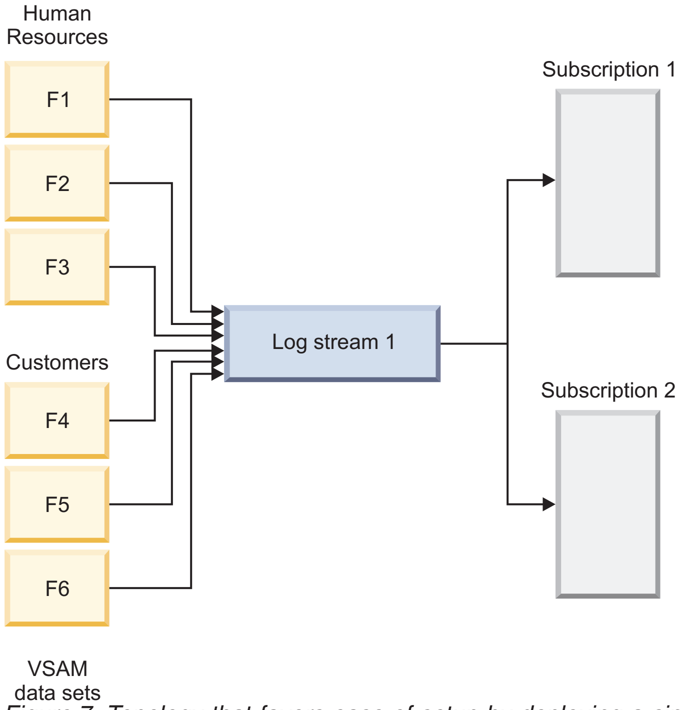

Figure 7. Topology that favors ease of setup by deploying a single log stream for all VSAM data sets

Tie-up records in CICS logs

Tie-up records must be available to Data Replication for VSAM prior to change data capture.

A VSAM data set is associated with a the name of a file definition in CICS and a DD name in CICS VR (CICS VSAM Recovery). When a VSAM data set is opened, CICS and CICS VR generate a *tie-up record* in the replication log. A tie-up record connects a data set name with the name that CICS or CICS VR uses in subsequent log records that describe changes.

When you set the bookmark for a subscription by using the SETLOGPOS parameter of the SET,REPL command, you need to ensure that tie-up records are available to Data Replication for VSAM prior to change data capture. The timestamp specified for the SETLOGPOS parameter must be a time when all data sets that are part of the subscription are closed in all CICS regions and in CICS VR.

This is also important when you activate replication mappings. When you activate a replication mapping, you need to ensure that a tie-up record is available to Data Replication for VSAM for that VSAM data set prior to change data capture. The timestamp specified in the activation request must be a time when the data set described by the replication mapping is closed in all CICS regions and in CICS VR.

Restriction: Do not set the CICS SIT parameter AKPFREQ to 0. This setting prevents Data Replication for VSAM from receiving periodic tie-up records. CICS does not recommended this setting. Without periodic tie-up records Data Replication for VSAM is not able to determine where log reading must begin when replication is restarted. Using AKPFREQ=0 can impact your ability to restart Data Replication for VSAM if required log records have expired.

Validation of the replication environment

Data Replication for VSAM validates key components to ensure 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 file that you have not yet augmented or a target file that you have not yet created.

The source server can also detect schema changes while replication is active.

Subscription validation

Data Replication for VSAM 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 VSAM data sets in the subscription, and identifies configuration problems or inconsistencies in subscription metadata that can prevent replication from starting:

- A source file with incorrect augmentation
- A target file that does not exist
- A mismatch between the attributes of source and target files
- A VSAM subscription that references two or more VSAM data sets that use different log stream IDs

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 VSAM data sets.

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 VSAM performs strict validations of the attributes of your source and target VSAM data sets for any replication mappings that require it. A connection to CICS must be established and the files must be defined in the file control table (FCT) with the required attributes.

Data Replication for VSAM 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

Validation of VSAM data sets

When you start replication, Data Replication for VSAM validates that certain attributes of the source and target files are consistent.

The validation process uses the following information to validate the attributes of VSAM data sets:

- The name of the file
- The type of entry, such as VSAM cluster
- The type of organization, such as KSDS, ESDS, or RRDS
- The maximum record length
- The offset of the key
- The length of the key
- The source and target file definitions must either be recoverable (LOG=ALL or LOG=UNDO) or non-recoverable(LOG=NONE)

Validating subscriptions:

To ensure that replication will start after you set up your replication environment, look for errors in the event log when you create or modify a subscription or validate replication mappings.

Before you begin

When you create or modify a subscription or start replication, ensure that the source and target servers are running and can communicate, so that they can connect and exchange metadata.

