2.4

IBM Tivoli Advanced Backup and Recovery for z/OS User's Guide

SC23-9730-04

Note:

Before using this information and the product it supports, read the "Notices" topic at the end of this information.

2023-09-05 edition

This edition applies to Version 2 Release 4 of Advanced Backup and Recovery for z/OS (product number 5698-B17) and to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

Chapter 1. Advanced Backup and Recovery for z/OS Overview

Advanced Backup and Recovery Manager is a comprehensive data backup and recovery solution. It is designed to safeguard your data and streamlines recovery processes.

This software comprises one common install, a common inventory data set for ABARS Manager and Critical Backup Tracking and Inventory, a separate database for Automated Critical Data Identification, a separate control file and database for Fast Replication Services (FRS), and common panels. Advanced Backup and Recovery for z/OS® is designed to meet your company's business resiliency needs either locally, or in the event of a disaster, from the minor to the catastrophic. Advanced Backup and Recovery for z/OS offers the following products and features:

- ABARS Manager
  - Enhances and extends ABARS functionality
  - Centralizes management of ABARS to make ABARS easy to use
  - Consolidates and manages essential tracking, status, and resource information
  - Adds functionality not included in native ABARS
  - CATSCRUB, designed for full volume restore, enables you to synchronize quickly one or more BCS catalogs with associated DASD volumes at your disaster recovery site
  - Incremental ABARS enables you to perform ABARS backups on an incremental basis
- Critical Backup Tracking and Inventory
  - Centralizes and inventories z/OS backups so you can recover anywhere, anytime
  - Identifies what has not been backed up at your site
- Automated Critical Data Identification
  - Identifies critical business data as the application executes
  - Identifies the data that needs to be backed up for business resiliency on-site or off-site (or both) and builds the list as input to a backup utility
  - Historical Dataset Usage feature shows, for a given search data set, all references to the search data set within a specific time frame
- Fast Replication Services
  - A storage-aware data backup and recovery solution
  - Provides instantaneous data backup and recovery options while using reduced CPU, I/O, and storage resources
- Critical Backup Tracking and Inventory and Automated Critical Data Identification working together
  - Use the list created by Automated Critical Data Identification to compare to what was actually backed up
  - Provide documented evidence that critical assets have a backup
  - Provide proof that critical business data is off-site if required
  - Alert on critical assets that don't have a backup

Advanced Backup and Recovery for z/OS panels provide access to functionality common to all features, ABARS Manager, Critical Backup Tracking and Inventory, and Automated Critical Data Identification, and are named either Advanced Backup and Recovery for z/OS or BKM.

ABARS Manager panels are named either ABARS Manager or ABM, Critical Backup Tracking and Inventory panels are named either Critical Backup Tracking and Inventory or CBTI, and Automated Critical Data Identification panels are named either Automated Critical Data Identification or ACDI.

Why use ABARS Manager instead of native ABARS?

ABARS Manager provides powerful enhancements and extensions to ABARS processing and availability. ABARS Manager is specifically designed to enhance, simplify, and automate Disaster Recovery and Local processing using ABARS backup and recovery functions. ABARS Manager makes ABARS and DFSMShsm more powerful, flexible, reliable, and easier to use.

Incremental ABARS allows you to combine the Disaster Recovery Solutions provided by ABARS, with the benefits of incremental backup strategy.

CATSCRUB synchronizes catalogs and volumes for Disaster Recovery at blazing speeds. The result is catalogs that correctly reflect the data on physical volumes.

ABARS Manager provides the following benefits:

- Centralized management and control information in a powerful Inventory Data Set (IDS) with in-depth tracking features
- Batch operation for routine operations and planned activities
- Interactive operation for detailed analysis, error correction, research, process restart, selective recovery, overlap analysis, and VTOC information
- Tape information is automatically created for recovery execution
- Extensive supplemental information held in the IDS, including Selection Data Set information and activity log details, as well as error status.
- The following extended ABARS functionality and reliability features:
  - Selection Data Set Language (SDSL) is a powerful tool that expands the ABARS language to over 30 different attributes for INCLUDE or EXCLUDE of data sets. Boolean logic makes it easy to define powerful selection rules.
  - Prevent BackLevel ensures that an application is not back-leveled by an older copy of a data set being restored.
  - GDG **REPLACE/NOREPLACE** gives you the control to restore non-GDS using the **NOREPLACE** parameter for non-GDS files, and **REPLACE** specifically for GDS files.
  - Selective Data Set Restore allows you to restore specific data sets, data sets you identify using masking, or data sets independent of the backup name.
  - Extended Rename provides the capability to rename more than the high level qualifier.

ABARS and other utilities

Tracking and restoring ABARS backups requires ABARS Manager. Unlike other utilities, ABARS-tracked backups require initiating backups using an ABARS Manager procedure.

If you use both ABARS Manager and Critical Backup Tracking and Inventory, all backed-up data sets (that is, if backed up either by ABARS or other z/OS backups) are presented in a consolidated view.

Automated Critical Data Identification

Automated Critical Data Identification uses information from job scheduling products, JCL libraries, and SMF records to help storage administrators easily determine which data sets should be included as part of an application backup. Automated Critical Data Identification provides storage administrators with a set of sophisticated tools to implement and manage an installation's application backup strategy.

Installations often back up unnecessary data using mask filters or manual lists that are often not maintained because identifying the data to be backed up can be a time-consuming task. Automated Critical Data Identification solves this problem by automating the task of identifying the critical data to be backed up. Using information from a variety of OS/390® or z/OS sources, Automated Critical Data Identification automatically constructs a list of data sets used by an application each time the application runs. The list of critical data sets is immediately available as input to a backup, or it can be used for analysis.

Automated Critical Data Identification features and benefits

Using Automated Critical Data Identification helps ensure that all critical data is included in the backup, while saving resources and reducing costs by backing up only the data required.

Automatically detects application changes

Automated Critical Data Identification automatically detects application changes that might require changes to backup selection criteria.

Support for forward and rerun recovery methods

Determining which data sets are critical can be a complex task without Automated Critical Data Identification. Detailed decisions depend largely on the planned recovery methodology. Each organization must make specific decisions about each application as to whether forward recovery (sometimes called forward resumption) or rerun recovery will be used. Automated Critical Data Identification provides automated support of rerun recovery and substantial support for forward recovery. Override and option settings are available either globally or at the application level to assist you in configuring Automated Critical Data Identification appropriately.

Job scheduler interfaces

Automated Critical Data Identification provides interfaces to these job scheduling products:

- ASG-Zeke for z/OS
- BMC CONTROL-M for z/OS
- Cybermation ESP
- Tivoli® Workload Scheduler (TWS)
- Unicenter CA-7 Job Management
- Unicenter CA-Jobtrac Job Management

Mirroring support

Automated Critical Data Identification supports remote mirroring to track tape data or other data that should be mirrored. Automated Critical Data Identification mirroring support provides several benefits.

Verifies the correct data sets are being mirrored

If you have data sets (for example on volumes PROD*) that are being mirrored and you want to identify data sets for those applications that were allocated on volumes outside of the mirrored volumes, you can define all applications on the PROD* volumes to Automated Critical Data Identification and it will track the applications as they run. You can review the Selection Data Set on a daily basis to identify those data sets not on mirrored volumes.

Identifies data sets on specific volumes

By browsing the contents of the Automated Critical Data Identification Selection Data Set, you can view the volume where each data set is located (which can alert you to potential problems). For example, if a data set should be located on a PROD* volume and it is located on a TEST volume, or another volume that is not mirrored, you can easily determine if you need to take action.

Identifies tape data sets

By excluding all data on volumes other than tape, Automated Critical Data Identification will provide a list of tape data. Automated Critical Data Identification can also identify data and distinguish between migration level 1 (ML1) DASD and migration level 2 (ML2) tape data sets. For example, by excluding all data on volumes other than TAPE and MIGRA2, Automated Critical Data Identification will provide a list of data that resides on tape.

Note: Automated Critical Data Identification does not determine tape sizing— it only determines whether or not the data set is on tape.

Identifies data required for point-in-time backups
If you are mirroring batch data you still need a point-in-time backup. If an error occurs in batch, and it is duplicated to the mirrored DASD, you might not be able to back out of the error because it was duplicated at your remote site and there was no point-in-time backup.

Reporting

Automated Critical Data Identification provides a variety of reports that can be used to list filters, the number of Data Set Detail (DSD) records, and other relevant information for each application tracked by Automated Critical Data Identification.

Automated Critical Data Identification and Critical Backup Tracking and Inventory

Together, Automated Critical Data Identification and Critical Backup Tracking and Inventoryprovide these features:

- The ability to compare the Automated Critical Data Identification critical data set list against Critical Backup Tracking and Inventory discovered backups.
- The ability to restore data sets for Automated Critical Data Identification applications and specific cycles within an application.
- Reports to interrogate tape management systems (TMS) to determine the location of backup tapes.
- Provide proof that critical data is backed up.
- Provide alerts for those data sets that are not backed up.

Automated Critical Data Identification components

Automated Critical Data Identification consists of several components. Some components are initiated by started tasks, and others are initiated by jobs that you must customize for your environment.

Automated Critical Data Identification Application Data Inventory database
The Automated Critical Data Identification Application Data Inventory database, a VSAM KSDS database, maintains and tracks application information. The information gathered by the job collection process is stored in the Automated Critical Data Identification Application Data Inventory database.

The Automated Critical Data Identification Application Data Inventory database provides a repository for user overrides, application level selection and analysis options, and details of job and data set collection and data set activity. Automated Critical Data Identification uses application information in the inventory to create Selection Data Sets in various supported formats.

Real time selection process (RSP)
After you place an application under Automated Critical Data Identification administration, RSP continuously monitors application activity to ensure the application critical data set list is current and accurate. RSP effectively handles a variety of typical but troublesome processing modes such as applications that run overlapping cycles, and applications that create, or use, pseudo-GDG data sets (data sets whose names change in a systematic non-standard manner).

RSP provides two methods you can use to collect data: RSP SMF Capture and RSP JCL Capture.

RSP SMF Capture
RSP SMF Capture uses the IEFU84 and IEFU83 SMF exit facilities to collect SMF data in real time, which it then uses to detect cycle control job completion and reevaluate the criticality of data sets in the application.

RSP JCL Capture
RSP JCL Capture uses the IEFUJI JCL exit facility to collect interpreted JCL information in real time.

BKMAPLEN
RSP uses an added job step, the application end step (BKMAPLEN), to signal that a full application cycle has been completed. This allows RSP to process any remaining SMF records, update

the Automated Critical Data Identification Application Data Inventory database, and rebuild the application Selection Data Set for the next backup event. There can be one, or many, application end steps in an application job stream, depending on the complexity of job relationships.

Job table
For each image, RSP builds an internal data space table, called the *job table* (or job reference table), containing the job names defined to each application. All data collection methods read the job table.

Job scheduler interfaces
Automated Critical Data Identification collects information about an application's jobs and job dependencies using multiple methods to gather information from a variety of sources. One method is to use an interface to a job scheduling product. Automated Critical Data Identification provides interfaces to a variety of job scheduling products.

Selection data set
The Selection Data Set contains the most recent critical data set entries to be backed up. Each application has a unique Selection Data Set. You define the Selection Data Set during the application setup process using an output format that is compatible with the backup product you are using. Automated Critical Data Identification populates the Selection Data Set using one of the following supported output formats:

- ABARS — (Default format) Uses standard INCLUDE, ALLOCATE and ACCOMPANY type entries.
- DSS — Uses the logical dump data set format.
- DMS — Uses primary data set entries.
- FDR/ABR — Uses the logical dump data set format using the FILTERDD ddname.
- DSSL — Uses the logical dump data set format using the FILTERDD ddname.
- CA-Disk — Uses primary data set entries.
- List — Generic list format.

SMF Scan
SMF Scan is intended for those installations that are not ready to implement RSP or for those installations that want an historical analysis of an application. SMF scan is a batch process that you can use to scan a user specified collection of captured SMF data. Identifying application data sets through SMF Scan is an optional step performed after job collection.

Verification
During Verification processing, Automated Critical Data Identification generates backup control statements in the selected output format and places them in the Selection Data Set. The Verification program can be used to test the creation of the Selection Data Set and determine the critical Selection Data Set entries for an application.

Historical Dataset Usage
The Historical Dataset Usage feature provides a simple method for determining all of the places where a given data set is referenced within a specific time period. For each reference that is found for the search data set, information about the referencing data set is displayed: data set name, application ID, SMF record type, and the date and time at which the reference was made. You can display additional information about any of the referencing data sets: date and time information for backups, SMF information, and information about how the data set was evaluated by Automated Critical Data Identification.

Automated Critical Data Identification architecture and processing

Through the use of various methodologies, functionality, and support capabilities, Automated Critical Data Identification provides you with the tools you need for critical data set identification in the most complex processing environments.

An overview of Automated Critical Data Identification architecture and the process flow to identify critical data are shown in Figure 1 on page 6.

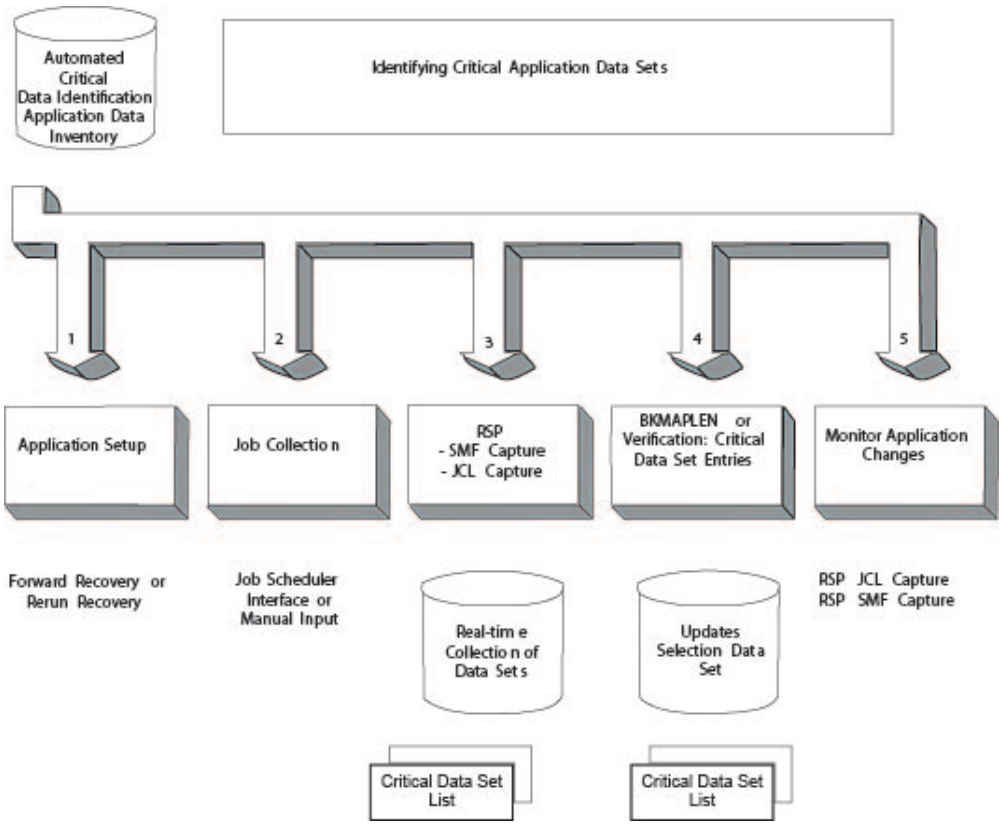


Figure 1. Automated Critical Data Identification architecture and process flow

1. During the application setup process, you define, for each application, information such as the recovery method and GDG support options. Automated Critical Data Identification uses this information to determine which data sets belong to an application and which data sets are critical for backup.
2. Automated Critical Data Identification performs the job collection process by gathering information about the jobs and job structures associated with an application from sources such as job scheduler products, job name masks, or user created lists. The information gathered by the job collection process is placed in the Automated Critical Data Identification Application Data Inventory database for later use in the data gathering process.
3. RSP continuously monitors the activity of the applications to ensure the application critical data set list is current and accurate. RSP captures and analyzes SMF and JCL data for applications defined to Automated Critical Data Identification as they run and updates the Automated Critical Data Identification Application Data Inventory database with the data set names used by the application.
4. BKMAPLEN or Verification is used to identify critical data and create the Selection Data Set. When creating the Selection Data Set, Automated Critical Data Identification combines data set detail records (DSDs), created from RSP JCL Capture, RSP SMF Capture, and SMF Scan into a single record for each data set. Data sets are evaluated for criticality. This step also inserts arbitrary data set names (external filters).
5. After the initial Selection Data Set has been created, Automated Critical Data Identification monitors changes to an application's data set usage and automatically maintains a current critical data set name list for use in backing up the application for local or disaster recovery purposes.

Application monitoring

Monitoring application changes ensures that any changes made to the application are reflected in the Selection Data Set and the Automated Critical Data Identification Application Data Inventory database.

After the job collection and real-time tracking is complete for an application, and the Selection Data Set has been created, both real-time and batch mechanisms are provided for collecting and monitoring data

set changes for any application. Figure 2 on page 7 shows the processes involved in monitoring and updating an application Selection Data Set.

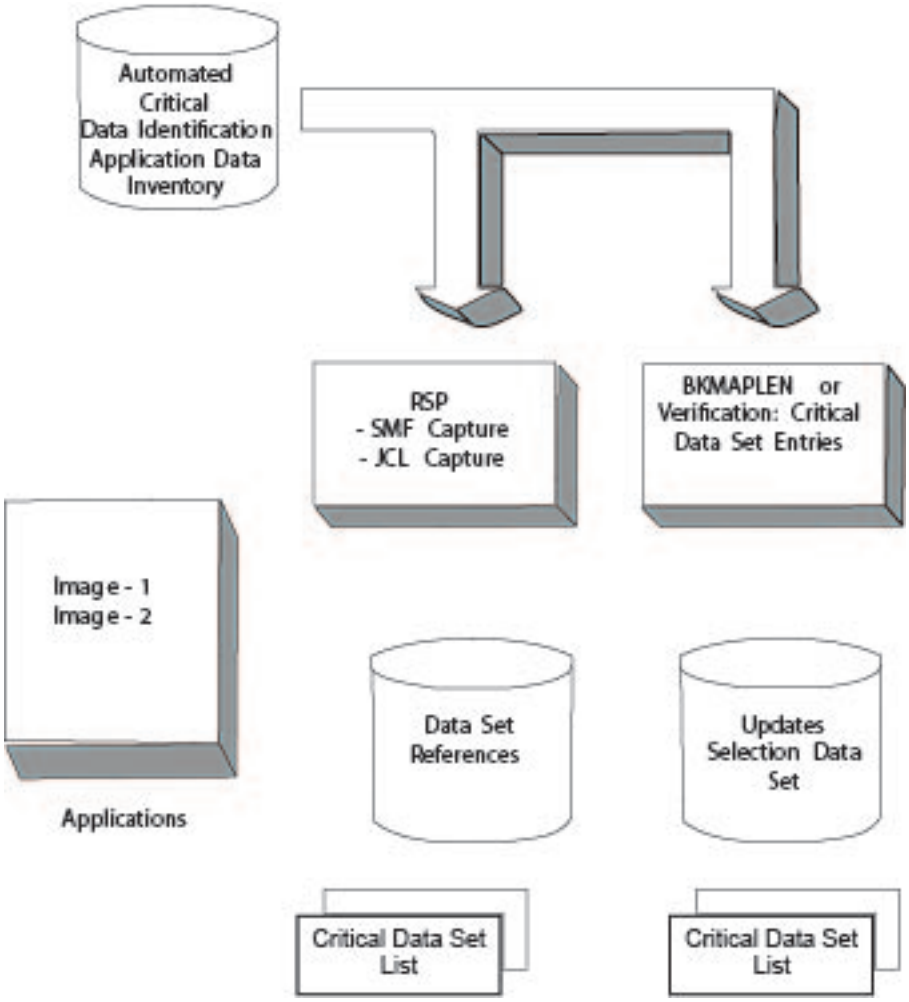


Figure 2. Continuous monitoring of application (in multi-image SYSPLEX modes)

RSP allows continuous application monitoring and collection of data set entries using RSP JCL Capture and RSP SMF Capture. RSP can monitor and track an application in both single and multi-image environments (or SYSPLEX). At the completion of an application's processing, the data set entries are updated and the Selection Data Set list is reconstructed, ensuring that the subsequent backup contains the most current critical application data sets.

Critical Backup Tracking and Inventory

Critical Backup Tracking and Inventory provides centralized backup control for business resiliency (both locally and for disaster recovery purposes).

Critical Backup Tracking and Inventory intercepts, inventories, and centralizes z/OS backups — without requiring JCL changes to existing backup utilities, giving you the ability to recover anywhere, anytime. It consolidates and manages essential backup tracking, status, and resource information for effective backup and recovery.

Critical Backup Tracking and Inventory:

- Provides business resiliency for local recovery or disaster recovery, from minor to catastrophic outages.
- Tracks and records backups and data without the need for any JCL changes to existing backup utilities.
- Quickly locates any backup copy recorded in the Inventory Data Set and easily enables you to display backup details and restore data.