Procedure

1. Run the Create Subscription or Modify Subscription wizard in the Classic Data Architect to add or modify replication mappings. The Classic Data Architect validates the replication mappings automatically.
2. After validation processing is complete, check the event log for configuration errors or inconsistencies in the data and correct them.
3. After fixing any errors, right-click the subscription in the Classic Data Architect, choose the option **Validate replication mappings**, and then review the event log again to confirm that the subscription is valid for replication.

Results

You can now start replication for this subscription.

Changing the definition of a replicated VSAM data set (schema change):

To alter the definition of a VSAM data set that is being replicated, you must follow the required procedure.

About this task

Restriction: Data Replication for VSAM will not detect a change in the definition of a VSAM data set that is being replicated. You need to follow the required steps to stop replication before making definition changes and later restart replication for the data set after changes are completed on both the source and target system. Failure to follow the required steps can lead to unpredictable results.

The following steps describe the proper procedure to use when altering the definition of a VSAM data set that is being replicated by Data Replication for VSAM.

Procedure

1. Stop replication for the subscription.
2. If you want to allow all other data sets in the subscription to continue replicating while you change the definition of the data set, park the replication mapping of the data set you will modify the structure of and restart replication for the subscription.
3. Change the definition of the source and target data sets.
4. If you restarted replication in step 2, stop replication for the subscription again.
5. Copy the contents of the source data set to the target data set or otherwise synchronize their contents so the data sets match exactly.
6. Set the log position for the replication mapping of the VSAM data set that you changed. If the replication mapping was parked (in step 2) the replication mapping is also activated so it is replicated by the subscription. The time that you specify should be after replication was stopped for the subscription and the data sets were synchronized and before application changes were allowed for the source data set (that is, during a quiesce period for the data set).
7. Start replication for the subscription.

Apply processing

Parallel apply can improve the performance of your replication environment. *Adaptive apply* manages data conflicts that can arise between source and target databases.

Parallel apply:

Parallel apply processing improves the performance of your replication environment by applying units of recovery (UORs) concurrently to target data sets.

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 data set record are a limiting factor. In this case, Data Replication for VSAM places a higher priority on maintaining the precise order of changes as they occurred at the source data set.

For UORs that change separate data sets, delays or outages can prevent Data Replication for VSAM from applying changes in strict order.

An administrator can disable parallel apply manually by modifying the subscription to perform serial processing.

Restrictions

UORs with ESDS inserts

> You might not get the full performance benefits of parallel apply if UORs insert ESDS records into the same data set. The system performs dependency analysis that causes these UORs to be applied serially.

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.

Because Data Replication for VSAM balances transactional consistency with low latency, it places a higher priority on applying changes in order to the same record than it does on maintaining precise order for different data sets or records.

Changes to the same record of a VSAM data set

Data Replication for VSAM applies changes to the same record of a VSAM data set in the order that these changes were written to the replication log stream. For each subscription that participates in parallel apply, the target server maintains a collection of UORs, identifies dependencies, and assigns work to available writer tasks as earlier UORs complete apply processing.

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 VSAM 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 records or data sets

Data Replication for VSAM can also apply large numbers of UORs that update different records and data sets in approximately the same order that they occurred at the source site. The number of records and data sets that a UOR updates determines the extent to which the target server checks for dependencies.

To optimize memory and performance, the target server determines whether to track keys or affected data sets for each UOR.

- If a UOR updates more keys than a predefined threshold, dependency analysis will stop tracking keys to reduce memory and CPU costs and track affected data set names instead.
- If a UOR updates more data sets than the threshold, the UOR becomes dependent on all prior UORs for the subscription; all subsequent UORs for the subscription are dependent on this UOR.

Adaptive apply:

Adaptive apply processing manages data conflicts that can arise while writing changes to target VSAM data sets.

Adaptive apply is a form of replication in which certain situations that arise while writing changes to the target VSAM data set are not considered to be errors. When encountered, these situations are reported, the current change operation is skipped, and replication continues with the processing of the next change.

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 record with a matching key is found that already exists in the target VSAM data set.

- 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 after an immediate stop or failure. Units of recovery (UORs) might be reprocessed for a period of time after restart, when parallel apply was active prior to the immediate stop or failure. The amount of time depends on the amount of parallelism. The time period is the time between the oldest contiguous UOR that is completely processed and the most recent UOR that is processed as part of parallel apply.

When you reprocess a UOR, apply processing can encounter data inconsistencies that generate errors:

- Missing records
- Duplicate records
- Target data set records 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 processing encounters a duplicate record.
- Delete processing or update processing cannot locate the correct record on the target.
- Delete processing or update processing does not perform an update when a mismatch occurs between the before image and the existing image in the target record.

If adaptive apply processing encounters a status code for a condition other than one of those listed above, replication stops for the subscription.

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 data sets, Data Replication for VSAM 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:

U1	**U2**	U3	U4	**U5**	**U6**
				(update RA)	(delete RA)
		↑ **restart point**			

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 VSAM 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 VSAM processing. Adaptive apply discards other types of changes for similar reasons:

- Changes to missing records
- Changes where the before image data does not match the data in the target record

Conflict resolution in high availability environments:

If you replicate VSAM 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

Subscriptions at two different sites (site A and site B) must be identical. They must have the same number of replication mappings, the replication mappings must be in the same states, and the names and attributes of the VSAM data sets must match.

Recursion occurs when you recapture changes written by a target server. Data Replication for VSAM automatically prevents recursion. Changes made by the default CICS apply transaction CFC1 do not generate replication log records. This avoids recursion because the replication log for these data sets will not capture changes made by Data Replication for VSAM.

Designing subscriptions for VSAM data sets (guidelines)

Typically, you design subscriptions by adding replication mappings that represent a specific business application, such as Payroll or Customers. 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 is also an important consideration that helps you to optimize your subscription design.

If you have a large number of independently updated VSAM data sets, it can be difficult to identify the minimum number of subscriptions that you require and optimize the assignment of VSAM data sets 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 data sets that service a single application, such as Payroll or Customers. If multiple applications update the same VSAM data set, that data set needs an individual subscription.

      A source unit of recovery (UOR) that consists of multiple data sets in multiple subscriptions can result in a UOR that is divided across multiple subscriptions that apply the UOR independently (a *split UOR*).

      Example: A source UOR contains data sets A, B, and C. Data sets A and B are in subscription 1 and data set C is in subscription 2. All subscriptions are sent to the same target. As a result, two UORs are applied at the target. The first target UOR is for data sets A and B; the second target UOR is for data set C.

      In cases like this, the timing of when the two target UORs are applied is not guaranteed. You should design subscriptions to include all data sets if you require consistency of the target data.
   b. Test replication workloads that you plan to run in your production configuration. Use multiple subscriptions in production to avoid systemic failures.

      Avoid creating one large subscription that contains all the VSAM data sets. 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 VSAM data sets that are related to other applications.
      - A single failure brings down all replication processing in the environment.
   c. Improve performance with parallel apply.

      Use parallel apply processing for all subscriptions when you do not require UORs to be applied at the target in a strict order. When necessary, the system will serially apply UORs that cannot be applied in parallel. For example, ESDS inserts might not qualify for parallel apply processing. When necessary, the system orders UORs that have dependencies. Two data sets with ESDS inserts can be processed in parallel, but the inserts to a single ESDS data set must be processed serially. If UORs for a subscription must be applied in strict, chronological order, flag the subscription for serial apply.

3. If you have a high-volume environment that replicates a relatively small number of VSAM data sets, you can consider defining separate subscriptions for each VSAM data set.
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 (VSAM to VSAM replication)

Use Classic Data Architect to create a subscription and add replication mappings for the subscription.

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.
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 data source names (DSNs) to replicate using the Add Replication Mappings page. Type search criteria in the **Enter search** field to display a list of DSNs in the **Matched DSNs** list, Select the DSNs you want from this list and click the arrow icon to move it to the **Selected DSNs** list.
8. Validate your replication mappings by clicking **Validate all** on the Validate Replication Mappings page. This validates all DSNs.

   Alternatively, you can select specific DSNs from the list (by clicking and holding down the **Ctrl** key) to validate, and clicking **Validate Selected DSNs**.

   When you validate your DSNs, if there is a red X in the **Exists On Target** or **Matching Source/Target DSNs** columns, this indicates that you have not yet installed the target data set. If this is the case, install the source DSNs at the target site. (See Loading or installing target IMS databases.)
9. Compare source and target DSN information on the Compare Source/Target DSNs dialog.
10. Click **Close** on the Compare Source/Target DSNs dialog.
11. Click **Next**.
12. Review the summary of replication mapping details.
13. Click **Finish** to exit the wizard. You are now ready to start replication.

Chapter 4. Administering Data Replication for VSAM

Administering your replication environment involves ongoing tasks such as starting and stopping replication for subscriptions, loading or installing target VSAM data sets, 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.

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.
- 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.

Monitoring a Data Replication for VSAM 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:

  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:

  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:

  replication_object_name

  Specify the name of the replication object–the data set 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:

  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                                              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                                               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                                               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

The troubleshooting information helps you understand and resolve problems with Data Replication for VSAM, and contains information about how to use troubleshooting and support resources that are provided with your IBM products.