- Provides easy to use ISPF and batch interfaces.
- Identifies what is *not* backed up at your installation to help ensure you are not missing critical data sets By tracking and providing an inventory of backups, you can easily find data sets not backed up by comparing them to the Inventory Data Set, DFSMShsm, BCDS, or both.
- Identifies multiple data set backups to help reduce resources.
- Automatically constructs restore JCL and utility control statements that will reverse the copy of a backed-up data set. Construction of recovery JCL control statements can be varied through the Critical Backup Tracking and Inventory file tailoring process.
- When used with Automated Critical Data Identification, Critical Backup Tracking and Inventory:
  - Uses the list created by Automated Critical Data Identification Data Inventory to compare to what was actually backed up
  - Provides documented evidence that critical assets have a backup
  - Alerts on critical assets that do not have a backup
  - Proves that critical business data is off-site if required

Critical Backup Tracking and Inventory features and benefits

Critical Backup Tracking and Inventory provides these features and benefits.

- Intercepts backups without JCL changes therefore making tracking backups simple to set up.
- Centralization enables you to easily determine what data is, or is not, backed up.
- Quickly locates backups. By tracking z/OS backups, Critical Backup Tracking and Inventory can locate any and all backups of the data set, regardless of the utility used, by simply specifying a data set name.
- Provides for business resiliency both locally and for disaster recovery:
  - Locally — Use Critical Backup Tracking and Inventory on site if a data set becomes deleted or corrupted, or a backout of a batch process is required.
  - Disaster Recovery — Use Critical Backup Tracking and Inventory off-site if a disaster strikes to easily restore and quickly get your systems up and running.
- Identifies what is not backed up at your site to help ensure you are not missing critical data sets that require a backup.
  - Compares data sets on your volumes, or in your ICF User Catalog, or both, to the Critical Backup Tracking and Inventory data set, and optionally to the DFSMShsm BCDS.
  - Critical Backup Tracking and Inventory identifies data sets that are not recorded in either and either have not been tracked yet or do not have a backup.
- Tracks multiple generations of backups. Critical Backup Tracking and Inventory can optionally track multiple generations of a backup, allowing restoration of a backup other than the most current generation.
- View data sets backed up from each utility including full volume dump, and provide data set restore off full volume dumps (a rare occurrence but sometimes necessary).
- Automatically recovers JCL and control statements. It is not uncommon for backups to be placed in production with no corresponding restore JCL or control statements created and available for quick access. Critical Backup Tracking and Inventory resolves this issue by automatically constructing the restore JCL or control statements that match the utility and commands used to originally back up data sets or volumes.
- Alerts if backups fail. Processes are not always set to fail or provide an alert if a backup fails. By tracking not only which data sets or volumes are backed up, but by also recording the success or failure, Critical Backup Tracking and Inventory provides a mechanism for alerting responsible parties that backups have failed.
- Interactive operation — for detailed analysis, error status, research, overlap analysis, and DSCB information.

- Identifies data sets backed up multiple times to save on tape and CPU resources.
- Identifies data sets that could possibly be overwritten during restore caused by:
  - Overlaps between full volume dumps and application backups.
  - Overlaps that can occur when multiple applications back up the same data set.

Critical Backup Tracking and Inventory components

Critical Backup Tracking and Inventory consists of several components. Some components are initiated by a started task and others are initiated by JCL that you must customize for your environment.

Critical Backup Tracking and Inventory (BKMSTSKA) started task
The started task intercepts supported in-stream backups performed during a backup and provides a method for tracking them in the Inventory Data Set.

Step initiation exit (IEFUSI)
Starts the Critical Backup Tracking and Inventory job intercepts.

Batch Intercept File (BIF)
The batch intercept file contain batch intercept file entries. Batch intercept file entries identify the jobs, steps, or program names to track.

Intercept modules for supported backup utilities
These modules enable you to track the backup utilities supported by Critical Backup Tracking and Inventory.

Inventory Data Set
A common Inventory Data Set is used for both ABARS Manager and Critical Backup Tracking and Inventory.

ISPF interface
Critical Backup Tracking and Inventory provides an ISPF-based user interface that enables you to easily perform backup and restore management tasks.

Batch utilities
The Critical Backup Tracking and Inventory batch utilities enable you to load and unload data from the Inventory Data Set, expire and remove backups that are no longer valid from the Inventory Data Set and generate a variety of reports.

How Critical Backup Tracking and Inventory tracks backups

To avoid requiring JCL changes to existing backup utilities, Critical Backup Tracking and Inventory identifies backups to be tracked using a seamless intercept triggered by matching job names, step names, and utility program names to a user-defined table. Although it is necessary to identify which jobs are backups, the risk and procedural difficulties associated with modifying production JCL are eliminated.

You identify backup jobs to Critical Backup Tracking and Inventory through batch intercept file entries. The batch intercept file entries identify a backup job or step and identify the expected backup utility.

To intercept backups, Critical Backup Tracking and Inventory uses a started task on each image which installs step initiation exit (IEFUSI). If a backup is to be intercepted, a resource manager is added. The resource manager gets control at step termination and collects the data needed to track the backup.

Subtasks process the results of intercepted utilities and record the pertinent data in the Inventory Data Set. For most types of backups, Critical Backup Tracking and Inventory captures catalog and volume metadata in synchronization with the status of the data sets at the time of the backup. This ensures that restore allocations match the status of data sets at the time they were backed up.

Backup utilities tracked by Critical Backup Tracking and Inventory

This information describes the backup utilities tracked by Critical Backup Tracking and Inventory.

- ABARS backups performed through ABARS Manager
- Adabas (ADARUN)

- Brightstor CA-FAVER (GVEXPORT)
- CA-Disk (ADSST001)
- Catalog RecoveryPlus (CR+) backups
- CFCAMS (REPRO and EXPORT)
- DFSMSdss (Full volume, physical dump by data set, logical backups, and Unix files backup)
- DFSMShsm ARCINBAK, AUTODUMP, and Incremental Backup
- FDR (FDR, FDRABR, FDRCPK, FDRDSF)
- IBM® Tivoli Advanced Catalog Management backups
- ICEGENER
- ICEMAN
- ICETOOL
- IDCAMS (REPRO and EXPORT)
- IEBCOPY
- IEBGENER
- SORT
- SYNCGENR
- SYNCSORT
- TAPECOPY (OTTC0003)
- VSAM Assist (VSAMASST)

Note:

- DFSMShsm AUTODUMP is supported through batch utilities.
- Typically, SORT is used to SORT data. Sorts of this type are not considered backups and should not be tracked by Critical Backup Tracking and Inventory. The function of Critical Backup Tracking and Inventory is to track data set backups. Critical Backup Tracking and Inventory tracks SORT because some installations might have jobs that sort data to an output file that they consider a backup. These are valid backups. Make sure that discrete job names are specified in the backup tracking INCLUDE filters instead of masks for any SORT job you want to track, in order to prevent tracking sort output files that are not backups, or to prevent failures. SORT functions often use PDS members as input to the SORT, however Critical Backup Tracking and Inventory requires you to use the full partitioned data set name. Member names are not supported.
- Critical Backup Tracking and Inventory does not support backups where the data sets backed up are defined by concatenated input files. For example, an IEBGENER step where the SYSUT1 DD statement uses concatenated files is not supported by Critical Backup Tracking and Inventory.

Tracking and restoration of VSAM KSDSs with alternate indexes

Some backup and restore utilities have a *SPHERE* option which causes the utility to back up all components and associations of a VSAM sphere. When using a utility that does not have a SPHERE option, Critical Backup Tracking and Inventory requires only that the base cluster be backed up. Backups of AIXs are ignored.

When a backup of a VSAM base cluster is tracked, all association information is saved in the Inventory Data Set. This information is then used to define and rebuild any associated AIXs and define any associated PATHs. This feature provides the sphere backup and restore capability to utilities that do not inherently perform this function.

After verifying the AIXs and PATHs are appropriately restored, you might want to remove potential unnecessary backups of AIXs, and user procedures, to capture catalog PATH information.

ABARS Manager overview

ABARS Manager provides powerful enhancements and extensions to ABARS processing and availability. Designed to enhance, simplify, and automate disaster recovery and local processing, it uses native ABARS backup and recovery functions.

You can use Incremental ABARS to combine the disaster recovery solutions provided by ABARS with the benefits of incremental backup strategy.

ABARS Manager's CATSCRUB feature lets you synchronize catalogs with volumes speedily so that catalogs correctly reflect the data on physical volumes.

Benefits of using ABARS Manager

Why would you want to use ABARS Manager?

- It provides centralized management and control information in the Inventory Data Set (IDS), with in-depth tracking features.
- You can use ABARS Manager in batch mode for routine operations and planned activities.
- You can use ABARS Manager in interactive mode for these purposes:
  - Detailed analysis
  - Error correction
  - Research
  - Restarting processes
  - Selective recovery
  - Overlap analysis
  - Obtaining VTOC information
- Tape information is created automatically for recovery execution.
- The IDS stores extensive supplemental information, including information about Selection Data Sets, activity log details, and error status.
- ABARS Manager's Selection Data Set Language (SDSL) extends the ABARS language to over 30 attributes for the inclusion or exclusion of data sets. Boolean logic simplifies the definition of powerful selection rules.
- The pre-processing of Selection Data Sets reduces runtime errors by identifying and correcting common ABARS syntax errors.
- The Prevent BackLevel feature ensures that an application is not back-leveled by an older copy during a data set restoration.
- The GDG REPLACE/NOREPLACE option gives you the control to restore non-GDS using the NOREPLACE parameter for non-GDS files and REPLACE specifically for GDS files.
- Selective data set restoration lets your restore specific data sets. You can identify the data sets by masking or you can indicate that you want specific data sets restored, regardless of the backup name.
- You can use the Extended Rename feature to rename multiple high-level qualifiers.

About ABARS and other utilities

Tracking and restoring ABARS backups requires ABARS Manager. Unlike other utilities, ABARS-tracked backups require that you initiate backups by using an ABARS Manager procedure.

If you use both ABARS Manager and Critical Backup Tracking and Inventory, all backed-up data sets (regardless of whether they were backed up by ABARS or other z/OS backup methods) are displayed in one consolidated view.

Fast Replication Services overview

FRS is a storage-aware backup and recovery solution that integrates storage processor fast-replication facilities with file based backup and recovery operations to allow instantaneous backups, reduce recovery time, and simplify disaster recovery procedures while using less CPU, I/O, and storage resources.

FRS provides a fast and easy-to-use implementation of a fast-replication backup and recovery methodology. It reduces backup windows by leveraging storage-based fast-replication such that backups of large amounts of data can be performed in seconds or less. It simplifies backup and recovery methodologies by allowing application level or selected data set recoveries to be performed from a point-in-time backup. Consistent backups can be created using application level point-in-time Fast Replication Services.

It provides application-level backup and recovery support even for complex applications, where all of the application's data must be backed up, restored, and recovered as a unit. In addition, when creating application level backups, FRS invokes storage-based fast-replication facilities using a Fast Replication Engine (FRE) through appropriate storage processor APIs, reducing host CPU and I/O resource utilization.

A FRS backup and recovery methodology can simplify application backup and recovery procedures and ensure that application data sets are recovered and all data relationships are preserved during a recovery process. A FRS backup and recovery methodology is one where all of the application data sets are backed up as a unit and the backup can be used to recover the application as a unit, or it can be used to recover individual application data sets. All data relationships are implicitly maintained during the recovery process, guaranteeing that data integrity is preserved for the application. A FRS backup and recovery methodology is much faster to perform than using traditional utility program approaches. When you use the tape offload feature, the same FRS backup and recovery procedures can be used for local site recovery as well as for off-site disaster recovery purposes.

The FRS application backup solution leverages modern storage processor capabilities and fast-replication products to perform backup and restore operations on behalf of the application. Supported fast-replication products include: IBM FlashCopy®, EMC TimeFinder/Mirror, EMC TimeFinder/Clone, EMC TimeFinder/Snap, and Hitachi ShadowImage. The advantages of using the FRS application backup methodology which uses storage-based fast-replication include:

- Complete application backups can be performed in seconds or less.
- There is very little or no impact to applications while the backup is performed.
- Point-in-time restartable backup copies of an entire application are created.
- No host CPU and I/O resources are used to create the backup.
- Application backup offloads can be used for local/remote application or data set recovery.
- Application backups can be offloaded using an independent tape copy process to reduce application disk storage utilization while preserving the backups created in multiple backup cycles.

The advantages of using a FRS backup to perform an application level recovery operation include:

- The entire application can be restored from disk instantaneously when storage-based fast-replication is used to restore the data.
- The application is recovered as a unit so that all data relationships are preserved during the restore and recovery processes, thus ensuring that the application's data integrity is preserved.

The advantages of using a FRS backup for off-site disaster recovery include:

- A FRS application level backup can be easily copied to tape and transported to a disaster recovery site.
- The application can be restored using a standard backup and recovery utility.

Usage scenarios

You can use Advanced Backup and Recovery Manager for z/OSData Recovery Manager Suite to meet your business resiliency needs either locally, or in the event of a disaster, from the minor to the catastrophic.

These scenarios illustrate how you can use Advanced Backup and Recovery Manager for z/OSData Recovery Manager Suite to address typical business problems.

Disaster recovery using Automated Critical Data Identification and Critical Backup Tracking and Inventory

You manually identify data used by production batch applications, started tasks, or individual users for disaster recovery purposes and want to avoid backing up more data than you need. You are concerned that you might not be backing up critical data and might not be able to recover your data in a timely manner.

Automated Critical Data Identification and Critical Backup Tracking and Inventory enable your business to benefit from an automated round-the-clock solution that identifies data used by each application for the purpose of protecting the data for both local and off-site recovery.

Automated Critical Data Identification

By using Automated Critical Data Identification you can provide an autonomic solution for the identification of your data.

Automated Critical Data Identification:

- Runs continuously, identifying data used by production batch applications, started tasks, or individual users
- Eliminates the manual, time-consuming, and error-prone task of identifying data for backup
- Identifies data as it is used, thus eliminating the need to back up data using masking every day, while reducing tape and CPU resources

Automated Critical Data Identification eliminates the need for manual and often error-prone identification of your business data requiring backup for disaster recovery preparedness. Automated Critical Data Identification accurately identifies data to ensure successful recovery. Because Automated Critical Data Identification identifies critical data as it is used by an application, it provides a reduction in backup and recovery windows, and, in turn, tape and CPU resources.

Critical Backup Tracking and Inventory

By using Critical Backup Tracking and Inventory, you centralize and inventory your backups to help ensure fast and accurate data recovery. Critical Backup Tracking and Inventory intercepts, inventories, and centralizes backups without requiring JCL changes. Critical Backup Tracking and Inventory tracks all backups that occur in the environment and reports on data that is not backed up.

With Critical Backup Tracking and Inventory you can:

- Avoid backing up the same data multiple times
- Ensure data sets are not back-leveled during restore
- Prove critical data is backed up
- Display multiple generations and recover from older backups
- Quickly locate any backup to ensure you select the appropriate backup for restore
- Identify the data that has not been backed up

Critical Backup Tracking and Inventory centralizes backups for local or disaster recovery so you can confidently locate and restore the correct data. Critical Backup Tracking and Inventory identifies data sets that are not backed up. When used with Automated Critical Data Identification, Critical Backup Tracking and Inventory provides an Application Backup and Restore feature that notifies you if critical data is not backed up.

Using Critical Backup Tracking and Inventory to centralize and inventory backups

Your backups are not centralized and therefore there is no way to easily identify what data is not backed up. You are concerned about the impact to your business if critical data assets are not backed up.

When backups are not centralized:

- Confusion exists about what is and what is not backed up because there is no inventory.
- Problems can occur at the local or disaster site in identifying all available backups and selecting the appropriate one for restore.
- Data sets might be backed up multiple times, tying up resources and escalating the likelihood of error.
- There is no way to easily determine what data is not backed up.

Critical Backup Tracking and Inventory enables you to centralize your backups and easily determine whether or not your critical data is backed up.

You use Critical Backup Tracking and Inventory to centralize your backups for local and disaster recovery so you can confidently find and restore the correct data. Critical Backup Tracking and Inventory shows you how to avoid backing up data multiple times to reduce tape and CPU resource requirements, and identifies data sets that do not have a backup.

Chapter 2. Starting a product installation or upgrade

The initial steps for a first-time installation are the same as they are for a product upgrade. This topic provides instructions for these common steps.

To start a product installation or an upgrade, complete these steps:

1. Read carefully the information that is provided in “Product notes” on page 19
2. Use the instructions in “Performing a full installation using SMP/E” on page 15 to install the base product.
3. Follow the instructions that are provided in Applying included maintenance.

Performing a full installation using SMP/E

Use the procedures that are provided in this topic to install Advanced Backup and Recovery for z/OS using SMP/E.

Note: Refer to the *SMP/E configuration notes* appendix for information you might find helpful for planning your SMP/E environment.

Complete these steps to perform an SMP/E installation:

Moving the installation files to the mainframe

You obtained the product installation files by downloading them from the web site to a PC. This topic provides instructions for moving the installation files from the PC to the mainframe.

1. Execute the `*.zip` file to extract the SMP/E binary transfer file (`SMPBIN`) and the SMP/E install JCL file (`ISMPJCL`) to the PC.
2. In binary mode, upload the extracted files to the mainframe.

   Note: Most emulators allow you to specify data set attributes for new data sets. Because there might be size restrictions for allocations, you might find it easier to use ISPF option 3.2 to allocate the receiving data sets before you upload them from the PC:

   - LRECL=80
   - RECFM=FB
   - DSORG=PS
   - For the install JCL file, use TRK,(1,1)
   - For the transfer file, use CYL,(40,40)
3. Make a note of the names you assign to the install JCL and transfer files because you need to provide these names later in the installation process.

Editing the install JCL

The base product install JCL must be modified appropriately for your site.

The product transfer files were created with the `TSO TRANSMIT to a data set` feature, requiring a **`RECEIVE`** command to unload. The `ISMPJCL` file that was shipped with the transfer file contains the necessary **`RECEIVE`** commands that correspond to the contents of the transfer file.

Note: To ensure that your libraries are propagated with the correct set of members, do not alter the content of the install JCL file except as instructed by the procedure that is provided in this topic.

1. Change all occurrences of `TEMPHLQ?` to the high-level qualifier that you specified when you copied or uploaded the transfer and install JCL files (`ISMPJCL.TEMP`, for example).
2. Change `SMPEHLQ?` to the high-level qualifier you want to use for your SMP/E permanent data sets (`ISMPJCL.SMP`, for example).

3. Optional: If you are familiar with **RECEIVE**, you are likely aware that if the **DATASET** keyword is not specified, **RECEIVE** defaults to forming the target data set name by substituting the INDA data set name high-level qualifier with the profile prefix value. It is not recommended that you delete the **DATASET** specification, but if you must do so, you must also insert a blank line between **RECEIVE** commands.

Submitting the install JCL

The modified install JCL must be submitted for execution.

1. Submit the edited ISMPJCL data set for execution.
2. When execution has finished, verify that the **RECEIVE** commands completed successfully by examining the IEB1098I and IEB147I output messages.
   - IEB1098I lists the number of members loaded and the number of members that should have been loaded.
   - IEB147I lists the end of the job and shows the highest condition code. Verify that the highest condition (or severity) code is zero.

Editing and submitting SMP/E INSTALL library members

The SMP/E INSTALL library members must be modified appropriately for your environment before you submit them for execution.

These members perform the following tasks:

- Define the SMP/E global zone environment
- Define the product CSI, target zone, and DLIB zone environments
- Define the members that perform SMP/E **RECEIVE**, **APPLY**, and **ACCEPT** processing

Refer to the *SMP/E configuration notes* appendix for detailed information about planning your SMP/E environment.

The SMP/E JCL and install method described in this topic are examples of how the SMP/E installation could be done. You are free to perform the SMP/E installation as you want and you are responsible for creating the JCL and methods for doing so.

For SMP/E installation, the SMP/E install library is created by the **RECEIVE**. The library should contain the JCL members described in the following table.

Member name	Description
BKMJ1SMA	Allocates the SMP/E temporary libraries needed for setting up a new environment for installation and maintenance.
BKMJ2SMI	Defines and initializes the SMP/E CSI, as well as the Global, Target, and Distribution zones. The job also provides SMP/E data set DDDEFs. You need to run this job if you want to have a separate CSI for Advanced Backup and Recovery for z/OS.
BKMJ3ALO	Allocates the target and distribution libraries for Advanced Backup and Recovery for z/OS.
BKMJ4DDF	Defines the SMP/E DDDEF zone entries for Advanced Backup and Recovery for z/OS.
BKMJ5REC	**RECEIVE** components for Advanced Backup and Recovery for z/OS.
BKMJ6APP	**APPLY** components for Advanced Backup and Recovery for z/OS.
BKMJ7ACC	**ACCEPT** components for Advanced Backup and Recovery for z/OS.
	These members are used for applying PTFs as maintenance.