Recovering data sets after CICS VR batch job failures

Data Replication for VSAM captures and replicates CICS VR changes written during backout.

If a batch job fails at the source site that includes a data set being logged for replication, you need to consider the implications for the corresponding target data set. The source and target data sets must be point-in-time consistent.

The actions that you take to restart a failed source batch job might require you to perform actions against your target data sets. For example, if you decide to restore a source data set to a point-in-time prior to the batch job and re-run the batch job, you also need to copy the restored version of the data set to the target and set the log position for the replication mapping before you re-run the batch job.

Recovering data sets using DWWBACK

After a batch job fails that updates VSAM data sets defined with FRLOG(UNDO) or FRLOG(ALL), you can run the CICS VR batch backout program (DWWBACK) to undo the changes to the source data sets. When you use DWWBACK, Data Replication for VSAM captures and replicates those changes to maintain the corresponding target data sets.

Recommendation: Use DWWBACK whenever possible. Other recovery methods are likely to result in a replication outage for the data sets.

For detailed information about the DWWBACK program, see Starting CICS VR batch backout.

Recovering data sets using other recovery methods

If you do not run DWWBACK, problems can arise that affect replication and require additional actions. You must ensure that the source and target data sets are point-in-time consistent before restarting the batch job. The actions that you take depend on the type of CICS VR batch logging that is enabled.

The following examples describe the actions that you will likely need to perform to recover VSAM data sets when you do not run DWWBACK.

Example: No logging

A batch job fails that updates VSAM data sets defined with FRLOG(NONE). The action that you take depends on whether you can restart the batch job to reverse the failed changes.

- If the batch job is restartable, you can restart the job at the source and no action is needed at the target.

- Otherwise, you need to restore the data sets. You can take the following actions:
  1. Stop replication for the subscription with a controlled stop.
  2. Park the replication mapping.
  3. Restart replication for the other replication mappings in the subscription.
  4. Restore the source data sets.
  5. Re-run the batch job.
  6. Determine a quiesce point.
  7. Stop replication for the subscription with a controlled stop.
  8. Copy the source data sets to the target and use the quiesce point to set the log position for the replication mapping.
  9. Restart replication.

Example: Forward recovery logging

A batch job fails that updates VSAM data sets defined with FRLOG(REDO). Changes applied by CICS VR during forward recovery are not replicated. You need to restore the target data sets to match the source data sets after forward recovery processing. To restore the data sets to a point-in-time by forward recovering changes against the data sets, you can take the following actions:

- If the batch job is restartable, you can restart the job at the source and no action is needed at the target.
- To restore the data sets to a point-in-time by forward recovering changes against the data sets, you can take the following actions:
  1. Stop replication for the subscription with a controlled stop.
  2. Park the replication mapping.
  3. Restart replication for the other replication mappings in the subscription.
  4. Forward recover the source data sets.
  5. Re-run the batch job.
  6. Determine a quiesce point.
  7. Stop replication for the subscription with a controlled stop.
  8. Copy the source data sets to the target and use the quiesce point to set the log position for the replication mapping.
  9. Restart replication.

Example: Undo logging

A batch job fails that updates VSAM data sets defined with FRLOG(UNDO). If the batch job was running for an hour and the subscription is fairly current, the bookmark will be after the fall-back source copy (a copy of each data set prior to the start of the batch job). In this case, you cannot restore the data sets back to the before-job copy and copy them to the target because you will not be able to set a log position earlier than the bookmark and restart replication successfully. To recover the data sets, you can take the following actions:

- If the batch job is restartable, you can restart the job at the source and no action is needed at the target.
- Otherwise, you can take the same actions as described for FRLOG(NONE).

Recovery scenarios associated with CICS and batch job failures

In the following recovery scenarios, VSAM log records are being replicated and a CICS failure or a batch job failure occurs.

The failure can occur after a log record is written but before the VSAM data set is updated. The failure can also occur between the time when the change to a VSAM data set is written and when the log record is written. In these situations, you might need to take actions to recover the VSAM data sets.

Non-Recoverable files are being replicated using MASSINSERT and a CICS TS failure occurs

In this case, the CICS Transaction Server (CICS TS) address space fails while changes to VSAM data sets are buffered and after the corresponding log records are written.

Data Replication for VSAM cannot detect or correct this failure. A data inconsistency occurs for KSDS and RRDS if the key is written after the failure, because the key is already at the target. For an ESDS, the next record that is written will cause a RBA mismatch.

You might need to reload all non-recoverable data sets after a CICS or batch job failure. You can use comparison tools, such as SuperC, to limit reloads by checking the accuracy of the target data sets compared to the source data sets. For a large number of data sets, reloading non-recoverable data sets might be the best choice to avoid the time that individual comparisons might take.

Non-Recoverable data sets are being replicated when MASSINSERT is not used and a CICS TS failure occurs

In this case, the CICS TS address space fails after changes to VSAM data sets are complete and before the corresponding log records are written.

Data Replication for VSAM cannot detect or correct this failure. A data mismatch is detected when the key is changed, but the key is not available at the target

You might need to reload all non-recoverable data sets after a CICS or batch job failure. You can use comparison tools, such as SuperC, to limit reloads by checking the accuracy of the target data sets compared to the source data sets. For a large number of data sets, reloading non-recoverable data sets might be the best choice to avoid the time that individual comparisons might take.

Recoverable ESDSs are written using MASSINSERT and a CICS TS failure occurs

In this case, the CICS address space is lost while ESDS changes are buffered and after the corresponding log records are written.

The source server will release ESDS inserts to the target server for apply processing in write order and break transactional consistency to maintain the target RBA. At this point, the target data set is corrupted in that it contains records that do not exist in the source and those records cannot be reversed.

At CICS restart, a write_delete record is written. This record notifies Data Replication for VSAM that an insert was replicated to the target that was not actually written to the source VSAM data set. Because VSAM does not support delete operations to an ESDS, the source server will stop replication for the subscription and issue message CECV0106E to the event log.

You need to synchronize the target data set from the source and restart replication after setting the log position of the replication mapping. Set the log position to a time after the reload and before the file is available to be updated.

For additional information about how CICS handles locking, see the related references to topics in the CICS library.

Recoverable non-ESDS data sets are written using MASSINSERT and a CICS TS failure occurs

In this case, the CICS address space is lost while changes that do not include ESDS inserts are buffered and after the corresponding log records are written.

The source server will will not forward uncommitted changes to the target server for apply processing. When CICS restarts, Data Replication for VSAM rolls back and discards the UOR.

Recoverable data sets are written when MASSINSERT is not used and a CICS TS failure occurs

In this case, the CICS address space is lost after data set changes are complete, but before the corresponding log records are written. Data Replication for VSAM handles this case and no user action is required.

At restart, CICS backs out the changes or invokes the logical delete exit for ESDS inserts. For ESDSs, the replication log will contain a a write_update record for a logical delete. Data Replication for VSAM ignores the original UOR and the changes written in backout with the exception of the ESDS update written by the logical delete exit.

CICS VR changes are replicated and a batch job fails

This scenario occurs while changes to VSAM data sets are buffered and after the corresponding log records are written.

Data Replication for VSAM cannot detect or correct this failure. A data mismatch is detected when the key is changed and that change is replicated.

You might need to reload all data sets modified by the batch job. You can use comparison tools, such as SuperC, to limit reloads by checking the accuracy of the target data sets compared to the source data sets. For a large number of data sets, reloading non-recoverable data sets might be the best choice to avoid the time that individual comparisons might take.

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, 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 16. 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
Logger service	CACLOG	LOG
Monitoring service	CECMAA	MAA
Region controller service	CACCNTL	CNTL
Remote operator command service	CACOPER	OPER
VSAM log reader service	CECLRS	LRSV

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 following table lists the configuration parameters that apply to the PAA service class.

Table 17. 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

Table 17. Configuration parameters for the PAA 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	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 18. 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.
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.

Table 18. Configuration parameters for the APLY service class. (continued)

Parameter	Default value	Description
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.
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 transactions in CICS 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.
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 transactions in CICS that are started when the apply service starts.

Table 18. Configuration parameters for the APLY service class. (continued)

Parameter	Default value	Description
*Do not use the DEPGRAPHKEYS, DONEUORLIMIT, DONEUORTRACKING, IMSV11CMPAT, IMSBOOKMARKDB, or STRICTVALIDATION configuration parameters for Data Replication for VSAM.		

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.

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 19. 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.

Table 19. Configuration parameters for the INIT service class (continued)

Parameter	Default value	Description
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 20. 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

Table 20. Configuration parameters for the LOG service class (continued)

Parameter	Default value	Description
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 following table lists the configuration parameters that apply to the MAA service class.

Table 21. 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 22. 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
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 23. 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

Table 23. Configuration parameters for the CNTL service class (continued)

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	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 VSAM.		

VSAM log reader service

The VSAM log reader service is assigned to the LRSV service class. The task name for the LRSV service class is CECLRS.