1. If you are setting up a new environment for product installation and maintenance, edit BKMJ1SMA to allocate the SMP/E temporary libraries.
2. If you want to have a separate CSI for this product, edit and submit the install library member BKMJ2SMI to define and initialize the SMP/E CSI and the Global, Target, and Distribution Zones, and provide SMP/E data set DDDEFs.

   Note: It is strongly recommended that you use the MVS™ SREL(Z038) Global CSI for the MVS SREL ID.
3. Edit and submit the install library's member BKMJ3AL0 to allocate the Target and Distribution libraries for Advanced Backup and Recovery for z/OS.
4. Edit and submit the install library's BKMJ4DDF member to define the SMP/E DDDEF Zone entries for Advanced Backup and Recovery for z/OS.
5. Edit and submit the install library's BKMJ5REC member to perform a **RECEIVE** of the components for Advanced Backup and Recovery for z/OS.
6. Edit and submit the install library's BKMJ6APP member to perform an **APPLY** (with the **CHECK** parameter) to validate the SMP/E environment.
7. Remove the **CHECK** parameter from the install library's BKMJ6APP member and resubmit it to install the components into the product libraries.

   Note: You might notice what appear to be duplicate members in the nonload product libraries; however, these duplicate members are only aliases. Be aware that if you use a single set of zones for all of your Rocket Mainstar products, performing the **APPLY** of the new release removes the older release from the target zone.
8. Edit and submit the install library's BKMJ7ACC to perform an **ACCEPT** of the components for Advanced Backup and Recovery for z/OS.
9. Optional: Back up the new Advanced Backup and Recovery for z/OS SMP/E environment.

Tailoring the product for your installation

Review the product distribution transmittal for special instructions related to the current release and take the appropriate action for your installation.

Chapter 3. Starting a product installation or upgrade

The initial steps for a first-time installation are the same as they are for a product upgrade. This topic provides instructions for these common steps.

These are the common steps for starting an installation or an upgrade.

1. Review the information in the “Product notes” on page 19 section.

   Note: The *Product notes* section provides important information about the software, authorization requirements, the distribution files and methods, the product data sets, how to use the necessary user exits, and other types of overview information. It is strongly recommended that you review it carefully.
2. Perform a full installation according to the instructions in “Performing a full installation” on page 23.
3. Apply all included maintenance for this release, according to the instructions in “Applying periodic maintenance” on page 23.

When you have completed these common steps, go on to the appropriate chapter for your situation:

- Chapter 4, “Completing a product installation,” on page 25
- Chapter 5, “Completing a product upgrade,” on page 49

Product notes

Before you begin installing Advanced Backup and Recovery for z/OS, you might find it helpful to understand the software and authorization requirements, the software distribution files and methods, which data sets are involved, how to use the necessary user exits, and other types of overview information.

Advanced Backup and Recovery for z/OS data sets

You can choose any data set names for the Advanced Backup and Recovery for z/OS data sets during installation. You must specify your chosen data set names in the `:INSTALLATION_DATASETS` and `:PRODUCT_DATASETS` sections of the BKMINI member of the product parameter library.

You may choose any high level qualifier names for the product data sets during installation. You will also supply that high level qualifier name to the batch procedure JCL members contained in the sample library in a later step in the installation procedure.

Note: If possible, keep the low level qualifiers shown in the following list. The user documentation and support personnel refer to the files by these names.

These are the product data sets:

***hlq*.ACDI**
: The ACDI database is the information repository for the Automated Critical Data Identification feature of Advanced Backup and Recovery for z/OS. This database contains information about applications that are defined to and monitored by ACDI functions. This data set is a VSAM KSDS structure and is not created automatically during installation. The database is created as one of the configuration tasks.

***hlq*.DAD**
: The Data Set Activity Database (DAD) is used to support incremental ABARS backups and Critical Backup Tracking and Inventory functionality. It contains information about change activity for data sets that were backed up for ABARS incremental aggregates and Critical Backup Tracking and Inventory tracked batch utility backup jobs. This data set is a VSAM KSDS structure and is not automatically created during installation. The DAD is created by one of the configuration tasks.

***hlq*.IDS**
: The Inventory Data Set (IDS) is the information repository for the ABARS Manager and Critical Backup Tracking and Inventory features of Advanced Backup and Recovery for z/OS. It contains information

about which aggregates have been backed up or recovered, which data sets were backed up or recovered as part of an aggregate, **ABACKUP** or **ARECOVER** tape volume usage, and other related information. For Critical Backup Tracking and Inventory, the IDS contains information about tracked batch utility backup jobs and data sets backed up. This data set is a VSAM KSDS structure and is not automatically created during installation. The IDS is created by one of the configuration tasks.

***hlq*.HDU**
The Historical Dataset Usage database stores information about all references to data sets. The information is stored for a maximum of 90 days. The Historical Dataset Usage database is created during the configuration process by tailoring and running JCL library member BKMDHDU.

These are the installation data sets:

***hlq*.SBKMASMP**
This library contains sample assembler source code for optional programs you might choose to use (modify, assemble, and link edit). Some of these programs might be system version-level or release-level dependent, and it is your responsibility to test and maintain any programs used from this library.

***hlq*.SBKMDBRM**
This library contains the database request modules used to access DB2® image copy information for the Db2 Image Copy Reporting feature.

***hlq*.SBKMEXEC**
This is the Automated Critical Data Identification REXX execution library. Some of these EXECs are in a special format used by Automated Critical Data Identification and are not readable if viewed or edited. This data set is automatically created when the product is installed.

***hlq*.SBKMJCL**
This library contains JCL and cataloged procedures. Because these members might be modified by future maintenance, do not modify the members unless you are directed to do so by Technical Support or by the product documentation. This data set is automatically created during installation.

***hlq*.SBKMLOAD**
This library contains the Advanced Backup and Recovery for z/OS load modules. The load library must be APF authorized before you begin the post-installation configuration tasks. This data set is automatically created during installation.

***hlq*.SBKMMENU**
This library contains the messages that are issued by the Advanced Backup and Recovery for z/OS ISPF interface. The data set is part of the distribution library set.

***hlq*.SBKMMSGS**
This library contains the product messages and their descriptions. This data set is created automatically during installation.

***hlq*.SBKMPARM**
This library contains information needed by Advanced Backup and Recovery for z/OS during product execution, such as configuration information and information about maintenance that has been applied. This data set is automatically created during installation.

***hlq*.SBKMPENU**
This library contains ISPF panels that are displayed when you use the Advanced Backup and Recovery for z/OS ISPF interface. The data set is part of the distribution library set.

***hlq*.SBKMRSMP**
This library is used by ABARS Manager and contains sample REXX source code for use with the ABARS Aggregate LoadBalancer facility. It is your responsibility to test and maintain programs used from this library. Contact Technical Support for further information or assistance using programs in this library. This data set is automatically created during installation.

***hlq*.SBKMSKEL**
This library is used by Critical Backup Tracking and Inventory only and contains the default skeleton restore JCL that is used during restores of backups that were tracked by Advanced Backup and Recovery for z/OS. This data set is automatically created during installation.

***hlq*.SBKMTENU**
This library contains the tables used by the Advanced Backup and Recovery for z/OS interface during operation. The data set is part of the distribution library set.

ABARS Selection Data Set

ABARS Manager proactively fixes errors in two ways: it uses the Backup Options setting in the BKMINI member of the product parameter library and it supplies a sample ABARS exit (ARCBEEXT) that bypasses all errors.

Fixing errors by using the BKMINI Backup Options settings

ABARS Manager uses the optional Backup Options settings in the product parameter library to fix errors using these steps:

1. ABARS Manager offloads the ABARS Selection Data Set.
2. ABARS Manager stores a copy of the Selection Data Set in the Advanced Backup and Recovery for z/OS Inventory Data Set (IDS).
3. ABARS Manager moves the incorrectly categorized data sets into the correct category. For example, it would move a DASD data set in the ACCOMPANY list to the INCLUDE list because ACCOMPANY supports only tape data sets.
4. ABARS Manager submits the **ABACKUP** command to ABARS.
5. When the ABARS **ABACKUP** has been run successfully, ABARS Manager updates the IDS with the aggregate backup information.
6. ABARS Manager restores the original contents to the Selection Data Set.

If the Backup Options are used and the **SDS_MODIFICATION_OK** token in *your.product*.SBKMPARM(BKMINI) is set to Y, ABARS Manager rewrites the Selection Data Set to contain the correct data set entries. If you set this token to N, ABARS Manager does not fix any errors and does not write to the Selection Data Set.

Selection Data Sets set up in a PDS do not work well with ABARS Manager when `SDS_MODIFICATION_OK=Y` because only one program can write to a PDS at a time. If PDS files are used, eventually there will be failures because it is very likely that two backups will be executing and attempting to use the PDS at the same time. Sequential files are always recommended, but if a PDS type structure is required, use PDSE.

PDSEs have some minor disadvantages compared to sequential data sets:

- Adding a new Selection Data Set (member) to a PDSE might fail because the PDSE is out of space, whereas failure is less likely if Selection Data Sets are individual sequential data sets.
- ISPF, Automated Critical Data Identification, and ABARS Manager can coexist peacefully when accessing different members or even the same member of a PDSE. However, Automated Critical Data Identification or ABARS Manager might fail if other programs lock the entire library either because they must (as is the case during a reorganization) or because they were not designed to be friendly (for example, locking the entire library when they really only needed exclusive use of a member). If the data set is locked out, ABARS Manager tries five times at 30-second intervals and then fails.

Fixing errors by using ARCBEEXT

Advanced Backup and Recovery for z/OS provides the sample IBM ABARS exit ARCBEEXT for ABARS Manager. This exit bypasses all errors. If you are using this exit, consider evaluating the sample settings.

Using the ARCSKEXT and ARCCREXT ABARS exits

ABARS Manager provides a number of extended functions by using the capabilities of the ABARS ARCSKEXT and ARCCREXT exits.

- ARCSKEXT is the Aggregate Recovery bypass (or skip) exit. This required exit is loaded automatically for all ARECOVER commands that are submitted through ABARS Manager
- ARCCREXT is the Aggregate Recovery conflict resolution exit. This required exit is loaded automatically for all ARECOVER commands that are submitted through ABARS Manager.

For proper operation of ABARS Manager, you must activate the ARCSKEXT and ARCCREXT exits in the ARCCMDxx member of the HSM parameter library. Use the DFSMShsm SETSYS operands EXITON(ARCSKEXT,ARCCREXT).

IBM provides versions of these two exits as part of the Advanced Backup and Recovery for z/OS distribution library. Local installation versions of these exits are also supported by ABARS Manager. The IBM version of these exits receives control first; then, ABARS Manager provides control to the entry point of the local version of each exit, if it was provided.

Note: This DFSMShsm exit environment must be established at the primary site and at the recovery site for ABARS Manager to provide all services and functions in both backup and recovery operations.

About using the ARCBEEXT exit

The ABARS backup error installation exit (ARCBEEXT) allows you to skip a data set when DFSMSdss is dumping level zero DASD data sets in the INCLUDE list.

Note: Advanced Backup and Recovery for z/OS provides a sample IBM ABARS ARCBEEXT for ABARS Manager. This exit bypasses all errors. If you are using this exit, consider evaluating the settings in the sample.

The ARCBEEXT is an optional exit that can be used two ways:

- To bypass errors, thereby increasing the likelihood that the backup completes successfully
- To fail the backup if the data set should be included in the backup

Advanced Backup and Recovery for z/OS provides a sample exit that is already coded to bypass these errors.

These are common errors that the ARCBEEXT bypasses:

- An uncataloged data set is encountered. The data set is in the list of data sets to back up but is not cataloged.
- An I/O error occurs during a backup of a data set.
- A data set fails serialization.
- A DFSMSdss dump processing error occurs.

For more information, refer to the DFSMShsm manuals.

APF authorization requirements

The Advanced Backup and Recovery for z/OS load library requires explicit APF authorization, even if it appears in the system link list with the LINKAUTH=LNKLST option (this requirement is included in the documented configuration procedure).

Although SMS-managed load libraries can be assigned APF authorization, most sites prefer to have all authorized libraries reside on specific non-SMS system volumes.

About data spaces

Several Advanced Backup and Recovery for z/OS programs use local data spaces for tables. A generalized routine allows the data spaces to be extended as table space is needed. For unusual amounts of data

that might exceed the capacity of a data space, the mechanism transfers data from the data space to a temporary VSAM data set against which remaining activity occurs.

Performing a full installation

Regardless of whether you are performing a first-time installation or are upgrading a previous installation, you must perform the installation steps that are provided in the product program directory.

Your product distribution includes the Advanced Backup and Recovery for z/OS Program Directory.

- Follow the installation instructions that are provided in the Program Directory.

Applying periodic maintenance

To take full advantage of product functionality that might have been released more recently than the base product, apply periodic maintenance to the base product installation before you move on to the configuration procedures.

To apply periodic maintenance, perform these steps:

- "Step 1: Backing up the base product data sets" on page 23
- "Step 2: Obtaining PTF maintenance from IBM" on page 23
- "Step 3: Performing SMP/E maintenance" on page 23
- "Step 4: Copying the maintenance to the existing production libraries" on page 23
- "Step 5: (Optional) Refreshing LINKLIST Look Aside" on page 24

Step 1: Backing up the base product data sets

It is strongly recommended that you make backup copies of the base product data sets so that you can restore them if the need arises.

- Take a backup of the Advanced Backup and Recovery for z/OS product data sets. Identify this backup in such a way that you can clearly tell it was taken prior to applying product maintenance.

Step 2: Obtaining PTF maintenance from IBM

The IBM Support web site provides access to all of the available PTFs for Advanced Backup and Recovery for z/OS.

1. Visit http://www-01.ibm.com/software/support/isa/ and use the IBM Support Assistant to obtain the PTFs.
2. Upload the maintenance to the mainframe.

Step 3: Performing SMP/E maintenance

Use SMP/E JCL to RECEIVE and APPLY the maintenance

Because you might need to restore a PTF, it is recommended that you do not execute an SMP/E ACCEPT on the maintenance.

Step 4: Copying the maintenance to the existing production libraries

All of the members of all libraries that were changed by the periodic maintenance must be copied to the production libraries.

1. Copy the libraries and appropriate members into your existing production libraries, using ISPF option 3.3, IEBCOPY, or some other utility that provides the necessary library copy options.
2. Copy from the target libraries all of the changed members that were listed in the `FIXLIST` data set, then retrofit your customizations to the new members.

3. From the target libraries, copy any other of the members that were identified in the FIXLIST data set as modified for the maintenance and to which you have not already applied the changes during previous maintenance.
4. Copy all other target libraries in full.

Step 5: (Optional) Refreshing LINKLIST Look Aside

If your product load modules are in LINKLST, you must refresh LLA.

Continue the installation or upgrade process by following the procedures that are provided in the appropriate chapter:

- Chapter 4, “Completing a product installation,” on page 25
- Chapter 5, “Completing a product upgrade,” on page 49

Where do I go next?

At this point, you have completed the initial steps for a first-time product installation or an upgrade and are ready to complete the installation or upgrade.

To complete your installation or upgrade, complete the steps that are provided in the appropriate chapter:

- If you are installing the product for the first time at your site, go on to Chapter 4, “Completing a product installation,” on page 25.
- Alternatively, if you are upgrading a previous version of Advanced Backup and Recovery for z/OS, go on to Chapter 5, “Completing a product upgrade,” on page 49.

Chapter 4. Completing a product installation

This topic provides instructions for completing a first-time installation.

If you followed the instructions in *Staring a product installation or upgrade* to begin an upgrade of a previous version of Advanced Backup and Recovery for z/OS, skip this chapter and go on to *Completing a product upgrade* .

Note: The Fast Replication Services (FRS) feature requires some minimal configuration before it can be used. Instructions for completing the FRS configuration are provided in “Preparing to use FRS” on page 303.

Completing an initial installation involves several configuration steps and setting up the started tasks:

Note: Technical Support might ask you to run BKMMODLV, should you encounter a problem during the configuration and started task set-up process. Instructions for running this job are provided in “Running BKMMODLV to validate the load module contents” on page 45.

Step1: APF authorizing the product load library

Begin your Advanced Backup and Recovery for z/OS configuration by APF authorizing the load library. This is a required task.

Consult your systems programmer to have this library added to the APF list and ensure appropriate access controls have been established.

Note: If your system parameter library member `IEASYS00` has parameter `LINKAUTH=LNKLST` specified, Advanced Backup and Recovery for z/OS does not honor it. The product load library must be APF authorized.

Step 2: Creating a new BKMINI member in the product parameter library

The next step in the installation completion process is to create a new BKMINI member in the product parameter library.

BKMINI keyword values are fetched one at a time at product startup. BKMIMERG creates a new BKMINI product parameter library member.

The JCL for running BKMIMERG is in the data set member*prefix.value*.SBKMJCL(BKMIMERG).

1. Provide a valid job card statement.
2. Change the STEPLIB data set name to the *prefix.value*.SBKMLOAD library.
3. Change the INI DD statement to point to *prefix.value*.SBKMPARM. Use *prefix.value*.SBKMPARM as the data set name and specify BKMINI# as the member name.
4. Change the MSCINI DD statement to point to the *prefix.value*.SBKMPARM data set, and ensure that the member name remains as BKMINI#.
5. Change the UPDATE DD statement to point to the *prefix.value*.SBKMPARM data set, and ensure that the member name remains as BKMINI.
6. Submit the job, and ensure that it executes and completes with a completion code 0 before starting the next task.

   If the job terminates with a nonzero completion code, first check to see if any error messages indicate a situation that is easily correctable. BKMIMERG can be run again. If the situation does not appear to be easily correctable, contact Technical Support for assistance before proceeding to the next task.

Step 3: Coding the variables in BKMINI

After you have created a new BKMINI, the next step is to code values for the variables in that member.

The values for the tokens in the BKMINI member of the product parameter library control many functions and processing, such as ISPF field defaults, batch command processing defaults, data set allocations, and so on.

Note: The values you specify can be changed directly by updating the BKMINI member of the product parameter library. After you have completed the customization process for the BKMINI member, the product parameter library data set should be treated as a read-only data set.

1. Update the tokens for **LOAD1**, **ISPPLIB1**, **ISPTLIB1**, **ISPMLIB1**, **PARMLIB**, **MSGLIB**, **SKELLIB**, **PACKED**, and **JCL** keywords, and ensure that the correct data set names are specified. The data set names specified here must be the libraries from which the product will be executed. In some cases, these are also the installation data sets, however, data centers can have various conventions that differ. If these token values do not accurately represent the target libraries, unpredictable results can occur.

   Be aware that the BKMINI member you are editing must reside in the library specified in the **PARMLIB** token. The data set you specify for the **LOAD1** token must be APF authorized.
2. If DFSMShsm is using RLS for the control data sets, set the RLS token values in the :SI040_VALUES section:
   a) Find the *VSAM-CLUSTER-NAME = RLS token.
   b) Delete the asterisk from the token name.
   c) Replace the string VSAM-CLUSTER-NAME = RLS with *YOUR.HSM.MCDS.NAME* = RLS.
   d) If you need to specify multiple DFSMShsm MCDSs or BCDSs, insert additional lines (for example, HSM.MCDS2 = RLS, HSM.BCDS2 = RLS, and so on).
   e) Only if you are directed to do so by Technical Support, specify the following RLS parameter: VSAM-CLUSTER-NAME.RLS_TIMEOUT = 25.
3. If you are using CA-MIM/MII, set the CA-MIM/MII token values in the :RESOURCE_SERIALIZATION section. Installations running CA-MIM/MII with multiple systems and shared DASD need to set the

BKMINI parameter **MIM_GDIF** in the :RESOURCE_SERIALIZATION section to YES. This ensures that when CA-MIM/MII GDIF is inactive, the Advanced Backup and Recovery for z/OS data sets are protected from data sharing corruption.

4. Review all of the keywords and tokens in the BKMINI member and set appropriate values for your installation. If you need more detailed information during this update process about the BKMINI sections and tokens, refer to the BKMINI configuration values appendix.
5. Verify that the default values coded for some keywords are appropriate for your installation.
6. Save the updated BKMINI member.

Step 4: Running the BKMJCLBL job

The next step is to run the job that replaces symbolics within the in-stream procedures within batch job JCL library members.

Advanced Backup and Recovery for z/OS batch job JCL library members use in-stream procedures. The in-stream procedures contain symbolics (such as &...) that refer to various product data sets. BKMJCLBL substitutes BKMINI token values into the PROC statements in the JCL library members.

Even though distributed product maintenance may not include new JCL members, changes to the BKMINI product parameter library member would necessitate this step. BKMJCLBL is in the product JCL library.

1. Modify the BKMJCLBL job JCL as indicated by the comments at the beginning of the member.
2. Be sure that the BKMINI tokens are set as desired before running BKMJCLBL. BKMJCLBL can be rerun at any time.
3. Submit the BKMJCLBL job.

Step 5: Creating the Automated Critical Data Identification database

The Automated Critical Data Identification database must be allocated and formatted before you can use Automated Critical Data Identification.