The VSAM log reader service must communicate with the capture service. For communication between these services to occur successfully, the VSAM log reader service and the capture service must be defined in the same data server.

The VSAM log reader service processes stream requests received from the capture service. Stream requests enable the VSAM log reader service to return VSAM change data from the necessary log streams.

Restriction: Only one instance of an VSAM log reader service can be defined in a single data server.

Note: This service does not apply to Classic CDC for IMS.

The following table lists the configuration parameters that apply to the LRSV service class.

Table 24. Configuration parameters for the LRSV service class.

Parameter	Default value	Description
INITIALTHREADS	1	Number of instances of this service that the region controller starts during initialization of the data server
SEQUENCE	0	Sequence number assigned to services
TRACELEVEL	4	Trace level

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 25. 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

* Only the MESSAGEPOOLSIZE parameter applies to Data Replication for VSAM.

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 26. Data type validation

Data type value	SQL data type	Data validated	Repair value
P	DECIMAL	Packed decimal data	-9

Table 26. Data type validation (continued)

Data type value	SQL data type	Data validated	Repair value
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

This value is set to 16777216 during the installation and customization process. The configuration indicates that the default value is changed.

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.

100 CNTL
INIT
LOG
MAA
OPER

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 121 summarizes the configuration parameters associated with each service.

APPLYBUFFERSIZE:

The APPLYBUFFERSIZE configuration parameter defines the size of the buffer that Data Replication for VSAM 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

CICSBOOKMARKDB:

The CICSBOOKMARKDB parameter specifies the name of the CICS file to use as a bookmark file for Data Replication for VSAM.

Description

The value of the CICSBOOKMARKDB parameter is the name of the CICS file that Data Replication for VSAM uses as the bookmark file.

Specifications

Use: Configuration parameter for the apply service.

Service class: APLY

Service task: CECAPLY

Data type: CHAR(8)

Default: None

Valid values: Any valid CICS file name.

CICSCONINTRVL:

The CICSCONINTRVL parameter defines the frequency at which the apply service retries connecting to CICS when CICS is not available at target server startup.

Description

The default value for this parameter is 15S (seconds). This value indicates that the apply service will retry failed CICS connection attempts every 15 seconds.

Specifications

Use: Configuration parameter for the apply service.

Service class: APLY

Service task: CECAPLY

Data type: TIME

Default: 15S

Valid values:

*n***MS**	*n* milliseconds
*n***S**	*n* seconds
*n***M**	*n* minutes

*n***H** *n* hours

CICSPORT:

The CICSPORT parameter specifies the port number that the apply service uses to communicate with the target CICS region for Data Replication for VSAM.

Description

The CICSPORT value must match the TCP port number that the CICS IP Socket Listener in the target CICS region monitors for connection requests from the Classic apply service.

Specifications

Use: Configuration parameter for the apply service.

Service class: APLY

Service task: CECAPLY

Data type: INT

Default: 0

Valid values: 0 – 65535

CICSRECONINTRVL:

The CICSRECONINTRVL parameter defines the frequency at which the apply service retries establishing a CICS connection after losing the connection.

Description

This wait time ensures that the apply service does not try to reconnect to CICS while CICS is still in the process of stopping.

The CICSRECONINTRVL parameter is enforced when the console message CECW0041E is issued to ensure that CICS has enough time to stop before the apply service attempts to reestablish a connection.

The default value for this parameter is 1M (minute). This value indicates that the apply service will start trying to connect to CICS one minute after a disconnect is received from CICS and message CECW0041E is issued.

Specifications

Use: Configuration parameter for the apply service.

Service class: APLY

Service task: CECAPLY

Data type: TIME

Default: 1M

Valid values:

*n***MS**	*n* milliseconds
*n***S**	*n* seconds
*n***M**	*n* minutes
*n***H**	*n* hours

CICSTSQNAME:

The CICSTSQNAME configuration parameter defines the name of the temporary storage queue (TSQ) that Data Replication for VSAM uses in the target CICS region.

Description

The value that you specify for the CICSTSQNAME configuration parameter must assign a unique name for the TSQ. This value must conform to the CICS rules for naming TSQs. The default value is CFCQ1.

Specifications

Use: Configuration parameter for the apply service.

Service class: APLY

Service task: CECAPLY

Data type: CHAR

Default: CFCQ1

COMMSTRING:

The COMMSTRING parameter specifies the protocol identifier and address for communication.

Description

The connection handler supports TCP/IP. 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

CONFLICTRPTLVL:

The CONFLICTRPTLVL configuration parameter enables informational messages during adaptive apply processing. The messages report conflicts that occur during record insert, update, or delete operations.

Description

The informational messages that describe the conflict situation are written to the event log. They indicate the name of the subscription and source system ID, the name of the file and the data set, and the key of the record that caused the situation.

The CONFLICTRPTLVL parameter enables informational messages for adaptive apply processing only. For example, a conflict occurs when a record insert is replicated and the record already exists in the target data set.

- For standard apply processing, this situation indicates an error. In this case, a message describing this situation is written to the event log and record dumps are written to the diagnostic trace log. The replication process stops and return an error condition.
- With adaptive apply processing, this situation does not indicate an error. When you specify a value of 1 or 2 for the CONFLICTRPTLVL parameter, an informational message that describes the situation is written to the event log. The current change operation is skipped but replication continues with the next change operation for the UOR.

Specifications

Use: Configuration parameter for the apply service.

Service class: APLY

Service task: CECAPLY

Data type: INT

Default: 1 (Summary)

Valid values:

0 (None)
No information-level message is written to the event log and a dump of record images is not written to the diagnostic trace log.

1 (Summary)
An information-level message is written to the event log. This is the default.

2 (All) An information-level message is written to the event log and dumps of the appropriate record images and log records are written to the diagnostic trace log. The following dump records are written:

- Insert: The record that was inserted into the source file and the existing record from the target file.
- Delete: The record that was deleted from the source file and the record from the target file (if any).
- Update: The old and new records from the source file and the record from the target file (if any).

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

2 Generates the following event messages:

- 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

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

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. 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 VSAM 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

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 CICS writer tasks that the target server can start.

Description

The Classic data server uses the value of the MAXWRITERTHREADS configuration parameter to determine the maximum number of CICS writer tasks that can be started.

When you define MAXWRITERTHREADS, you also need to consider the following associated parameters:

- The value of MAXWRITERTHREADS is an important factor in defining the CICS system initialization table (SIT) parameters MXT and EDSALIM. You need to set these parameters to a sufficient value to ensure that the Classic data server can start the correct number of writer threads. The CICS Transaction Server (CICS TS) Version 5.1 defaults should be sufficient for most environments.
- The value of MAXWRITERTHREADS must be greater than or equal to the value of the WRITERTHREADS configuration parameter.

See "WRITERTHREADS" on page 156 for specific guidelines about defining each of these parameters.

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

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:

- 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.

Monitoring connections

The following optional parameters for the monitoring service activate security checking for client connections from the Classic Data Architect:

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

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

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

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.

USECICSMAPFIRST:

The USECICSMAPFIRST configuration parameter specifies whether the CICS writer transactions should use the CICS mapping file CECCFMAP to search for mappings between VSAM data sets and local CICS files, in addition to remote CICS files, before searching the File Control Table (FCT).

Description

The CICS mapping file CECCFMAP contains mappings between local and remote CICS files and their associated VSAM data sets defined in the replication mapping for a subscription.

When the configuration parameter USECICSMAPFIRST is set to TRUE, the apply service searches CECCFMAP for a mapping between the data set and the local CICS file definition. Specifying TRUE improves performance when subscription processing starts.

When the configuration parameter USECICSMAPFIRST is set to FALSE or is not defined, the File Control Table (FCT) is searched for the files defined to CICS. This method consumes more resources. CECCFMAP is only searched for data sets that do not match existing CICS file definitions.

For additional information about the CICS mapping file, see *Mapping CICS file names to VSAM data set names*.

Specifications

Use: Configuration parameter for the apply service.

Service class: APLY

Service task: CECAPLY

Data type: Boolean

Default: FALSE

Valid values: TRUE or FALSE

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 CICS writer tasks 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 CICS writer tasks to start when the target server is started.

CICS loads a system initialization table (SIT) at the start of its initialization process. You need to set the following parameters to a sufficient value to ensure that the Classic data server can start the correct number of writer threads. In most cases the default value is sufficient.

Note: The default values of the following parameters are increased in CICS Transaction Server for z/OS (CICS TS), version 5.1.

- MXT controls the maximum number of tasks in CICS. The default in CICS TS 5.1 is 500.

  You need to ensure that the value that of the MXT parameter is large enough to account for the utility program thread and any writer program threads that the Classic data server starts. You need to set this parameter to a number that is greater than the sum of the following elements:
  - 1, for the utility program transaction.
  - The maximum number of writer threads that the Classic data server can start, based on the value of the MAXWRITERTHREADS configuration parameter.
  - The number of any other CICS transactions that will run in the CICS region
- EDSALIM specifies the upper limit of the total amount of storage within which CICS can allocate the individual extended dynamic storage areas (EDSAs) that reside in 31-bit (above-the-line) storage. The default in CICS TS 5.1 is 800M.