This database is a key-sequenced ICF VSAM data set that contains information used by Automated Critical Data Identification to identify applications, and to associate data sets and jobs with applications.

After the database has been initialized, IDCAMS REPRO or any other utility that handles ICF VSAM clusters can be used to back up, restore, or copy the contents of the database.

The option to retain extra cycles of SMF and JCL data might require a larger Automated Critical Data Identification database. The Automated Critical Data Identification started task, BKMSTSKR, writes directly to the Automated Critical Data Identification database from each contributing image.

Although a mechanism exists to allow a database reorganization or rebuild with no data loss, you need to ensure that the database is sized correctly and that candidate volumes are allocated for use if space is not available on the current volume. The appendices contain formulas for sizing your Automated Critical Data Identification database, suggestions for how to monitor the space usage, and a procedure for reallocating the database if it runs out of space.

The database should be defined with SHAREOPTIONS (2,3) to permit concurrent updating by multiple Automated Critical Data Identification users. Automated Critical Data Identification uses ENQ/DEQ locking to prevent more than one user from updating the same application (as defined to Automated Critical Data Identification) concurrently. While the Automated Critical Data Identification database might be shared across MVS systems in a sysplex, this is not recommended unless a cross-system enqueue mechanism, such as GRS or MIM, is in place.

The BKMDB01 member of the product parameter library contains IDCAMS control statements that you can use to define the database.

1. The first two statements are provided to allow you to delete an old copy of the database. If this is not necessary, just leave them as they are; they will not affect processing.

2. The name in the DEFINE CLUSTER must be the same as was coded in the BKMINI member for token ACDIDB in the :PRODUCT_DATASETS section. We recommend naming the INDEX and DATA components explicitly. The syntax is provided in BKMDB01.
3. If the database is allocated on SMS managed volumes, it should be assigned an SMS Management Class that prevents auto-backup.
4. Make the necessary changes to satisfy your installation's requirements.

   It is recommended that you allocate a candidate volume; the syntax example in BKMDB01 supports this. Do not change the **FREESPACE** parameter. Do not reduce the allocation amount to less than CYL(10,10).

   Allocate only in terms of cylinders.
5. Save your changes.
6. Make necessary updates to JCL library member BKMDEFN.
7. Verify that the JCL is correct.
8. Submit member BKMDEFN.
9. Check all return codes.

Step 6: Creating the Historical Dataset Usage database

The Historical Dataset Usage database stores the information needed by the Historical Dataset Usage search feature to locate all references to the search data set.

- Edit appropriately and submit JCL member BKMDHDU.

Step 7: Creating the DAD database

The DAD database supports incremental ABARS backups and Critical Backup Tracking and Inventory functionality. Edit and submit JCL member BKMDDAD to allocate the DAD database that is used by all sharing systems.

1. Set these parameters:
   a) Specify a space parameter. CYL(50,50) is recommended. It is also recommended that you monitor the size of this data set to ensure that it does not grow past the maximum VSAM size.
   b) Set the DAD database share options to 2 and the key to 56 bytes at position zero.
   c) The other values (CISZ and record size) are variable but recommended.
   d) If the DAD database is allocated on SMS-managed volumes, assign to it an SMS management class that prevents auto-backup.
2. Make the necessary changes to satisfy your installation's requirements.
3. Save your changes.
4. Verify that the JCL is correct.
5. Submit the JCL for execution.
6. Check all return codes.

Step 8: Creating the Inventory Data Set (IDS)

The Inventory Data Set is necessary to support Advanced Backup and Recovery for z/OS functionality. The JCL member BKMDEFDB allocates the IDS and primes it before the product is invoked.

1. Edit JCL member BKMDEFDB and adjust as necessary. The recommendation is to allocate candidate volumes and secondary space.
2. Verify that the name in the DEFINE CLUSTER is the same as was coded in the BKMINI member for parameter **INVENTORY_DATASET** in the :PRODUCT_DATASETS section.
3. If the IDS is allocated on SMS-managed volumes, assign an SMS management class that prevents auto-backup.

You can use the procedures in the product JCL library to backup or reorganize the Advanced Backup and Recovery for z/OS IDS.

4. Submit BKMDEFDB for execution.
5. Check all return codes.
6. If you find you need to expand the IDS after it is populated, do the following:
   a) Rename the current IDS.
   b) Modify and run BKMDEFDB to create a new IDS.
   c) Use IDCAMS REPRO to copy the contents of the old IDS to the new IDS.

Step 9: Ensuring adequate TSO address space region size

Ensure that your TSO region size is at least 6 megabytes. If you intend to run Advanced Backup and Recovery for z/OS in split screen mode, you might need an even larger region.

Step 10: Verifying the installation and configuration

At this point in the configuration process, you can cycle the product to determine whether it was properly installed and configured.

1. Enable TSO Profile options **WTPMSG** and **MSGID** so that error messages are written to the terminal.
2. Go to TSO ISPF option 6.
3. Use the following command to invoke Advanced Backup and Recovery for z/OS: EX '*hlq*.SBKMPARM(BKM)'
4. If the verification fails, contact Technical Support for assistance.

Step 11: Specifying and displaying EBCDIC characters

Your installation might need to specify different masking characters to achieve the appropriate result if your code tables differ from the USA EBCDIC code set.

For product code shipped in binary, when you are specifying input where the product takes special action based on specific characters, you are responsible for entering the EBCDIC character peculiar to your code tables, that results in the binary value for the EBCDIC character specified in the product manuals, according to the USA EBCDIC code set.

For example, if an exclamation mark (!) is called for, and your code tables do not translate the ! character to a hexadecimal 5A, you must enter the character that will translate to a 5A.

Do not change the distributed ISPF panels. Other programs might reference ISPF panel attribute bytes. A panel change involving an attribute byte will not be consistent with the program code.

Depictions of product output shown in the product manuals are based on the USA EBCDIC code set. Actual output might vary if your EBCDIC code tables are different.

Step 12: Completing the post-installation tasks

After you have performed the initial installation of Advanced Backup and Recovery for z/OS and completed the configuration tasks, you need to perform several post-installation tasks before Advanced Backup and Recovery for z/OS is completely ready for use.

These are the post-installation tasks:

- "Task 1: Updating the cross-system ENQ lists" on page 30
- "Task 2: Verifying that the configuration values were set in BKMINI" on page 31
- "Task 3: Provide access authority" on page 31
- "Task 4: Adding BKMWABAR to the TMOEDMxx file" on page 31
- "Task 5: Replacing the DFHSMABR procedure" on page 31

Task 1: Updating the cross-system ENQ lists

The IDS can reside on shared DASD and can be accessed from multiple images only if you have a product such as MIM or GRS to ensure serialization. The VSAM I/O code underlying Advanced Backup and Recovery for z/OS relies on the propagation of enqueues to all images that access a common data set.

Enqueues all use a major name of SIS2000. The reserves that Advanced Backup and Recovery for z/OS uses against QNAMES can and should be converted to ENQs, with scopes of SYSTEMS. Ensure that this requirement is brought to the attention of the person who handles your serialization package.

Experience with GRS has shown that cross-system ENQs are honored without defining the QNAMEs. However, with MIM, there is no indication that an ENQ is being propagated correctly. Therefore, it is critical to define the QNAME SIS2000.

Note: Before bringing down MIM or GRS, you must first terminate all Advanced Backup and Recovery for z/OS programs.

Consider a MIM example in which you want to add the SIS2000 QNAME to MIM. Add the following statement to your MIM parameter library member. Be aware that this is only an example; the version of MIM you are using might require different statements.

```
SIS2000   GDIF=YES,SCOPE=SYSTEMS,EXEMPT=NO,ECMF=NO,
          RPTAFTER=0,RPTCYCLE=60,TRACE=NONE
```

Now consider a GRS example in which you want to add the SIS2000 QNAME to GRS. Add the following statement to your GRS parameter library member. This is only an example; the version of GRS you are using might require different statements.

```
RNLDEF RNL(CON) TYPE(GENERIC) QNAME(SIS2000)
```

Task 2: Verifying that the configuration values were set in BKMINI

Examine the product parameter library member BKMINI to verify that all the Advanced Backup and Recovery for z/OS configuration values have been appropriately set.

If you make changes to BKMINI, remember to run BKMJCLBL afterward.

Task 3: Provide access authority

You need to provide access to the Advanced Backup and Recovery for z/OS data sets.

If you are using Advanced Backup and Recovery for z/OS on a trial basis, you might have already completed this task.

1. Provide access authority to Advanced Backup and Recovery for z/OS data sets.

   To provide information for constructing resource security rules for Advanced Backup and Recovery for z/OS data sets for end-user usage, users need the required access authority to the product data sets.
2. Provide access authority to non-product data sets, jobs, and users.
3. Set the level of authority required by specific jobs or users to access certain data sets or perform certain functions.

   A profile is required for Top Secret and ACF2.

Task 4: Adding BKMWABAR to the TMOEDMxx file

If you plan to use the ABARS Manager to monitor ABARS backups and recoveries, use CA1 for tape management, and use an External Data Manager (EDM), you need to add the BKMWABAR program name to the TMOEDMxx member of the CA 'ppoption' file.

The TMOEDMxx member should look similar to the following example:

```
********** TMOEDM00 **********
EDM=HSM1,PGM=ARCCTL,DSN=-
EDM=HSM1,PGM=BKMWABAR,DSN=-
```

Task 5: Replacing the DFHSMABR procedure

The IBM standard DFHSMABR procedure executes program ARCWCTL. If you plan to use the ABARS Manager to monitor ABARS backups and recoveries, you need to replace DFHSMABR with the Advanced Backup and Recovery for z/OS BKMABARS procedure, which invokes the product and ABARS process through program BKMWABAR.

This change results in Advanced Backup and Recovery for z/OS being invoked for every ABARS function in the installation, even those using only native ABARS functions. Using a very short and reliable code path, Advanced Backup and Recovery for z/OS recognizes native ABARS functions and bypasses itself, giving control directly to ABARS for those requested native ABARS activities. This technique provides maximum value to you, and avoids changes to ABARS itself, ensuring reliable and supportable ABARS processing in all circumstances.

BKMABARS is shipped in the product JCL library, and is compatible with both native ABARS and ABARS Manager.

1. Copy the JCL library's BKMABARS member into a system procedure library.
2. Ensure that the **LOAD1** and **PARMLIB** parameters are correct in the BKMABARS member PROC statement.
3. Add the BKMABARS procedure library to the RACF® started task table.
4. Give the BKMABARS task the same RACF authority as the IBM DFHSMABR task.

Task 6: Modifying member ARCCMDxx in the DFSMShsm parameter library

If you plan to use the ABARS Manager to monitor ABARS backups and recoveries, you need to customize the ABARS environment by modifying the DFSMShsm parameter library member ARCCMD*xx* (where *xx* is your parameter library member suffix).

Advanced Backup and Recovery for z/OS provides sample ARCSKEXT and ARCCREXT exit modules in the product load library, under the names BKMSKEXT and BKMCREXT. The Advanced Backup and Recovery for z/OS versions of the exits are loaded automatically when Advanced Backup and Recovery for z/OS executes.

Local installation versions of these exits are also supported. The Advanced Backup and Recovery for z/OS versions of these exits receive control first, then provide control to the entry point of the local version of each exit, if provided.

The ABARS backup error installation exit (ARCBEEXT) allows you to skip a data set when DFSMSdss is dumping level zero DASD data sets in the INCLUDE list. For more information, consult the DFSMShsm manuals.

Advanced Backup and Recovery for z/OS provides a sample ARCBEEXT. The sample source and installation JCL are shipped as members BKMBEEXT and BKMBJCL, respectively, of the *hlq*.SBKMASMP library. The sample exit code is set to bypass all errors by providing a return code of four. The Advanced Backup and Recovery for z/OS ARCBEEXT also issues WTOs that can be removed or customized.

1. Change the **ABARSPROCNAME(xxxxxxxx)** keyword to ABARSPROCNAME(BKMABARS).
2. For installations that operate under DFSMShsm version 1.4 and higher, use the DFSMShsm **SETSYS** command in the DFSMShsm parameter library to direct the activity logs to DASD: SETSYS ABARSACTLOGTYPE(DASD).

   Advanced Backup and Recovery for z/OS obtains information about the aggregate backups and recoveries that have taken place by processing the ABARS activity logs and then storing that information in the IDS. Advanced Backup and Recovery for z/OS cannot process those logs unless they have been directed to DASD.
3. Set the **ABARSACTLOGMSGLVL** parameter to FULL.

   If **ABARSACTLOGMSGLVL** is set to REDUCED, data set names are not written to the ABARS activity logs, which means Advanced Backup and Recovery for z/OS cannot find the data set information.
4. In member ARCCMDxx, activate the ARCSKEXT exit, which is used for selective data set restoration and incremental backups, and the ARCCREXT exit, which is the aggregate recovery conflict resolution exit used for **GDGREPLACE**. Use IBM's exit names for the **SETSYS EXITON** command.

   Do not copy the Advanced Backup and Recovery for z/OS versions of these exits to your SYS1.LINKLIB.
5. (Optional) If you want to use the Advanced Backup and Recovery for z/OS version of ARCBEEXT, do the following:
   a) Use the ARCBEEXT in member BKMBEEXT of the SBKMASMP library as it is, or modify it as appropriate.
   b) Use member BKMBJCL of the SBKMASMP library to assemble and link edit the code into your link library.
   c) If the ARCBEEXT exit has not already been activated, activate the exit by updating your HSM parameter library member ARCCMDxx.

The following figure shows an example of the DFSMShsm **SETSYS** command that activates the ARCBEEXT, ARCSKEXT and ARCCREXT exits:

```
/*************************************************************/
/*              DFHSM EXITS                                  */
/*************************************************************/
   SETSYS -
   EXITON(ARCBEEXT ARCSKEXT ARCCREXT)
```

Task 7: Updating the ACF$CMD system security command table

If you have additional security, such as ACF2, that restricts execution of non-authorized TSO commands, you need to add the appropriate commands as allowable in the ACF$CMD system security command table.

- Add the appropriate commands to the ACF$CMD system security command table:
  - BKMMAIN1
  - BKMCMD
  - **AMPQSCAN**
  - **AMPRAC01**

Task 8: Setting up ABARS Monitor

If you plan to use the ABARS Manager to monitor ABARS backups and recoveries, you need to set up the ABARS Monitor.

1. Add module BKM01IKB to the AUTHTSF table of SYS1.PARMLIB(IKJTSOnn).
2. Ensure TSO profile options **WTPMSG** and **MSGID** are turned on by performing these steps:
   a) Type TSO PROFILE LIST at the **Option** prompt.
   b) If necessary, type TSO PROFILE MSGID WTPMSG at the **Option** prompt.

Task 9: Setting up the SDSL VOL option

Because the **SDSL VOL** option collects volume data set information through the IDCAMS DCOLLECT function, you must specify IDCAMS in the SYS1.PARMLIB(IKJTSOnn) member's AUTHTSF table. This is an optional task.

Task 10: Loading the demonstration aggregates

This is an optional task. If you requested the demonstration aggregates file from Technical Support, you need to load it.

The *hlq*.DEMOAGGS.UNLOAD file was sent in binary format by email as demoaggs.bin

1. Use binary format to upload demoaggs.bin to a sequential mainframe file with these attributes: RECFM=FB, LRECL=80, and BLKSIZE=3120.
2. At the TSO READY prompt, type receive indataset('bkm.demoaggs.bin').

```
receive indataset('BKM.demoaggs.bin')
 INMR901I Data Set BKM.DEMOAGGS.UNLOAD from DS01 on NODENAME
 INMR906A Enter restore parameters or 'DELETE' or 'END' +
 <Enter>
 INMR001I Restore successful to data set 'DS01.DEMOAGGS.UNLOAD'
   READY
```

3. After the file successfully uploads, issue the **BKM LOADIDS** command.

Task 11: Adding security for primary and line commands

If you want to control user access to Advanced Backup and Recovery for z/OS primary and line command execution, you need to specify the appropriate SAF protections.

This is an optional task.

Advanced Backup and Recovery for z/OS controls user access to primary and line commands through SAF FACILITY profiles that are recognized by RACF, ACF2, and Top Secret. If you do not specify any SAF protections, all users are granted access to all Advanced Backup and Recovery for z/OS primary and line commands.

Note: A profile is required by Top Secret and ACF2.

Other functions are covered by your current SAF environment, such as the **ES** line command.

Task 12: Defining security class profiles for CATSCRUB

If you want to control user access to the Advanced Backup and Recovery for z/OS **CATSCRUB** command, you need to define an SAF security class profile with the appropriate protection.

Advanced Backup and Recovery for z/OS uses SAF security class profiles to control access to the potentially dangerous **CATSCRUB** command. These profiles are recognized by RACF, ACF2, and Top Secret. The profiles are established at the command level, which gives you maximum flexibility in controlling who is allowed to run the **CATSCRUB** command.

Task 13: Editing the FDR_LOADLIB token in BKMINI

If you use FDR to back up data and plan to track the backups using Critical Backup Tracking and Inventory, you need to edit the **FDR_LOADLIB** token in BKMINI.

In the :PRODUCT_DATASETS section of the Advanced Backup and Recovery for z/OS product parameter library's BKMINI member, code the **FDR_LOADLIB** token LINKLIST to indicate that the FDR load library is in the LINKLIST. If the FDR load library is not in the LINKLIST, specify the data set name of the FDR load library as shown in the following figure.

```
/*********************************************************************/
/* THIS SECTION DEFINES THE DATASETS TO BE USED BY BKM SUITE. NAMES   */
/* ARE USER CHOSEN.  NAMES ARE FULLY QUALIFIED WITHOUT QUOTES.       */
/*********************************************************************/
:PRODUCT_DATASETS
FDR_LOADLIB  = SYS1.OUR.FDR.LOADLIB
```

Task 14: Editing the ADABAS_LOADLIB token in BKMINI

If you use Adabas from Software AG to back up data, and you plan to track the backups using Critical Backup Tracking and Inventory, you need to modify the product parameter library's BKMINI member's **ADABAS_LOADLIB** token.

Code the **ADABAS_LOADLIB** token LINKLIST to indicate that the Adabas load library is in the LINKLIST. If the Adabas load library is not in the LINKLIST, specify the data set name of the Adabas load library as shown below.

```
/*********************************************************************/
/* THIS SECTION DEFINES THE DATASETS TO BE USED BY BKM SUITE. NAMES   */
/* ARE USER CHOSEN.  NAMES ARE FULLY QUALIFIED WITHOUT QUOTES.       */
/*********************************************************************/
:PRODUCT_DATASETS
ADABAS_LOADLIB  = SYS1.OUR.ADABAS.LOADLIB
```

Task 15: Defining the security class profiles for the Critical Backup Tracking and Inventory started task

If you have specified FACILITY profile for CSVDYLPA.ADD.*, provide UPDATE access to allow the Critical Backup Tracking and Inventory started task to load its Advanced Backup and Recovery for z/OS modules into DLPA. If profile CSVDYLPA.ADD.* has not been specified, nothing needs to be done.

Task 16: Specifying the SMF/JCL collection method

Automated Critical Data Identification uses system SMF and JCL data to determine which data sets belong to a particular application.

SMF data can be collected using either the batch Automated Critical Data Identification SMF scan program or the RSP IEFU83 exit, both of which require that record types 14, 15, 61, 64, 65, and 66

be recorded. If you are using the application cycle control function, SMF record type 30 is also required. This might require a change to member SMFPRMxx of SYS1.PARMLIB.

If you want to use either or both of the Automated Critical Data Identification features requiring the recording of SMF records, but do not want certain record types actually written to the log, contact Technical Support for an additional IEFU83 exit which allows IEFU83 exits to examine given record types but then suppresses the recording to the SMF log.

Note: If you are using ABARS and select the ABARS format, Automated Critical Data Identification categorizes the data sets into one of the three ABARS categories for Selection Data Set data: INCLUDE, ACCOMPANY, or ALLOCATE.

Task 17: Providing access authority to the Automated Critical Data Identification data sets

If you are using Advanced Backup and Recovery for z/OS on a trial basis, you might have already completed this task. Otherwise, you need to provide access to the Advanced Backup and Recovery for z/OS data sets.

1. Provide access authority to Advanced Backup and Recovery for z/OS data sets.

   To provide information for constructing resource security rules for Advanced Backup and Recovery for z/OS data sets for end-user usage, users need the required access authority to the product data sets.
2. Provide access authority to non-product data sets, jobs, and users.
3. Set the level of authority required by specific jobs or users to access certain data sets or perform certain functions.

   A profile is required for Top Secret and ACF2.

Task 18: Setting up foreground verification

You must add a module to the AUTHTSF table to enable foreground verification.

- Add module BKM01IKB to the AUTHTSF table of SYS1.PARMLIB(IKJTSOnn).

Task 19: Verifying that the Automated Critical Data Identification Selection Data Sets are physical sequential

Automated Critical Data Identification Selection Data Sets must be sequential files.

- Automated Critical Data Identification updates the Selection Data Set. Due to the nature of PDSs and PDSEs, two or more members cannot be written to at the same time. You must verify that the Selection Data Sets are physical sequential files.