  You need to set a value for a sufficient amount of storage to prevent CICS processing from stopping. Also ensure that any REGION parameters specified in the CICS JCL are sufficiently large to allow for an increase to the EDSALIM value.

The value of WRITERTHREADS must be less than or equal to the value of the MAXWRITERTHREADS configuration parameter.

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.

```
►►──DISPLAY──,──REPL──,──┬─SUBSCR──=──subscription-name─┬──────────────────────►◄
                         └─SUBSCR──=──*─────────────────┘
```

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.

```
►►──DISPLAY──,──REPL──,──┬─SUBSCR──=──subscription-name─┬──,──┬────────┬───────►◄
                         └─SUBSCR──=──*─────────────────┘     └─DETAIL─┘
```

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.

```
►►──DISPLAY──,──REPL──,───MAPPING──=──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.

```
►►──DISPLAY──,──REPL──,──┬─SUBSCR──=──subscription-name─┬──,──┬────────────┬──────►◄
                         └─SUBSCR──=──*─────────────────┘     └─THRESHOLDS─┘
```

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.

```
►►──START──,──REPL──,──┬──SUBSCR──=──subscription-name──┬─CONTINUOUS─┬──┬──,──────►◄
                       │                                └─DESCRIBE───┘  │
                       │      ┌─CONTINUOUS─┐                            │
                       └─ALL──┤            ├────────────────────────────┘
                              └─DESCRIBE───┘
```

Parameters

subscription name
: The name of the subscription that contains the data sets 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
: 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.

```
►►──SET──,──REPL──,──┬──SUBSCR──=──subscription-name──┬──,──────────────────────────►
                     └──ALL───────────────────────────┘

►──┬───────────────────────┬───┬─────────────────────────┬──────────────────────────►
   └─'APPLYCACHE──=──n'────┘   └─'CAPTURECACHE──=──n'────┘
```

```
►──┬──────────────────────────┬──┬──────────────────────────────────┬──────────────────►
   └─'PERSISTENT──=──┬─YES'─┬─┘  └─SETLOGPOS──,──┬─GMTTIME──=────┬─┘
                     └─NO'──┘                    └─SERVERTIME──=─┘

►──┬─────────────────────────────────────────────────┬─────────────────────────────────►
   └─'STATUS──=──┬─ACTIVE─┬──,──OBJECT──=──object'───┘
                 └─PARK───┘

►──┬────────────────────────────────────────────────────────────┬──────────────────────►◄
   ├─┬─THRESHOLDS1─┬──,──┬─DISCRETETIME─┬──=──threshold─────────┤
   │ ├─THRESHOLDS2─┤     ├─MAXTIME──────┤                       │
   │ ├─THRESHOLDS3─┤     └─RESETTIME────┘                       │
   │ └─THRESHOLDS──┘                                            │
   ├─┬─THRESHOLDS1─┬──,──┬─HEARTBEAT─┬──=──timeval──────────────┤
   │ └─THRESHOLDS──┘     └─MEANTIME──┘                          │
   └─┬─THRESHOLDS2─┬──,──WORKLOAD────=──workload-name───────────┘
     └─THRESHOLDS3─┘
```

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.

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.

Ensure that all data sets that are part of the subscription are closed in all CICS regions and in CICS VR at the specified time. This ensures that CICS and CICS VR will generate a tie-up record when the data sets are be opened after the specified time.

`"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.

```
►►──STOP──,──REPL──,──┬─SUBSCR──=──subscription-name─┬──,──┬────────────────┬──►◄
                      └─ALL──────────────────────────┘     │ ┌─CONTROLLED─┐ │
                                                           ├─┴────────────┴─┤
                                                           ├─IMMEDIATE──────┤
                                                           └─CAPTURE────────┘
```

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.

Recommendation: Run Data Replication for VSAM as a started task in production environments, or in environments where performance is a concern, to ensure that it is given priority as a higher service class.

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: 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 27. 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

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.

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
```

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>
```

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
```

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 message list information related to the logger service, issue the following command:

```
DISPLAY,CONFIG,SERVICE=LOG
```

Parameters

listname
The service list name. You can specify one of the following service list names:
- EIFEVENTSERVERS
- MSGLIST

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

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 28. Service classes.

Service class	Service type
APLY	Apply service
CAP	Capture service

Table 28. Service classes. (continued)

Service class	Service type
CNTL	Region controller service
GLOB	Global service
INIT	Client connection handler service
LOG	Logger 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=,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=,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:

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:

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)
```

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.

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).”
2. Create filters for the output. See “Filters for modifying output from the log print utility (CACPRTLG)” on page 181.
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 .

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.

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