Task 20: Turning off PF key function display

Some Automated Critical Data Identification panels require that **PFSHOW** be turned off.

If you cannot see part of a Automated Critical Data Identification panel because the PF key functions are displayed, type PFSHOW at the ISPF Command line to toggle the PF key function display.

Task 21: Editing and running the BKMDB2BG job

If you use IBM's DB2 and plan to use Advanced Backup and Recovery for z/OS to report on Db2 image copies, for each of your Db2 subsystems, you need to edit *prefix.value*.SBKMJCL(BKMDB2BG) appropriately and submit it for execution.

Instructions for editing the job appear in the comments at the top of the member.

Note: You may leave the default value (BKMPLAN1) for the plan name, or you may use your own plan name.

Task 22: Modifying SBKMASMP library member BKMTAPUX

BKMTAPUX contains sample code that must be tailored to operate correctly in your environment.

The BKMTAPUX member contains sample code for interfacing with your tape management system to provide or change the information for the tape data sets used at your installation. You can use the BKMTAPUX exit as a standalone tape manager instead of some other tape management system.

BKMTAPUX attributes are as follows:

- It must be reentrant and link edited into the product load library.
- This exit is called after each Critical Backup Tracking and Inventory call to the tape manager.
- On return from the exit, registers 2–13 must contain the same elements they contained upon entry, and the exit must place a return code in register 15.
- All communication with Critical Backup Tracking and Inventory is done through control block BKMTUXCB (refer to *prefix.value*.SBKMASMP).
- Any data that is to be changed or returned to the product should be placed into the appropriate field of the control block.
- Critical Backup Tracking and Inventory replaces the original metadata with that returned by the exit.
- Critical Backup Tracking and Inventory performs no return code checking, so it is the exit that must issue any necessary messages.

When BKMTAPUX takes control, the register contents are as follows:

- Register 1 contains the address of the BKMTUXCB control block.
- Register 13 contains the address of a standard 72–byte register save area.
- Register 14 contains the return address.
- Register 15 contains the entry point address.

A tape manager call session consists of one or more data calls, plus a last call. When the first data call for a session is made, control block field **TMUUSERWD** contains zeroes. During the session, for each object tape data set, the supported tape manager (if any) is queried for the metadata. All extracted data is sent to the tape exit through the control block. The data returned by the exit replaces the original data. When there are no more tape data sets for a session, Critical Backup Tracking and Inventory makes a last call to the exit (see **TMUFLG1/TMULCAL**).

Note:

- Control block field **TMUUSERWD** is reserved for use by the exit and it is not modified by Critical Backup Tracking and Inventory.
- If the exit will not be used again during a session, set the flag **TMUFLG1/TMUNMC** in the control block. When this flag is set, there is no termination call, so any clean up must be done prior to returning to Critical Backup Tracking and Inventory.

Step 13: Setting up the Advanced Backup and Recovery for z/OS started tasks

After you have installed Advanced Backup and Recovery for z/OS, configured it, and performed the post-installation tasks, you need to set up the Critical Backup Tracking and Inventory and Automated Critical Data Identification started tasks.

About the Critical Backup Tracking and Inventory started task

The Critical Backup Tracking and Inventory started task intercepts backups performed during backup execution and provides a method to store important information about the tracked backups in the Advanced Backup and Recovery for z/OS Inventory Data Set (IDS).

Each execution of a backup job that is tracked is recorded in the IDS by date and time. Backups are tracked by backup task within a job step. When they are no longer valid, backup inventory records are removed from the IDS by the BKMEXPIRE command found in the product JCL library's BKMEXPIR member. You need to set this job up and schedule it to execute frequently.

The IDS contains tracked backup job information, such as job name, step name, utility name, backup date and time, return codes, input and output data set names, and so on. The data is viewable via ISPF and allows dynamic recovery of full volume or selected data sets.

Critical Backup Tracking and Inventory functionality is supported by these components:

- The BKMSTSKA started task
- The required job initiation exit IEFUSI
- A batch intercept file (BIF)
- Tracking modules for supported backup utilities
- The Advanced Backup and Recovery for z/OS IDS
- ISPF interface
- Batch utilities

The Critical Backup Tracking and Inventory started task does the following:

- Builds an in-storage job mask table from the batch intercept file
- Installs the IEFUSI exit
- Sets up communication with other participating systems and ISPF users
- Attaches a task to find and process 'spun off' data sets from tracking modules

Critical Backup Tracking and Inventory tracks the following backup utilities:

- ABARS backups executed through ABARS Manager
- Adabas (ADARUN)
- Brightstor CA-FAVER (GVEXPORT)
- CA-Disk (ADSST001)
- Catalog RecoveryPlus (CR+) backups
- CFCAMS (REPRO and EXPORT)
- DFSMSdss (full volume, physical dump by data set, logical backups, and Unix files backup)
- DFSMShsm ARCINBAK, AUTODUMP, and Incremental Backup

  Note: AUTODUMP is supported through a batch utility.
- FDR (FDR, FDRABR, FDRCPK, FDRDSF)
- IBM Tivoli Advanced Catalog Management backups
- ICEGENER
- ICEMAN
- ICETOOL
- IDCAMS (REPRO and EXPORT)
- IEBCOPY
- IEBGENER
- SORT
- SYNCGENR
- SYNCSORT
- TAPECOPY (OTTC0003)
- VSAM Assist (VSAMASST)

Note:

- Typically, SORT is used to sort data. Sorts of this kind are not considered backups and should *not* be tracked by Critical Backup Tracking and Inventory. The function of Critical Backup Tracking and Inventory is to track data set backups. However, Critical Backup Tracking and Inventory tracks SORT because some installations might have jobs that sort data to an output file that they do consider to be a backup. These are valid backups. Ensure that discrete job names are specified in the backup tracking INCLUDE filters instead of masks for any SORT job you want to track, to prevent tracking sort output files that are not backups, or to prevent failures. SORT functions often use PDS members as input to the SORT; however, Critical Backup Tracking and Inventory requires you to use the full partitioned data set name. Member names are not supported.
- Critical Backup Tracking and Inventory does not support backups where the data sets backed up are defined by concatenated input files. For example, an IEBGENER step where the SYSUTI1 DD uses concatenated files is not supported by Critical Backup Tracking and Inventory.

About the BKMSTSKD started task

BKMSTSKD is used by both ABARS Manager (Incremental ABARS feature only) and Critical Backup Tracking and Inventory to track new or changed data. It was installed as you completed the configuration tasks.

The BKMSTSKD started task must be active to track new and changed data and to use Automated Critical Data Identification's application backup and restore feature when you want to use the BKUPEND program's **CHANGED-SINCE-LAST-BKUP-RC** parameter. This parameter tracks any data set identified as critical by Automated Critical Data Identification and has an associated backup in Critical Backup Tracking and Inventory, but has changed since the last backup.

About the BKMSTSKR started task

BKMSTSKR is used by Automated Critical Data Identification to collect JCL and SMF data as jobs execute.

The BKMSTSKR started task must be active to track application usage of data. For those application jobs registered to Automated Critical Data Identification, BKMSTSKR monitors data set usage and record data set usage statistics in the Automated Critical Data Identification data base. This information is then used to construct the critical data set list at the end of the application cycle.

About installing the started tasks on multiple systems

If you are installing Critical Backup Tracking and Inventory and its started task (BKMSTSKA) is to be used on multiple images that share DASD, you need just one IDS that is shared among images, and one each of the Critical Backup Tracking and Inventory files (specified during installation) that is shared among images. However, there must be a Critical Backup Tracking and Inventory started task on each image within the shared DASD configuration.

At a minimum, you need one Critical Backup Tracking and Inventory started task for each image on which you intend to track backups. The Critical Backup Tracking and Inventory started task is required on each image to enable tracking backups on the image the jobs execute.

There must be a BKMSTSKD (DAD) started task on each image within the shared DASD configuration from which you want to collect information. You need only the one Data Set Activity Database (DAD) that is shared among these images, and one each of the files that described in “Task 4: Creating the BKMSTSKD log data set” on page 41 and “Task 5: Creating the BKMSTSKD MSGQUEUE data set” on page 41. At a minimum, you need one DAD started task for each image from which you intend to track new or changed data. The DAD started task is required on each image to enable the tracking of new or changed data sets on the image the jobs execute.

If you are installing Automated Critical Data Identification and RSP on multiple images that share DASD, you need only the one Automated Critical Data Identification database that is shared among images. You also need one each of the files allocated and initialized in “Task 1: Allocating and initializing the RSP data sets” on page 39. However, there must be an RSP started task on each image (within the shared DASD

configuration) from which you want to collect JCL and SMF data. If the RSP tasks are up and running on each image, it does not matter to which image the job scheduler directs the job. At the end of the application, BKMAPLEN is executed to marry the JCL and SMF information from all images in date and time stamp order to ensure that the data is presented accurately.

To set up the Advanced Backup and Recovery for z/OS started tasks, perform these tasks:

Task 1: Allocating and initializing the RSP data sets

Run the BKMRSPSU job to allocate and initialize the files used by RSP.

Be aware of the following facts as you complete this task:

- The RSP Message Log data set records status information and is used as a tracing and debugging tool by the Automated Critical Data Identification RSP control program.
- The Job Flat data set is used to update the RSP job table.
- The Message Queue data set holds RSP messages and statistics.

1. Use the ISPF editor to inspect and edit (if necessary) the BKMRSPSU member of the Advanced Backup and Recovery for z/OS JCL library:
   a) Verify that the MSGLOG DCB attributes are `DSORG=PS`, `RECFM=F`, `LRECL=256`, `BLKSIZE=256`.
      Do not change the DCB characteristics.
   b) Allocate the MSGLOG data set with a minimum of 5 cylinders of primary space. A secondary space allocation is not required.
   c) Verify that the MSGLOG data set name matches the `BKMINI` **`LOG`** token value in section `:ACDI_RSP`.
   d) Verify that the JOBFLAT DCB attributes are `DSORG=PS`, `RECFM=VB`, `LRECL=255`, `BLKSIZE=0`. Do not change the DCB characteristics.
   e) Allocate the JOBFLAT data set with a minimum of 10 tracks of primary space and 2 tracks of secondary space.
   f) Verify that the JOBFLAT data set name matches the `BKMINI` **`JOBFLAT`** token value in section `:ACDI_RSP`.
   g) Verify that the RSPMSGQ DCB attributes are `DSORG-PS`, `RECFM=F`, `LRECL=4096`, `BLKSIZE=4096`. Do not change the DCB characteristics.
   h) Allocate the RSPMSGQ data set with a minimum of 8 tracks of primary space. A secondary space allocation is not required.

i) Verify that the RSPMSGQ data set name matches the BKMINI **RSP_MSGQUEUE** token value in section :ACDI_RSP.

2. Submit BKMRSPSU for execution.

The overflow data sets required by RSP are set up within the product parameter library's BKMINI member and are created dynamically when RSP is brought up. You need to specify appropriate values for their overflow parameters as you complete "Task 2: Editing the Advanced Backup and Recovery for z/OS parameter library member BKMINI" on page 40.

Task 2: Editing the Advanced Backup and Recovery for z/OS parameter library member BKMINI

The Advanced Backup and Recovery for z/OS product parameter library member BKMINI requires some post-configuration changes.

Note: If your DFSMShsm CDS data sets are using RLS, these values must be set in the BKMINI member of product parameter library as specified in the post-installation tasks.

1. Use the ISPF editor to modify the tokens in the :BKMSTSKA section of BKMINI.

   Note: If you plan to use the Critical Backup Tracking and Inventory flat file processing function, make the appropriate value specifications for the **CBTI_FLATFILE_OPTION**, **CBTI_FLATFILE_VSAMDS**, and **AUTO_PROCESS_FLAT_FILES** options.
2. Edit the tokens in the :CBTI_DUMP_BACKUP_PROCESS_OPTIONS section of BKMINI.
3. Add the DAD data set to the **DAD_DATASET** token in the :PRODUCT_DATASETS section.
4. Set all keyword values in the :BKMSTSKD_OPTIONS section.
5. Edit the overflow parameters in the :BKMSTSKD_OPTIONS section of BKMINI.

   There must be one overflow data set for each instance of BKMSTSKD. Each overflow data set must include a SYSID value that specifies the SMF system ID on which the task is to run.
6. Edit the overflow parameters in the :ACDI_RSP section.

   There must be one overflow data set for each instance of BKMSTSKR. Each overflow data set must include a SYSID value that designates the SMF system ID on which the task is to run.
7. Save your changes to BKMINI.

Task 3: Allocating and initializing the BKMSTSKA data sets

The message log, message queue, and batch intercept data sets used by the BKMSTSKA started task must now be allocated and initialized.

The Message Log data set is used to record informational, warning, and error messages generated by the BKMSTSKA started task. The setup job DD name for the Critical Backup Tracking and Inventory Message Log data set is MSGLOG. Only one MSGLOG is required.

The Message Queue data set is used to communicate between participating systems and ISPF users. The setup job DD name for the Critical Backup Tracking and Inventory Message Queue data set is MSGQUEUE. Only one MSGQUEUE is required.

The Batch Intercept File (BIF) data set is a table that the started task BKMSTSKA uses to determine which backup jobs and programs to track. The BIF consists of job names, program, PROC and step names entered into a Critical Backup Tracking and Inventory panel. The setup job DD name for the Critical Backup Tracking and Inventory Batch Intercept File is BIF. Only one BIF is required.

1. Using the ISPF editor, edit the JCL library member BKMSETUP to allocate the files used by BKMSTSKA.
2. For the MSGLOG data set, specify DSORG=PS, RECFM=F, LRECL=256, BLKSIZE=256, and specify a minimum of 5 cylinders of primary space and zero tracks of secondary space.
3. For the MSGQUEUE data set, specify DSORG=PS, RECFM=F, LRECL=4096, BLKSIZE=4096, and specify a minimum of 2 cylinders of primary space and zero tracks of secondary space.

4. For the BIF data set, specify DSORG=PS, RECFM=FB, LRECL=251, BLKSIZE=0, and specify a minimum of 10 tracks of primary space and 2 tracks of secondary space.
5. Save your changes to BKMSETUP and submit it for execution.

Task 4: Creating the BKMSTSKD log data set

You need to create a log data set that BKMSTSKD maintains and updates with event and diagnostic information.

The log file is a wraparound file used by product development and is a required file that is allocated through supplied JCL members. You need one BKMSTSKD started task on each image from which you intend to capture data set usage information. All instances of BKMSTSKD share a single log file.

- To create the log file, modify and execute the BKMDLOG member of the Advanced Backup and Recovery for z/OS JCL library.

Task 5: Creating the BKMSTSKD MSGQUEUE data set

The MSGQUEUE data set that BKMSTSKD uses for communication between participating systems and ISPF users must be created.

There is one MSGQUEUE data set for all the Incremental ABARS BKMSTSKD started tasks that update the same DAD (Data Set Activity Database).

- To create the MSGQUEUE file, modify and execute the BKMDMSGQ member of the Advanced Backup and Recovery for z/OS JCL library.

Task 6: Modifying BKMSTSKA and adding it to a system procedure library

The JCL library's BKMSTSKA member must be reviewed and copied to a system procedure library.

1. Verify that the symbolic parameter values are correct.
2. Save your changes to BKMSTSKA.
3. Copy BKMSTSKA to the SYS1.PROCLIB on each image where you intend to run the started task. You can use a default procedure library other than SYS1.PROCLIB, but if you specify a non-default library, you must specify the target procedure library in the start command when you start the task.

   Note:

   If you want to override the BKMINI **CBTI_HARDCOPY_MSGS_ONLY** value for specific LPARs, you can use parameter specifications in the BKMSTSKA procedure for the specific LPARs.

   Example 1:

   Consider an example in which you want the intercept log and intercept messages in JESMSGLG on all LPARS for intercepted job steps. You can set the BKMINI token **CBTI_HRDCPY_MSGS_ONLY**=N, and then use the S BKMSTSKA command to start the started task.

   Example 2:

   Consider a different example in which you do not want the intercept log and intercept messages in JESMSGLG on any LPAR for intercepted job steps. You can set the BKMINI token **CBTI_HRDCPY_MSGS_ONLY**=Y, and then use the S BKMSTSKA command to start the started task.

   Example 3:

   Consider a third example in which you want the intercept log and intercept messages on all but one LPAR for intercepted job steps. You can set BKMINI token **CBTI_HRDCPY_MSGS_ONLY**=N and then use the S BKMSTSKA,PARM='HCLON' command to start the started task on that one specific LPAR.

   Example 4:

   Finally, consider an example in which you want the intercept log and intercept messages in JESMSGLG on only one specific LPAR for intercepted job steps. You can set the BKMINI token

CBTI_HRDCPY_MSGS_ONLY=Y and then use the S BKMSTSKA,PARM='HCLOF' command to start the started task on that one specific LPAR.

Task 7: Modifying BKMSTSKD and adding it to a system procedure library

The JCL library's BKMSTSKD member must be copied and added to a system procedure library.

1. Modify the BKMSTSKD procedure and change the product parameter library and LOAD1 to match the product load library data set name chosen during installation.
2. Although you can test BKMSTSKD as a submitted job, it is typically a started task. Copy the BKMSTSKD member to a procedure library and take whatever steps are necessary to ensure that the task is always running.

   BKMSTSKD uses CSVDYNEX, which already includes a retry mechanism; therefore, it is not recommended that the BKMSTSKD task be automatically restarted if it terminates abnormally.

Task 8: Modifying BKMSTSKR and adding it to a system procedure library

The JCL library's BKMSTSKR member must be modified and copied to a system procedure library.

1. To the Advanced Backup and Recovery for z/OS JCL library's BKMSTSKR member, add your LOAD1 and product parameter library names.
2. Save your changes to BKMSTSKR.
3. Copy BKMSTSKR to the SYS1.PROCLIB on each image where you intend to run the started task. You can use a default procedure library other than SYS1.PROCLIB, but if you specify a non-default library, you must specify the target procedure library in the start command when you start this task.

Task 9: Enabling the required exits in SMFPRMxx

The IEFUSI, IEFU83, IEFU84 (optional), and IEFUJI user exits in the SMFPRMxx member of SYS1.PARMLIB must be enabled prior to use.

A sample SMFPRMxx member is shown in the following figure; this is an example member only–your SMFPRMxx member might look different.

Note: For MVS SP5.n and higher, the U83, U84, and UJI exits are installed and removed by the BKMSTSKR task using the CSVDYNEX feature.

```
DATA SET: SYS1.PARMLIB                                  MEMBER: SMFPRMXX
DATE: 2003/11/10          TIME: 16:55                         PAGE:   1

ACTIVE                           /*ACTIVE SMF RECORDING*/
DSNAME(SYS1.MANA, SYS1.MANB)     /*TWO DATA SETS MANA AND MANB */
NOPROMPT                         /*DO NOT PROMPT THE OPERATOR FOR OPTIONS*/
REC(PERM)                        /*TYPE 17 ALL RECORDS*/
MAXDORM(3000)                    /*WRITE AN IDLE BUFFER AFTER 30 MIN*/
STATUS(010000)                   /*WRITE SMF STATS AFTER 1 HOUR*/
JWT(0030)                        /*522 AFTER 30 MIN*/
SID(SYA1)                        /*SYSTEM ID*/
LISTDSN                          /*LIST DATA SET STATUS AT IPL*/

SYS(NOTYPE(5,19,25,32,34,36,37,40,41,49:50,60,128,169,253),
      EXITS(IEFU83,IEFU84,IEFACTRT,IEFUJV,IEFUJI,IEFUSI,IEFUTL),
      INTERVAL(010000),NODETAIL)
/*  FOR BATCH JOBS THE TYPE 0 INTERVAL RECORDS WILL BE RECORDED EACH
    HOUR. THIS WILL PROVIDE THE MEANS TO ALLOCATE RESOURCE USAGE FOR
    LONG-RUNNING JOBS ON AN HOURLY BASIS.  */

SUBSYS(TSO,
    NOTYPE(5,19,25,32,34,36,37,40,41,49:50,57,60,128,169,253),
    EXITS(IEFU83,IEFU84,IEFACTRT,IEFUJV,IEFUJI,IEFUSI,IEFUTL),
    INTERVAL(010000),NODETAIL)

/*  FOR TSO WRITE RECORD TYPE 30 BUT DO NOT PROVIDE INTERNAL
    RECORDS. THIS MEANS THAT IN THE CASE OF A SYSTEM FAILURE
    ALL RESOURCE UTILIZATION WILL BE LOST.  */

SUBSYS(STC,
    NOTYPE(5,19,25,32,34,36,37,40,41,49:50,57,60,128,169,253),
    EXITS(IEFU29, IEFU83,IEFU84),INTERNAL(01000))

/*  WRITE SELECTED RECORDS AS PER PARMS, TAKE ONLY THREE EXITS,
    NOTE: IEFU29 EXECUTES IN THE MASTER ASIS WHICH IS A STC ADDRESS
    SPACE SO IEFU29 MUST BE ON FOR STC. */

SUBSYS(HSC0,INTERVAL(000500),TYPE(254))   /* NEARLINE */
SUBPARM(HSC0(SUBTYPE(1,2,3,4,5,6,7,8)))
SUBPARM(IXFP(204,SRP,DDSR))               /* ICEBERG */
```

1. Define the default exit IEFUSI in the EXIT list for SYS and subsystems.

   This allows Critical Backup Tracking and Inventory to install its IEFUSI exit dynamically.
2. Define the IEFU83 and IEFU84 exits in the EXIT list for SYS and subsystems, specifying that record types 14, 15, 30, 61, 64, 65, and 66 are to be included.
3. Define IEFUJI exit in the EXIT list for SYS and subsystems.
4. Verify that the **TYPE** or **NOTYPE** parameter allows these record types: 14, 15, 30, 61, 64, 65, and 66.
5. If this is the first time the IEFUSI exit has been enabled in this member, refresh the product parameter library to recognize the enablement.

Task 10: Building the JOBFLAT file

The job contained in the BKMBLDJB member of the JCL library creates the JOBFLAT file that BKMSTSKR task uses to hold the job names that were extracted from its database.

- Submit the JCL library's BKMBLDJB member for execution.

Task 11: Starting BKMSTSKA

The next step in setting up the started tasks is to start up the BKMSTSKA task.

The Advanced Backup and Recovery for z/OS product parameter library's BKMINI member is shipped with CBTI_AUTOSTART = YES. If this parameter is set to YES, issue the following command to start the BKMSTSKA started task: S BKMSTSKA,JOBNAME=CBTI (you can substitute a different job name, if you want).

Note: It is strongly recommended that you leave CBTI_AUTOSTART set to YES.

If this parameter in BKMINI parameter has been set to NO, you must issue the following two commands to start the task and its subtasks:

```
S BKMSTSKA,JOBNAME=CBTI

F CBTI,AUTOSTART
```

In general, **AUTOSTART** is the only command needed to start BKMSTSKA. See member BKMSTSKA in the Advanced Backup and Recovery for z/OS JCL library. The member and procedure name is BKMSTSKA and the started task name is CBTI. All modify commands are based on the identifier CBTI.

AUTOSTART issues the following commands:

```
F CBTI,LOAD
F CBTI,TURNON
F CBTI,STARTMON
F CBTI,STARTFC
```

Task 12: Starting BKMSTSKD

The BKMSTSKD started task JCL must be modified for each image where you intend to run BKMSTSKD, and then the task must be started.

All executions of BKMSTSKD on each image use the same DAD database and the same log file. When starting the BKMSTSKD started task, the target procedure library member name must be specified on the start command. It is recommended that a **JOBNAME=** parameter be placed after the member name.

The BKMSTSKD program installs an IEFU83 exit dynamically under MVS 5 and higher, using the CSVDYNEX facility. This facility does not require the IEFU83 program name to be specified in SYS1.PARMLIB member SMFPRMxx. However, it does require that IEFU83 (the IBM 'dummy' exit routine) be specified for SYS and SUBSYS exits for which SMF data is to be collected.

The installation must be collecting SMF 14, 15, 61, 64, 65, and 66 records. Verify that the **TYPE** or **NOTYPE** keyword values allow the generation of these record types. An additional general purpose IEFU83 exit is available for users who wish to suppress the actual logging of these record types, or in fact record types needed by any other IEFU83 exits that are not desired in the log.

Note: Because the **AUTOSTART** option is generally used, the full set of BKMSTSKD operator commands are not documented here.

1. Copy the BKMSTSKD member to each system's SYS1.PROCLIB.
2. Specify the target procedure library member name as part of the start command:
   - Issue the S BKMSTSKD,JOBNAME=DAD,PARM=AUTOSTART command (where the assumption is that the procedure library member name is BKMSTSKD and that DAD is the identifier). You can make the job name unique for each image, which makes it easy to stop the DAD task on any one image, if necessary.
   - You can change the BKMSTSKD JCL to include: //BKMSTSKD EXEC PGM=BKMSTSKD,PARM=AUTOSTART or //BKMSTSKD EXEC PGM=BKMSTSKD, PARM=A.
   - Alternatively, update BKMINI so that AUTOSTART=YES, which causes BKMSTSKD to assume that PARM=AUTOSTART. This is the preferred method of starting BKMSTSKD.

Task 13: Starting BKMSTSKR

You need to start the BKMSTSKR task.

Before you run RSP, include the BKMAPLEN procedure in the application job stream, either as the last job or as a step within the last job (if applications have been set up). If you do not include BKMAPLEN in the job stream, Automated Critical Data Identification collects data continuously but never cycles it out of the database, which can cause the database to fill up.

The **AUTOSTART** command that is used to start this task also issues the following commands:

```
F RSP,U83(LOAD)
F RSP,UJI(LOAD)
F RSP,DSPC(CREATE)
F RSP,U83(TURNON)
F RSP,UJI(TURNON)
```

```
F RSP,ACDIUPDT(START)
F RSP,OVFIMTSK(START)
F RSP,OVFEXTSK(START)
F RSP,BKMSTSKR(START)
```

1. If you set the **AUTOSTART** parameter to NO when you edited the product parameter library's BKMINI member, enter the following commands:S BKMSTSKR,JOBNAME=RSP (you can use a different job name if you want), and then F RSP,AUTOSTART.
2. If you set the **AUTOSTART** parameter to YES when you edited the product parameter library's BKMINI member, enter the following command: S BKMSTSKR,JOBNAME=RSP (or use a job name of your own choosing).
3. If you subsequently stop and start RSP and get an ABEND 201 message, restart RSP because its previous shutdown was abnormal.

Task 14: Tracking critical messages

If you want Advanced Backup and Recovery for z/OS to track continuously application job activity, data set changes, and backups, it is recommended that you monitor started task critical messages with an automated operations tool.

All Advanced Backup and Recovery for z/OS messages are documented in the *IBM Tivoli Advanced Backup and Recovery for z/OS: Messages* reference guide.

If you do not monitor these messages and take corrective action promptly, you might miss critical information, making application backup contents and backup inventory information incomplete.

Started task messages indicative of serious issues are the following:

```
BKMF0000I  TRACKING SUSPENDED - JOBNAME:nnn STEP NAME: sss.
BKMB2904E  ABNORMAL TERMINATION OF DADDBUPD
BKMB3202E  ALLOCATE OF OVERFLOW DATASET FAILED
BKMB3203E  OPEN OF OVERFLOW DATASET FAILED
BKMK1010E  ABNORMAL TERMINATION OF BKMRUPDT
BKMK3802E  ALLOCATE OF OVERFLOW DATASET FAILED
BKMK3803E  OPEN OF OVERFLOW DATASET FAILED
BKMK3806W  OVERFLOW DATASET IS FULL
BKMK0316W  DATABASE UPDATE TASK NOT ACTIVE FOR SYSID
```

Running BKMMODLV to validate the load module contents

You might be asked by Technical Support to run the BKMMODLV JCL to identify the current modification levels of the Advanced Backup and Recovery for z/OS load modules. BKMMODLV identifies and displays the PTFs and enhancements that have been applied to the product software, based on information stored internally within the load modules.

Be aware that only the most recent update contains the fix and revision number and the change date. All other fixes have the ID number only.

The following figure shows sample BKMMODLV JCL.

```
//BKMMODLV JOB 'MOD LEVEL RPT',CLASS=A                 >>> SEE #1
//*-------------------------------------------------------------------*/
//* 1. CHANGE JOB STATEMENT TO LOCAL STANDARDS                        */
//*-------------------------------------------------------------------*/
//*******************************************************************
//* JCLFIXES                                                        *
//* SAMPLE JCL FOR LISTING THE BUILD INFORMATION INCLUDED IN      *
//* LOAD MODULES.                                                   *
//*******************************************************************
//*
//BKMMODLV PROC LOAD1=BKM.SBKMLOAD
//       SOUT='*'
//BKM01MOD EXEC PGM=BKM01MOD
//STEPLIB  DD   DISP=SHR,DSN=&LOAD1
//SYSUT1   DD   DISP=SHR,DSN=&LOAD1
//SYSPRINT DD   HOLD=YES,SYSOUT=&SOUT
//         PEND
//*
//RUN      EXEC BKMMODLV
```

To edit the BKMMODLV JCL:

1. Edit the STEPLIB statement to specify the Advanced Backup and Recovery for z/OS load library.
2. Use the SYSPRINT statement to specify the output DD.
3. Use the SYSUT1 DD statement to specify the primary load library you want processed.
4. Execute the JCL.

   The resulting report shows the last fix and a list of fixes for all the modules in the SYSUT1 library.

The following figure shows an example of the BKMMODLV report.

```
Highest Build Date is 2015/02/17 09:18

Highest PMR Date is 2014/07/24 12:10  PMR01099
Highest ETR Date is 2013/05/14 11:44  ETR26736

Member   DD          Last fix  Vers   Rev    Build Date/Time  Macro

BKMADMUE SYSUT1 LAST BRM04890  2.3    13     2013/05/10 15:47 MSCHDR
         SYSUT1 LIST 09942,19302,20013,21482,16642,19970,23002,22701,
         SYSUT1 LIST 23101,22849,24720,26736,05093,04890

BKMAFILT SYSUT1 LAST ETR26736  2.3    04     2013/05/10 15:47 MSCHDR
         SYSUT1 LIST 19166,22041,17994,26736

BKMAGGMR SYSUT1 LAST ETR26736  2.3    06     2013/05/10 15:47 MSCHDR
         SYSUT1 LIST 13462,14367,21482,19970,25285,26736

BKMAGGTB SYSUT1 LAST BRM05142  2.3    05     2013/05/10 15:47 MSCHDR
         SYSUT1 LIST 13462,14367,21482,19970,05142

BKMAGINT SYSUT1 LAST ETR26736  2.3    02     2013/05/10 16:15 MSCHDR
         SYSUT1 LIST 21293,21482,26736

BKMAGMNG SYSUT1 LAST ABR02237  2.4    16     2014/04/11 09:28 MSCHDR
         SYSUT1 LIST 09388,09498,09471,10064,11179,11248,11236,11838,
         SYSUT1 LIST 14902,21482,19970,24720,25732,26697,26736,02237

BKMAGMST SYSUT1 LAST           1.04   0      2013/05/14 09:55 EHDR
         SYSUT1 LIST 9619,9742,10064,12127

BKMALZIP SYSUT1 LAST ETR26736  2.3    04     2013/05/14 11:16 MSCHDR
         SYSUT1 LIST 14367,21482,19970,26736

BKMAPI00 SYSUT1 LAST ABR00865  2.3    12     2013/05/14 11:16 MSCHDR
         SYSUT1 LIST 00214,00214,00214,00214,00492,00495,00438,00651,
         SYSUT1 LIST 00651,00651,00777,00865,00865

BKMAPI10 SYSUT1 LAST ABR00882  2.3    11     2013/05/14 11:16 MSCHDR
         SYSUT1 LIST 00214,00214,00214,00495,00651,00651,00651,00652,
         SYSUT1 LIST 00438,00777,00882

BKMAPI20 SYSUT1 LAST ABR00882  2.3    00     2013/05/14 11:16 MSCHDR
         SYSUT1 LIST 00882

BKMAPLEN SYSUT1 LAST BRM06588  2.3    43     2013/05/14 11:16 MSCHDR
         SYSUT1 LIST 21260,23580,17994,20989,25150,26951,26952,04446,
         SYSUT1 LIST 04514,04557,04580,26736,00336,01762,01096,01476,
         SYSUT1 LIST 06438,06588
```

Chapter 5. Completing a product upgrade

This topic provides instructions for completing the upgrade of an earlier version of Advanced Backup and Recovery for z/OS to the current version.

Before you begin the procedures that are provided in this chapter, you must have completed the procedures that were provided in the Chapter 3, “Starting a product installation or upgrade,” on page 19 chapter.

Note: The Fast Replication Services (FRS) feature requires some minimal configuration before it can be used. Instructions for completing the FRS configuration are provided in “Preparing to use FRS” on page 303.

To complete your product upgrade, perform the following steps:

- “Step 1: APF authorizing the product load library” on page 49
- “Step 2: Creating a new BKMINI parameter library member” on page 49
- “Step 3: Coding the tokens in the BKMINI parameter library member” on page 50
- “Step 4: Running BKMJCLBL” on page 50
- “Step 5: Updating system procedure libraries” on page 50
- “Step 6: Completing the post-upgrade tasks” on page 50

  To complete this step, you might need to complete several tasks.

Step 1: APF authorizing the product load library

The Advanced Backup and Recovery for z/OS load library must be APF authorized. This is a required task.

- Consult your systems programmer about having this library added into the APF list and ensuring that appropriate access controls have been established.

  Note: If your system parameter library member `IEASYS00` has parameter `LNKAUTH=LNKLST` specified, Advanced Backup and Recovery for z/OS ignores it. The product load library must be APF authorized.

Step 2: Creating a new BKMINI parameter library member

`BKMINI` keyword values are fetched one at a time when the product is started up. BKMIMERG either merges new and updated `BKMINI#` product parameter library data with your existing `BKMINI` product parameter library member, or, in the case of initial installs, this task creates a new `BKMINI` product parameter library member.

The JCL to run BKMIMERG is in the data set *prefix.value*`.SBKMJCL(BKMIMERG)`, which was created during the install process.

1. Provide a valid job card statement.
2. Change the `STEPLIB` data set name to the *prefix.value*`.SBKMLOAD` library.
3. Change the `INI` DD statement to point to the *previous.version*`.SBKMPARM`.

   Using the `SBKMPARM` from the previous product installation as input updates the new version of `BKMINI` with the parameter values from your previous version of the product.
4. Change the `MSCINI` DD statement to point to the *prefix.value*`.SBKMPARM` data set, and ensure that the member name remains as `BKMINI#`.
5. Change the `UPDATE` DD statement to point to the *prefix.value*.`SBKMPARM` data set, and ensure that the member name remains as `BKMINI`.
6. Submit the job, and ensure that it executes and completes with a completion code 0 before proceeding to the next task.

If the job terminates with a non-zero completion code, first check to see if any error messages indicate a situation that is easily correctable. BKMIMERG can be rerun. If the situation does not appear to be easily correctable, contact Technical Support for assistance before proceeding to the next task.

Step 3: Coding the tokens in the BKMINI parameter library member

Examine the product parameter library member BKMINI and verify that all the Advanced Backup and Recovery for z/OS configuration token values have been set appropriately.

Review the token setting options in the *BKMINI configuration values* appendix to determine if you need to change any of the token values for your installation.

Step 4: Running BKMJCLBL

Advanced Backup and Recovery for z/OS batch job JCL library members use instream PROCs. The instream PROCs contain symbolics (&...) that refer to various product data sets. BKMJCLBL substitutes BKMINI token values into the PROC statements in the JCL library members.

Even though distributed product maintenance may not include new JCL members, changes to the BKMINI product parameter library member would necessitate this step. BKMJCLBL is in the product JCL library.

1. Because you are updating into an existing JCL library, it is recommended that you back it up before running this job.
2. Modify the BKMJCLBL job JCL as indicated by the comments at the beginning of the member.
3. Be sure that the BKMINI tokens are set as desired before running BKMJCLBL. BKMJCLBL can be run again at any time.
4. Submit the BKMJCLBL job.

Step 5: Updating system procedure libraries

If you intend to run Advanced Backup and Recovery for z/OS out of a system procedure library, you must ensure that all necessary JCL members are copied into that procedure library.

- Copy the following JCL members into the system procedure library from which you intend to run Advanced Backup and Recovery for z/OS:
  - BKMABARS
  - BKMBPROC
  - BKMRPROC
  - BKMSTSKA
  - BKMSTSKD
  - BKMSTSKR
  - BKMFPROC

Step 6: Completing the post-upgrade tasks

You might need to perform some post-upgrade tasks before the product is completely ready for use. Many of these tasks were completed when the previous version of the product was installed at your site.

Review the following list and verify that all of the necessary post-upgrade tasks have been completed at your site. Detailed instructions for performing these tasks are provided in bkm_install/concept/bkmucon_inst “Step 12: Completing the post-installation tasks” on page 29.

1. Updating the cross-system ENQ lists.
2. Verifying that the configuration values were set in BKMINI.
3. Providing access authority for the product data sets,
4. Adding BKMWABAR to the TMOEDMxx file.

5. Replacing the DFHSMABR procedure.
6. Modifying member ARCCMDxx in the DFSMshsm parameter library.
7. Updating the ACF$CMD system security command table.
8. Setting up ABARS Monitor.
9. Setting up the **SDSL VOL** option.
10. Loading the demonstration aggregates.
11. Adding security for primary and line commands.
12. Defining security class profiles for CATSCRUB.
13. Editing the **FDR_LOADLIB** token in BKMINI
14. Editing the **ADABAS_LOADLIB** token in BKMINI
15. Defining the security class profiles for the Critical Backup Tracking and Inventory started task.
16. Specifying the SMF/JCL collection method.
17. Providing access authority to the Automated Critical Data Identification data sets.
18. Setting up foreground verification.
19. Verifying that the Automated Critical Data Identification Selection Data Sets are physical sequential data sets.
20. Turning off PF key function display.
21. Editing and running the BKMDB2BG job.
22. Modifying the SBKMASMP library's BKMTAPUX member.

Chapter 6. Completing a product upgrade

This topic provides instructions for completing the upgrade of an earlier version of Advanced Backup and Recovery for z/OS to the current version.

Note: If you followed the instructions in Chapter 2, “Starting a product installation or upgrade,” on page 15 to start an initial installation, skip this chapter and finish your initial installation by following the procedures that are provided in Completing a product installation.

Before you can complete your product upgrade, you must have completed the procedures that were provided in Chapter 2, “Starting a product installation or upgrade,” on page 15.

Note: The Fast Replication Services (FRS) feature requires some minimal configuration before it can be used. Instructions for completing the FRS configuration are provided in “Preparing to use FRS” on page 303.

To complete your product upgrade, perform the following steps:

APF authorizing the product load library

The Advanced Backup and Recovery for z/OS load library must be APF authorized. This is a required task.

- Consult your systems programmer about having this library added into the APF list and ensuring that appropriate access controls have been established.

  Note: If your system parameter library member `IEASYS00` has parameter `LNKAUTH=LNKLST` specified, Advanced Backup and Recovery for z/OS ignores it. The product load library must be APF authorized.

Creating a new BKMINI parameter library member

`BKMINI` keyword values are fetched one at a time when the product is started up. BKMIMERG merges new and updated `BKMINI#` product parameter library data with your existing `BKMINI` product parameter library member.

The JCL to run BKMIMERG is in *`prefix.value`*`.SBKMJCL(BKMIMERG)`, which was created when you completed the steps in the *Starting a product installation or upgrade* chapter.

1. Provide a valid job card statement.
2. Change the `STEPLIB` data set name to the *`prefix.value`*`.SBKMLOAD` library.
3. Change the `INI` DD statement to point to the *`previous.version`*`.SBKMPARM`.

   Using the `SBKMPARM` from the previous product installation as input updates the new version of `BKMINI` with the parameter values from your previous version of the product.
4. Change the `MSCINI` DD statement to point to the *`prefix.value`*`.SBKMPARM` data set, and ensure that the member name remains as `BKMINI#`.
5. Change the `UPDATE` DD statement to point to the *`prefix.value`*`.SBKMPARM` data set, and ensure that the member name remains as `BKMINI`.
6. Submit the job, and ensure that it executes and completes with a completion code 0 before proceeding to the next task.

   If the job terminates with a nonzero completion code, first check to see if any error messages indicate a situation that is easily correctable. BKMIMERG can be rerun. If the situation does not appear to be easily correctable, contact Technical Support for assistance before proceeding to the next task.

Coding the tokens in the BKMINI parameter library member

Examine the product parameter library member BKMINI and verify that all the Advanced Backup and Recovery for z/OS configuration token values have been set appropriately.

Review the token setting options in the *BKMINI configuration values* appendix to determine if you need to change any of the token values for your installation.

Running BKMJCLBL

Advanced Backup and Recovery for z/OS batch job JCL library members use in-stream PROCs. The in-stream PROCs contain symbolics (&...) that refer to various product data sets. BKMJCLBL substitutes BKMINI token values into the PROC statements in the JCL library members.

Even though distributed product maintenance may not include new JCL members, changes to the BKMINI product parameter library member would necessitate this step. BKMJCLBL is in the product JCL library.

1. Because you are updating into an existing JCL library, it is recommended that you back it up before running this job.
2. Modify the BKMJCLBL job JCL as indicated by the comments at the beginning of the member.
3. Be sure that the BKMINI tokens are set as desired before running BKMJCLBL. BKMJCLBL can be run again at any time.
4. Submit the BKMJCLBL job.

Updating system procedure libraries

If you intend to run Advanced Backup and Recovery for z/OS out of a system procedure library, you must ensure that all necessary JCL members are copied into that procedure library.

- Copy the following JCL members into the system procedure library from which you intend to run Advanced Backup and Recovery for z/OS:
  - BKMABARS
  - BKMBPROC
  - BKMRPROC
  - BKMSTSKA
  - BKMSTSKD
  - BKMSTSKR
  - BKMFPROC

Completing the post-upgrade tasks

You might need to perform some post-upgrade tasks before the product is completely ready for use. Many of these tasks were completed when the previous version of the product was installed at your site.

Review the following list and verify that all of the necessary post-upgrade tasks have been completed at your site. Detailed instructions for performing these tasks are provided in Completing the post-installation tasks.

1. Updating the cross-system ENQ lists.
2. Verifying that the configuration values were set in BKMINI.
3. Providing access authority for the product data sets,
4. Adding BKMWABAR to the TMOEDMxx file.
5. Replacing the DFHSMABR procedure.
6. Modifying member ARCCMDxx in the DFSMShsm parameter library.
7. Updating the ACF$CMD system security command table.

8. Setting up ABARS Monitor.
9. Setting up the **SDSL VOL** option.
10. Loading the demonstration aggregates.
11. Adding security for primary and line commands.
12. Defining security class profiles for CATSCRUB.
13. Editing the **FDR_LOADLIB** token in BKMINI
14. Editing the **ADABAS_LOADLIB** token in BKMINI
15. Defining the security class profiles for the Critical Backup Tracking and Inventory started task.
16. Specifying the SMF/JCL collection method.
17. Providing access authority to the Automated Critical Data Identification data sets.
18. Setting up foreground verification.
19. Verifying that the Automated Critical Data Identification Selection Data Sets are physical sequential data sets.
20. Turning off PF key function display.
21. Editing and running the BKMDB2BG job.
22. Modifying the SBKMASMP library's BKMTAPUX member.

Chapter 7. Accessing Advanced Backup and Recovery for z/OS features

Use the information in this chapter to access the Advanced Backup and Recovery for z/OS features and capabilities. Some features are accessible through both the ISPF dialogs and as batch processes; others are available as batch processes only.

- Accessing Automated Critical Data Identification
- Accessing ABARS Manager
- Accessing Critical Backup Tracking and Inventory
- Accessing CATSCRUB
- Searching for data sets and historical data set usage information
- Accessing reports
- Accessing the ABARS Monitor
- Accessing backup and recovery history information
- Using the Action bar
- Accessing online help
- Viewing messages

Accessing Automated Critical Data Identification

You can use either of two methods to access Automated Critical Data Identification.

- From the **Main Menu**, choose the **Automated Critical Data Identification (ACDI)** option.
- Type EX next to member BKMMGR and press Enter.

Using the Automated Critical Data Identification Scroller

The Automated Critical Data Identification Scroller provides commands that enable you to rearrange columns or rows of data and perform selected operations. The Scroller is available when a table is displayed in Automated Critical Data Identification.

- To use the Automated Critical Data Identification Scroller, press PF1 from any of these panels:
  - The **APPLICATION LIST** panel.
  - The **DATASET DETAIL RECORDS** panel
  - The **DATASET RECORDS FOR INCLUSION IN SDS** panel
  - The **DATASET RECORDS WITH URD INFORMATION** panel
  - The **CURRENT RSP JOB TABLE** panel

Accessing ABARS Manager

You can use the Advanced Backup and Recovery for z/OS ISPF panels to access ABARS Manager functionality.

Using ABARS Manager, you can generate a filtered list of ABARS backup and recovery events. From this filtered list, you can do any of the following:

- View a list of the data sets involved in an event.
- View detailed information about an event, including error, event record, summary, and activity log information.

- Delete an event record.
- Display space usage and allocation information for aggregates.
- View conflict, instruction, restart, and Selection Data Set information for ABACKUP and ARECOVER events.
- Browse or edit an aggregate's instruction data set, SDSL parameters, or Selection Data Set.
- Browse or edit cataloged restart and conflict ARECOVER data set information.
- Submit an ABACKUP or ARECOVER.
- Submit incremental backups.
- Monitor ABARS backups or recoveries as they are executed and associate the ABARS started task with the actual aggregate.

To access ABARS Manager functionality, use the appropriate method:

- From the **Main Menu** panel, select option 1 and press enter to access the ISPF panels for setting up aggregate management and group filter management.
- From the **Main Menu** panel, select option 2 and press Enter to display the **Backup and Recovery Management** panel, where you can access the panels from which you can manage ABARS backup and recovery events and from which you can manage logical aggregate definitions.

Accessing Critical Backup Tracking and Inventory

This information describes how to access Critical Backup Tracking and Inventory.

- From the **Main Menu**, select the **Backup and Recovery Management** option.

  The **Backup and Recovery Management** panel is displayed. From this panel you can access the Critical Backup Tracking and Inventory backup and recovery management options:

  - **Full Volume Dumps**
  - **Jobs**
  - **Job Groups**
  - **Applications**

Accessing CATSCRUB

Use the **CATSCRUB** command to synchronize the ICF catalogs and data volumes when a system is recovered at a disaster recovery site.

1. On the Advanced Backup and Recovery for z/OS **Main Menu** panel, type 4 in the **Command** line and press Enter to display the **Features** panel.
2. On the **Features** panel, type 1 in the **Command** line and press Enter to display the **Submit CATSCRUB** panel.
3. Use the online help to guide you as you specify the keyword settings for the **CATSCRUB** command.
4. Type S in the **Build JCL/Edit for Submit** field and press Enter to view the JCL before submitting it.

Refer to the appropriate chapter in this guide for further information about using this feature.

Searching for data sets and historical data set usage information

Use the Search for Data Sets feature to search all backup events in the IDS for a specified data set name or mask or to search for all data sets that reference the search data set name or mask.

The full and quick search results include data from ABARS Manager and from Critical Backup Tracking and Inventory. The historical data set usage search results include data from Automated Critical Data Identification. The Search for Data Sets feature provides three methods of searching:

- A full search displays all generations for each data set found. This type of search can take quite a bit of time and is limited to 50,000 data sets.

- A quick search displays generation 0 for each data set found. This type of search is usually quicker than a full search and there is no limit to the number of data sets returned by the search.
- The Historical Dataset Usage search displays a list of all data sets that referenced the search data set within a specific time period.

To use the Search for Data Sets feature, complete the following steps:

1. On the Advanced Backup and Recovery for z/OS **Main Menu** panel, type 5 in the **Command** line and press Enter to display the **Search Selection** panel.
2. On the **Search Selection** panel, type the appropriate number in the **Command** line and press Enter to display the **Search for Dataset Setup** panel.
3. Use the online help to guide you as you complete the field specifications on the **Search for Dataset Setup** or **Quick Search for Historical Dataset Record** panel, and then press Enter to start the search and eventually display the search results.
4. Refer to the *Searching for data sets* chapter in this guide for further information about using this feature.

Accessing reports

You can access the Advanced Backup and Recovery for z/OS reports from the **Main Menu** panel.

The following reports are available:

- ABARS Manager backup reports:
  - Overlap by Dataset
  - Overlap by Backup Event
  - Dataset List
  - Event List
  - Tape Pull List
  - Disaster Recovery Space Summary
- ABARS Manager recovery reports:
  - Overlap by Dataset
- Critical Backup Tracking and Inventory backup reports:
  - Overlap by Dataset
  - Overlap by Backup Event
  - Overlap by Job Groups
  - Dataset List
  - Data List by Jobname w/Outputs
  - Full Volume Dump List
  - Jobname List
  - Jobnames Not Tracked by a Job Group
  - What Isn't Backed Up
  - Critical in ACDI with No CBTI Backup Record
  - Critical in ACDI with CBTI Backup Record
  - BKUPEND Summary Report
  - BKUPEND Validation Report
  - Overlap by BKUPEND Application
  - Tape Pull Report
  - Db2 Image Copy Reporting

- ACDI - Critical Files Recovery Space Summary
- Not Critical in Automated Critical Data Identification with Critical Backup Tracking and Inventory Backup Record
- IMS Image Copy report
- Backup List Report

To access these reports, perform the following steps:

1. On the Advanced Backup and Recovery for z/OS **Main Menu** panel, type 6 in the **Command** line and press Enter to display the **Reports** panel.
2. On the **Reports** panel, type the appropriate number in the **Command** line and then press Enter to display a **Reports Menu** panel.
3. On the Reports Menu panel, type the appropriate number in the Command line and press Enter to display a panel where you can specify the options for the report and then specify whether you want the to build the report JCL and review it or submit it for execution or view the report online. Use the online help for the options panel to guide you as you specify the options for the report.

Refer to the *Generating backup reports* chapter in this guide for further information about using this feature.

Accessing the ABARS Monitor feature

Use the Advanced Backup and Recovery for z/OS Monitor feature to view information about in-progress ABARS backups and recoveries.

1. Type 7 in the **Command** line on the Advanced Backup and Recovery for z/OS **Main Menu** panel, and then press Enter to display the Monitor panel.
2. On the **Monitor** panel, type 1 in the **Command** line and then press Enter to display the **Monitor Search Setup** panel, where you can specify a filter that will be used to generate a list of backups or recoveries (or both) whose progress you want to monitor.
3. Use the online help to guide you as you complete the fields on the **Monitor Search Setup** panel, and then press Enter to display the **ABARS Manager Monitor** panel, which shows the filtered list of in-progress backups or recoveries (or both).
4. Refer to the *Monitoring in-progress ABARS backups and recoveries* chapter in this guide for further information about using this feature.

Accessing the event history feature

Use the event history feature of Advanced Backup and Recovery for z/OS to view detailed information about backup and recovery events, organized according to the event's date. You can view information for all backup and recovery events or you can view information for a filtered list of events.

1. On the Advanced Backup and Recovery for z/OS **Main Menu** panel, type 8 in the **Command** line and press Enter to display the **History Search Setup** panel.
2. Use the online help for the **History Search Setup** panel to guide you as you build the filter that determines which events will appear in the **History** panel, then press Enter to display the **History** panel.
3. To view detailed information about an individual event, type the appropriate line command in the **C** column next to the appropriate event, then press Enter.
4. Refer to the appropriate chapter in this guide for further information about using this feature.

Accessing the Fast Replication Services feature

Fast Replication Services (FRS) functionality depends on application backup profiles. The profiles include the specifications that dictate the type of backup, the type of backup utility to be used, Fast Replication target volumes, how the offload occurs, whether data encryption occurs, and information about the

offload data sets. The profile is specified as a control card parameter in the backup and restore job JCL. All access (profile viewing, creation, and modification) to the profiles is through the Profile Display List.

1. From the **Main Menu** panel, select the **Setup and Configuration** option and then press Enter to display the **Setup and Configuration** menu panel.
2. From the **Setup and Configuration** menu, select the **Perform Fast Replication Services** option and then press Enter to display the **Fast Replication Engine Profile Services** panel.
3. You can use the fields on the **Fast Replication Engine Profile Services** panel to filter the subsequent list of existing profiles so that only those whose attributes match your filters appear in the list. The first time you access the profile list, just press Enter to display an unfiltered list of all existing FRS profiles on the **Profile Display List** panel.

   From the **Profile Display List** panel, you can view existing profile attributes, create new profiles, modify existing profiles, and so on.

Accessing online help

Press the HELP function key from any Advanced Backup and Recovery for z/OS panel to display online help information about the panel, its fields, and the commands available from that panel.

Using the action bar

The action bar at the top of the Advanced Backup and Recovery for z/OS ISPF panels provides access to various types of product information and functions.

Not all of the items described here are available from every Advanced Backup and Recovery for z/OS primary panel.

Menu
The **Menu** item provides access to the Advanced Backup and Recovery for z/OS messages and BKMINI values and provides an exit from the product.

Diagnostics
The **Diagnostics** item provides access to information about the product version and release, the load modules, current memory usage, ABARS Manager event trace records in the Advanced Backup and Recovery for z/OS Inventory Data Set (IDS), and the availability of the Fast Replication Services (FRS) feature on the system where you are currently logged on.

Preferences
The **Preferences** item provides access to settings that determine how space usage is displayed and job card specifications.

Utilities
The **Utilities** item is available from the Advanced Backup and Recovery for z/OS **Main Menu** panel only. This item provides access to the utilities you can use to load data into and unload data from the IDS.

The :BACKUP_OPTIONS section of product parameter library member BKMINI contains the options whose settings are used to fix category conflicts that might arise when an ABARS ABACKUP is issued. The settings assigned to these options when Advanced Backup and Recovery for z/OS was installed and configured apply to all aggregates that are added to the IDS. You can use the Utilities item's Reset Aggregate Options option to review and change these settings for all aggregates globally or for a specified list of aggregates.

If the DAD started task is being used on your system, the **Utilities** item's DAD Monitor option shows the DAD Started Task Monitor. The DAD Started Task Monitor provides information about the DAD status, DAD statistics from any image on which a DAD is running, and provides access to the DAD commands for the current image.

Admin
Use the **Admin** item to specify utility equivalents for Critical Backup Tracking and Inventory backups. For example, DFSMSdss executes ADRDSSU; if you use program name other than ADRDSSU, you can include your program name as an equivalent using this option.

To use the action bar, complete the following steps:

1. Position the cursor on the action bar item of interest and press Enter to display the menu for that item.
2. In the text entry area provided on the item's menu, type the number for the option you need, and then press Enter to display the appropriate panel.
3. Press PF1 on any panel to display help information for the fields and commands available on that panel.

Preferences

You can use the Action bar's **Preferences** menu to specify default values for job cards and the unit you want used when Advanced Backup and Recovery for z/OS displays space amounts.

Specifying preferences for displaying data set space usage information

You can use the Action bar's **Preferences** menu to specify your preferences for the unit of measure Advanced Backup and Recovery for z/OS uses when displaying data set space usage information. You can also use this menu to specify your preference for the number of places displayed to the right of the decimal point when displaying data set space usage information.

1. Put the cursor on the Action bar's **Preferences** menu and press Enter to display the menu options.
2. Type 1 in the menu's text entry area and press Enter to display the **Dataset Space Display Options** panel.
3. On the **Dataset Space Display Options** panel, perform these steps:
   a) In the **Space Unit** field, specify the appropriate alphanumeric character. Press the HELP function key for additional information about the options offered.
   b) In the **Places right of decimal** field, specify the appropriate number.
4. Type SAVE in the **Command** line and press Enter, or press the END function key, to save your specifications and close the **Dataset Space Display Options** panel.

Specifying default job card information

You can use the Action bar's **Preferences** menu to specify default job card information for ABARS Manager, Critical Backup Tracking and Inventory, and CATSCRUB. Job card lines that you leave blank are ignored when the job is submitted.

1. Put the cursor on the Action bar's **Preferences** item and press enter to display the menu options.
2. To specify job card options for ABARS Manager, complete these steps:
   a) Type 2 in the menu's text entry area and press Enter to display the **ABARS Manager Job Card Options** panel.
   b) In the **Randomize Last Character of Job Name** field, type Y if you want the last character of the job name randomized; if you want the job name on the job card used exactly as it appears on this panel, type N.
   c) In the **ABACKUP JOB CARD** fields, change the job card to affect how the job card is used during an ABACKUP operation.
   d) In the **ARECOVER JOB CARD** fields, change the job card to affect how the job card is used during an ARECOVER operation.
   e) In the **LOAD ACTIVITY LOG/LOAD IDS/UNLOAD IDS/REPORTS JOB CARD** fields, change the job card to affect how the job card is used for a load or unload of the Inventory Data Set.
   f) To save your specifications and close the ABARS Manager Job Card Options panel, press the END function key and press Enter, or type SAVE in the **Command** line and press Enter.
3. To specify job card options for Critical Backup Tracking and Inventory, perform these steps:
   a) Type 3 in the menu's text entry area and press Enter to display the **Job Card Options** panel for Critical Backup Tracking and Inventory.

b) In the **Randomize Last Character of Job Name** field, type Y if you want the last character of the job name randomized; if you want the job name on the job card used exactly as it appears on this panel, type N.

c) In the **BLDREST JOB CARD** fields, change the job card to affect how the job card is used during a restore operation.

d) In the **LOAD ACTIVITY LOG/LOAD IDS/UNLOAD IDS/REPORTS JOB CARD** fields, change the job card to affect how the job card is used for a load or unload of the Inventory Data Set.

e) To save your specifications and close the Critical Backup Tracking and Inventory Job Card Options panel, press the END function key and press Enter, or type SAVE in the **Command** line and press Enter.

4. To specify job card options for CATSCRUB, perform these steps:

a) Type 4 in the menu's text entry area and press Enter to display the **Job Card Options** panel for CATSCRUB.

b) In the **Randomize Last Character of Job Name** field, type Y if you want the last character of the job name randomized; if you want the job name on the job card used exactly as it appears on this panel, type N.

c) In the **CATSCRUB JOB CARD** fields, change the job card to affect how the job card is used during a CATSCRUB operation.

d) To save your specifications and close the CATSCRUB Job Card Options panel, press the END function key and press Enter, or type SAVE in the **Command** line and press Enter.

5. To specify job card options for Automated Critical Data Identification, perform these steps:

a) Type 5 in the menu's text entry area and press Enter to display the **Job Card Options** panel for Automated Critical Data Identification.

b) In the **ACDI JOB CARD** fields, change the job card to affect how the job card is used during an Automated Critical Data Identification utility operation.

c) To save your specifications and close the **ACDI Job Card Options** panel, press the END function key and press Enter.

Utilities

You can use the **Utilities** menu on the **Main Menu** panel's Action bar to access the utilities that let you load an ABACKUP or ARECOVER activity log, load and unload the Advanced Backup and Recovery for z/OS Inventory Data Set (IDS) and to access the DAD Started Task Monitor.

Loading activity logs into the IDS

You can use the Action bar's **Utilities** menu to load ABARS ABACKUP or ARECOVER activity logs into the Inventory Data Set (IDS).

1. Place the cursor on the Action bar's **Utilities** item and press Enter to display the menu options.
2. In the menu's text entry area, type 1 and press Enter to display the **Load Activity Log** panel.
3. If you want to filter the list of data sets displayed on the **Load Activity Log** panel, specify a data set name or mask in the **Dataset Name or Mask** field and press Enter.
4. Use the appropriate line command for the data sets in the list on the lower part of the panel:
   - Type B in the **Cmd** field next to a data set you want to browse.
   - Type L in the **Cmd** field next to a data set you want loaded into the IDS.
   - Type X in the **Cmd** field next to a data set you want deselected for loading into the IDS.
5. Press Enter to pop up a window for each data set you selected for loading.
6. Make the appropriate specifications on each pop-up window and press Enter. When you complete the last pop-up window, the **Load Activity Log** panel displays *SELECTED in the **Selected for Load** column for every data set that will be loaded into the IDS.
7. Specify whether you want to view the JCL for the load job before submitting it:

- Type S next to the **Build/Submit JCL** field if you want the JCL built and submitted for execution.
- Type S next to the **Build/View JCL** field if you want to review the JCL and then submit it manually for execution.

8. Press Enter.

Reloading extracted data into the IDS

You can use the Action bar's **Utilities** menu to reload into the IDS data that was extracted from it previously. You can reload ABARS Manager data, Critical Backup Tracking and Inventory data, or all extracted data.

Critical Backup Tracking and Inventory and ABARS Manager share an IDS.

If you need to remove Critical Backup Tracking and Inventory entries from an IDS, you can do so by first unloading the ABARS Manager backup data, deleting the IDS, creating a new IDS, and then finally reloading the extracted ABARS Manager backup data into the new IDS.

To reload extracted data into the IDS, complete the following steps:

1. Place the cursor on the Action bar's **Utilities** menu and press Enter to display the menu options.
2. Type the appropriate number in the menu's text entry area:
   - Type 2 and press Enter if you want to reload only the ABARS Manager data that was extracted previously from the IDS.
   - Type 3 and press Enter if you want to reload only the Critical Backup Tracking and Inventory data that was extracted previously from the IDS.
   - Type 4 and press Enter if you want to reload all data that was extracted previously from the IDS.

   The appropriate **Load Inventory Data Set** panel appears.
3. Use the online help to guide you as you complete the **LOAD OPTIONS** fields on the **Load Inventory Data Set** panel.
4. Specify whether you want the JCL submitted automatically:
   - Type S in the **Build/Submit JCL** text entry field if you want to have the JCL build and submitted for execution automatically.
   - Type S in the **Build/View JCL** text entry field if you want to review the JCL and submit it manually.
5. Press Enter.

Unloading data from the IDS

You can use the Action bar's **Utilities** menu to unload ABACKUP and ARECOVER event data from the IDS into a data set. You can unload ABARS Manager data, Critical Backup Tracking and Inventory data, or all ABARS event data.

Critical Backup Tracking and Inventory and ABARS Manager share an IDS.

If you need to remove Critical Backup Tracking and Inventory entries from an IDS, you can do so by first unloading the ABARS Manager backup data, deleting the IDS, creating a new IDS, and then finally reloading the extracted ABARS Manager backup data into the new IDS.

To unload event data from the IDS, complete these steps:

1. Place the cursor on the Action bar's **Utilities** menu and press Enter to display the menu options.
2. Type the appropriate number in the text entry area on the **Utilities** menu and press Enter:
   - Type 5 and press Enter if you want to unload only ABARS Manager event data from the IDS.
   - Type 6 and press Enter if you want to unload only Critical Backup Tracking and Inventory event data from the IDS.
   - Type 7 and press Enter if you want to unload all event data from the IDS.

   The appropriate **Unload Inventory Data Set** panel appears.

3. Use the online help to guide you as you complete the **UNLOAD OPTIONS** and **OPTIONAL** fields on the **Unload Inventory Data Set** panel.
4. Specify whether you want the JCL submitted automatically:
   - Type S in the **Build/Submit JCL** text entry field if you want to have the JCL build and submitted for execution automatically.
   - Type S in the **Build/View JCL** text entry field if you want to review the JCL and submit it manually.
5. Press Enter.

Resetting the category conflict backup options for aggregates

Whenever a new ABACKUP category conflict fix option is added to the :BACKUP_OPTIONS section of the product parameter library's BKMINI member, you can use the Action bar's **Utilities** menu to update one or more aggregates with the new option setting.

When Advanced Backup and Recovery for z/OS was initially installed at your site, the :BACKUP_OPTIONS section of BKMINI contained the options available to fix category conflicts that occur during an ABACKUP. The values assigned to these options are the defaults for all aggregates (the defaults can be overridden for an individual aggregate as it is added to the IDS). When the available options in :BACKUP_OPTIONS change, you can avoid having to update every aggregate in the IDS by using the Reset Aggregate Options option on the **Utilities** menu.

To reset the aggregate backup category conflict options, complete the following steps:

1. Put the cursor on the Action bar's **Utilities** item and press Enter to display the menu options.
2. Type 8 in the menu's text entry area and press Enter to display the **Select Aggregate Option** window.
3. On the **Select Aggregate Options** window, type S in the text entry area next to the option with which you want to update the aggregates in the IDS, and then press Enter to display the **Aggregate Option Description** window.
4. On the **Aggregate Option Description** window, read the option description and take the appropriate action:
   - If you selected the appropriate option on the **Select Aggregate Options** window, press Enter to display the **Reset Aggregate Options** panel, where you can select the aggregate whose option settings you want to update.
   - If you selected the wrong option on the **Select Aggregate Options** window, press the END function key, and then repeat steps 3 and 4 before continuing with step 5.
5. Use the online help to guide you as you make your specifications on the **Reset Aggregate Options** panel, then type SAVE in the **Command** line and press Enter.

Accessing the DAD task monitor

You can use the Action bar's **Utilities** menu to access the DAD task monitor, which shows DAD status and statistics information and provides access to the DAD commands.

1. Put the cursor on the Action bar's **Utilities** men and press Enter to display the menu options.
2. Type 9 in the menu's text entry area and press Enter to display the **DAD Monitor** panel.
3. If you want to display DAD status information, do the following:
   a) Type 1 in the **Command** line and press Enter to display the **Select SYSID** panel.
   b) In the **Enter SYSID** field, type the system ID for which you want to display DAD task status information, and then press Enter to display the **DAD Task Status by SYSID** panel.
4. If you want to display DAD statistics information, do the following:
   a) Type 2 in the **Command** line and press Enter to display the **Select SYSID and Date** panel.
   b) In the **Enter SYSID** field, specify the system ID for the DAD task whose statistics you want to view, and then press Enter to display the **DAD Processing Statistics** panel.
5. If you want to access the DAD commands, do the following:

a) Type 3 in the **Command** line and press Enter to display the **DAD Commands** panel.

b) In the **Command** line on the **DAD Commands** panel, type the appropriate number and press Enter.

Specifying backup utility equivalents

You can use the Action bar's **Admin** menu to specify the backup utility equivalents for Critical Backup Tracking and Inventory. For example, DFSMSdss executes ADRDSSU; if your site uses a program name other than ADRDSSU, use this feature to specify the name of an equivalent program.

1. Put the cursor on the Action bar's **Admin** item and press enter to display the menu options.
2. Type 1 in the menu's text entry area and press Enter to display the **Define Utility Equivalents** window.
3. Use the text entry areas on the **Define Utility Equivalents** window to specify the appropriate program names. You may specify as many as five equivalent names for any one utility.
4. When you have completed your specifications, press the END function key to save your specifications and close the **Define Utility Equivalents** window.

Viewing Advanced Backup and Recovery for z/OS messages

Product messages are distributed with the product in the messages library members. You can view the messages directly from the messages members by using an ISPF menu option or by using Chicago-Soft's® Quick-Ref™, if it is installed in your environment. You can also view messages by using the **Menu** item on the product's Action bar.

Message identifier naming conventions

Advanced Backup and Recovery for z/OS message identifiers are in the form *pppmmmnnx*.

Where:

- *ppp* is the product code, BKM.
- *pppmmm* is the member identifier.
- *nn* is the message number.
- *x* is the message type indicator (E, I, or W for error, informational, or warning, respectively).

Determining which MESSAGES member contains a message's description

Messages are in the MESSAGES library in a member name that matches the first six characters of the message identifier (ID). For example, if the message identifier is ABC6709I, you would browse member ABC670 in the messages library.

Using the product interface to view messages

You can access the messages library by using the Action bar provided at the top of every panel in the product's ISPF user interface.

1. Position the cursor on the Action bar's **Menu** item and press Enter to display the menu items.
2. In the text entry area on the menu, type 1 and then press Enter to open the **Choose Message Prefix** panel.
3. On the **Choose Message Prefix** panel, type the appropriate letter in the **CMD** line to specify whether you want to look up a Advanced Backup and Recovery for z/OS message or a Fast Replication Services message and then press Enter.
4. Type a message ID or a prefix mask (the mask must be at least 6 characters and cannot include masking characters, such as % or *) and then press Enter.
5. Scroll through the messages to locate the messages of interest.

Browsing the Messages library

You can browse the members of the Advanced Backup and Recovery for z/OS messages library.

Be sure you browse the most current members of the messages library. Members are updated and distributed with new releases of the product, cumulative maintenance, and (occasionally) by PTFs.

The information for a message resides in a messages library member whose name matches the first six characters of the message identifier. For example, if you receive a message that begins with the identifier BKMA0803E , you need to browse the content of member BKMA08 to view the message information.

To browse the members of the message library, complete these steps:

1. Open the appropriate member of the messages library in an ISPF browse session.
2. Use the SCROLL function keys or the FIND command to locate the message identifier of interest to you. For example, to locate the information for BKM0803E, type `F  BKM0803E  1` in the **Command** line and press Enter.

Chapter 8. Using Automated Critical Data Identification to identify critical data

This topic provides an overview of how to use Automated Critical Data Identification. More specific information about each step in the process is provided in subsequent chapters.

Before you use Automated Critical Data Identification, you must identify the applications you want Automated Critical Data Identification to manage. Specifically, you must identify:

- The applications you want Automated Critical Data Identification to manage.
- The names of the jobs and job streams that comprise each application (including their sequence or structure).
- The names of the SMF data sets (if you plan to use SMF Scan).

1. **Define the application to Automated Critical Data Identification.** Specify Setup options for each application you want Automated Critical Data Identification to manage. These options describe which job collection method is used for the application and whether the application is defined for forward or rerun recovery. Automated Critical Data Identification uses this information to determine which data sets belong to the application and their criticality for backup.
2. **Collect the application job structure (job collection).** Job collection determines which jobs belong to the application and any job scheduling product that is used to run the application. You can perform job collection through a job scheduling interface, using ISPF Edit (using job names or job name masks), or by using a pre-staged or user-supplied data set.
3. **Refresh the job table.** Before an application can be tracked by RSP or SMF Scan, you must refresh the job table to update the job table with the collected job names. To refresh the job table, run BKMJRFSH, a member of the product JCL library.
4. **Identify the application data sets.** Automated Critical Data Identification uses JCL and SMF data to determine which data sets are used by the application, and of these data sets, which are required for application recovery. RSP analyzes all JCL and SMF data to determine which data sets are used by the application and which are required for backup and recovery.

   Note: You can use SMF Scan to analyze SMF data. However, SMF Scan can only be used to analyze the SMF data (it does not analyze JCL data).
5. **Run the jobs that belong to the application.** Running the jobs that belong to the application enables Automated Critical Data Identification to record the application's data set information in the Automated Critical Data Identification Application Data Inventory database.
6. **Create the Selection Data Set.** After the data has been collected and stored in the Automated Critical Data Identification Application Data Inventory database, create the Selection Data Set (the list of data sets to back up) using one of these methods:
   - If you used RSP, run BKMAPLEN. BKMAPLEN identifies the end of the application and builds the Selection Data Set.
   - If you used SMF Scan, run the Verification program. Verification takes the records from the Automated Critical Data Identification Application Data Inventory database and builds the Selection Data Set.
   - You can also use Verification to re-evaluate data and rebuild the Selection Data Set. For example, if you run BKMAPLEN and you want to create some filters or change some settings and you want to know the results of your changes.
7. Optional: **Validate the backup.** Together, Automated Critical Data Identification and Critical Backup Tracking and Inventory provide an optional feature (Application Backup and Restore) to help ensure that the data sets identified by Automated Critical Data Identification have been backed up and that none were missed. This feature is helpful if you have backups in place and are using the Selection Data Set from Automated Critical Data Identification as a comparison to what was backed up (instead of using it as input to a backup).

Note: Note that the DAD started task (BKMSTSKD) must be up in order to support Critical Backup Tracking and Inventory job tracking for those jobs that use DFSMSdss and include the FREE=CLOSE JCL parameter; for more information about the FREE=CLOSE JCL parameter, see Chapter 17, "Critical Backup Tracking and Inventory setup and configuration," on page 165.

Identifying applications and job streams

Before using Automated Critical Data Identification, you must identify the applications, or job streams, to be processed by Automated Critical Data Identification.

Automated Critical Data Identification considers an *application* to be a group of related jobs. The data sets created and accessed by these jobs are the data sets that Automated Critical Data Identification includes in the Selection Data Set for backup.

When you define an application to Automated Critical Data Identification you do not need to include all jobs that comprise the application. For example, a finance application might consist of several separate job streams, such as one for accounts payable, another for accounts receivable, and so on. You can define each of these smaller applications separately if needed. Therefore, it is important that the person who will perform the analysis, knows the application structure and function in order to determine how to subdivide it, when to backup the data, and which critical data sets are necessary for disaster recovery.

Note: When converting an application to an Automated Critical Data Identification managed backup, there might be questions as to which jobs and data sets comprise the application, their size, usage patterns, and importance. Even if the Automated Critical Data Identification process will be performed by a central storage administration group, it is helpful to have someone available who knows the application and can answer questions and resolve any problems that might occur. Therefore, it is recommended that you identify a contact person for each application.

Identifying anchor jobs and tables

Some applications consist of several job streams. You must identify the name of the anchor jobs or tables for each job stream so Automated Critical Data Identification can determine which jobs comprise the application.

By examining the JCL and SMF records from one reference point to the next, Automated Critical Data Identification can accurately determine which data sets are critical for application backup.

Tip: The *anchor job* provides a reference point for the application. It is typically, though not always, the first job in the application job stream.

- You can usually obtain information about anchor jobs or tables from the contact person for the application. However, if anchor job names, or job names specified using a job name mask are used, Automated Critical Data Identification can determine which jobs comprise the application from the job scheduling system.

  Note: If you plan to use a job scheduler interface to perform job collection you must identify the anchor job, or tables, for each application job stream associated with the job scheduling product you are using. For example, if you use the Unicenter CA-7 scheduler interface, you would specify the name of the anchor job for the application to identify the jobs related to the application. (If you are using a scheduler interface such as CONTROL-M, you would specify *anchor tables* instead of anchor jobs.)

Application checklist

Use this checklist to help you track the information you gather for each application you define to Automated Critical Data Identification.

Table 1. Application checklist for Automated Critical Data Identification

Item	
Application name:	

Table 1. Application checklist for Automated Critical Data Identification (continued)	
Item	
Application description:	
Contact person:	
Anchor job names, job tables, or job name masks:	
Selection DSNs:	
Does the User ID or Batch ID that will be used to run Automated Critical Data Identification have READ access to the procedure libraries, job scheduling product files, and SMF data sets?	
Does the User ID or Batch ID that will be used to run Automated Critical Data Identification have WRITE access to the Automated Critical Data Identification database and Selection Data Sets?	
Have all appropriate SAF authorities been assigned?	

Chapter 9. Defining an application to Automated Critical Data Identification

When you define an application to Automated Critical Data Identification, you specify general information about the application, whether you want to collect SMF or JCL data, and whether or not the application is defined for forward or rerun recovery.

Automated Critical Data Identification uses the information you specify to determine which data sets belong to the application and their criticality for backup.

1. Determine the recovery method for the application:

Option	Description
Forward recovery	Forward recovery assumes the backup is performed at the *end* of an application cycle, or at various points within an application cycle. At the disaster site, the application data is restored and the application continues forward with the next processing cycle. This recovery method requires any output data sets needed as input for the next cycle be included in the application backup. **Note:** Forward recovery is the simplest recovery method to implement. Rerun recovery is generally more difficult to implement— but it is more efficient because less data is backed up.
Rerun recovery	Rerun recovery assumes the input data sets are backed up *before* the beginning of application cycle processing. At the recovery site, the application is restored and the last cycle is rerun. Output data sets are recreated during the rerun of the application. This recovery method requires only critical data sets (inputs to the application) be included in the application backup. Noncritical data sets (outputs) are not included because they consume backup time and resources.

2. From the Advanced Backup and Recovery for z/OS Main Menu, select the **Automated Critical Data Identification** option.

   The Automated Critical Data Identification **Main Menu** is displayed.
3. Select the **SELECT/CREATE APPL** option.

   The **SELECT Current/NEW Application Name** panel is displayed.
4. Specify the application you want to define to Automated Critical Data Identification using one of these methods:
   - To define a new application, type the application name in the **Specify NEW Application Name** field and press Enter.
   - To modify a previously defined application, specify the existing application name in the **Use Current/OLD Application Name** field and press Enter.

     Note: The default is the last specified application.
   - To select an application from a list of applications that have been previously defined to Automated Critical Data Identification, type Y in the **Select From Application List** field and press Enter. The **CREATE APPLICATION LIST** panel is displayed:
     a. In the **ENTER APPLICATION OR MASK** field, specify the application name or mask. For example, you can specify `BK00*` to list all applications that begin with "BK00." (Alternatively, you can press Enter to display a list of all previously defined applications. The **APPLICATION LIST** panel is displayed.)
     b. Type S in the **CM** field to select the application you want to modify and press Enter.

   After you have selected the application you want to define, the **Application AutoBuild Options** panel is displayed.

5. Select the **SETUP** option.

   The **SETUP OPTIONS** panel is displayed.
6. Specify these options as they apply to the application:
   - **APPLICATION General Setup**
   - **Backup End**
   - **Collection**
   - **RSP**
   - **JCL**
   - **Batch SMF Scan**
   - **Current VERIFICATION**
   - **VERIFICATION GDG**
   - **SELECTION DATASETS**

   Note: Any values you define for an application are *not* effective if the corresponding option is globally disabled in the BKMINI member of the product parameter library.
7. When you are finished, press PF3 to save your changes or type `Cancel` to cancel your changes.

Specifying Application General Setup options

Application General Setup options enable you to specify an optional description of the application and, if you are using ABARS, the ABARS aggregate to associate with the application.

1. (Optional) In the **Appl DESC** field, type a description of the application (maximum of 48 characters). You can overwrite the contents of this field if needed.
2. If you are using ABARS, type the name of an ABARS aggregate to associate with the application in the **Associate with ABARS Aggregate** field.
   a) To determine which aggregates can be associated with the current application, type a question mark (?) in the first position of the field and under **Current VERIFICATION Options** type **ABARS** for **SELECTION Dsn Format** field to display a list of current ABARS aggregates within the current SMS configuration.

      A list of current aggregates is displayed as shown below.

```
AGGREGATE LIST
    Row 1 to 15 of 135
   Cmd ==>
   Use 'S' To Select

  -Opr- ---NAME---

     _    ABRTST01
     _    ABRTST02
     _    ABRTST03
     _    ABRTST11
     _    ABRTST12
     _    ABRTST13
     _    ABRTST14
     _    ABRTST15
     _    ABRTST99
     _    ABR2363A
     _    ABR2363B
     _    ABR2363C
     _    ABR2363D
     _    ABR2363E
     _    ABR2363F
```

   b) To select an aggregate to associate with the application, type an S in the **Opr** column next to the aggregate you want to select.

   If you specify an ABARS aggregate, the Selection Data Set specified in the ABARS aggregate group definition is used when Automated Critical Data Identification generates the Selection Data Set control

statements (INCLUDE, EXCLUDE, ACCOMPANY, and ALLOCATE). If the aggregate exists, Automated Critical Data Identification automatically supplies the Selection Data Set name. (If you do not want to write to this Selection Data Set, change the name of the Selection Data Set in the **SELECTION DATASETS** section on the **SETUP OPTIONS** panel.)

Note: If you are not using ABARS, leave the **Associate with ABARS Aggregate** field blank. If the application is not associated with an ABARS aggregate, Automated Critical Data Identification automatically supplies the Selection Data Set name.

Specifying Backup End options

Backup End options are application-specific BKUPEND program related parameters. These options are available to use with Critical Backup Tracking and Inventory to track and restore by application.

The default values for the Backup End options are obtained from the BKUPEND token values found in the BKMINI CBTI_BACKUP_END_OPTIONS section.

Note: If you are not using the Application Backup and Restore feature of Critical Backup Tracking and Inventory you do not need to specify Backup End options.

1. In the **Backup End Required** field, specify the relationship between BKMAPLEN and BKUPEND processing.

   Note: If BKUPEND is required, it is to be executed between BKMAPLEN executions.

   Valid options are:

Option	Description
N	There is no need to perform the BKUPEND after each BKMAPLEN process. If the BKUPEND process is not performed, no related message is issued by the BKMAPLEN process. In addition, the BKMAPLEN process is not required prior to the BKUPEND process. If the BKMAPLEN process is not performed, no related message is issued by the BKUPEND process.
W	The BKUPEND process should be performed after each BKMAPLEN process. If the BKUPEND process is not performed, a related warning message is issued by the BKMAPLEN process. In addition, the BKMAPLEN process should be performed prior to the BKUPEND process. If the BKMAPLEN process is not performed, a related warning message is issued by the BKUPEND process.
E	The BKUPEND process is required after each BKMAPLEN process. If the BKUPEND process is not performed, a related error message is issued by the BKMAPLEN process. In addition, the BKMAPLEN process is required prior to the BKUPEND process. If the BKMAPLEN process is not performed, a related error message is issued by the BKUPEND process.

2. In the **Backup Cycles to Retain** field, specify the number of application cycles (000-999) that are to be tracked by Critical Backup Tracking and Inventory in the Advanced Backup and Recovery for z/OS Inventory Data Set.

   Critical Backup Tracking and Inventory tracks cycles, including the backup information found for each critical data set discovered by Automated Critical Data Identification, thus allowing you to restore data sets that are specific to a given application cycle.

3. In the **Application Only Backup** field, specify if only application-created backups should be recorded for cycle point recovery:

Option	Description
Y	Record only backups tracked by Critical Backup Tracking and Inventory and created by backup jobs identified using the Automated Critical Data Identification **BACKUP JOB LIST** application option.
N	Record any backup tracked by Critical Backup Tracking and Inventory.

4. In the **Include HSM Backups** field, specify whether BKUPEND searches for a backup in the DFSMShsm incremental backup inventory for any critical data set identified by Automated Critical Data Identification:

Option	Description
Y	Search for and report backups made by DFSMShsm incremental backup.
N	Do not search for and report backups made by DFSMShsm incremental backup.

5. In the **Include Physical Backups** field, specify whether BKUPEND searches for a backup in the Critical Backup Tracking and Inventory tracked volume dump inventory:

Option	Description
Y	Search for and report backups made by tracked volume dumps.
N	Do not search for and report backups made by tracked volume dumps.

6. In the **Exclude Accompany** field, specify whether BKUPEND searches for a backup in the Automated Critical Data Identification Application Data Inventory database for any critical data set classified in the Automated Critical Data Identification application as an ABARS ACCOMPANY data set:

Option	Description
Y	Do not search for and report backups for ACCOMPANY data sets.
N	Search for and report backups for ACCOMPANY data sets.

7. In the **Exclude Allocate** field, specify whether BKUPEND searches for a backup in the Automated Critical Data Identification Application Data Inventory database for any critical data set classified in the Automated Critical Data Identification application as an ABARS ALLOCATE data set.

Option	Description
Y	Do not search for and report backups for ALLOCATE data sets.
N	Search for and report backups for ALLOCATE data sets.

8. In the **Bad Backup RC** field, specify the backup return code (4 or 8) that indicates a bad backup. If the return code from any current backup found is greater than or equal to the return code specified, a warning message is issued.
9. In the **No Backup RC** field, specify the return code (0, 4, or 8) to indicate a matching data set has no backup. If no backup is found for a critical data set a warning message, with the return code specified, is issued.
10. In the **Check Changed Since Last Bkup** field, specify whether to report on any critical data sets changed since the last tracked backup:

Option	Description
Y	Identify data sets changed since the current backup.
N	Do not identify data sets changed since the current backup.

Note: This option requires the use of the changed data set tracking function provided by the BKMSTSKD (DAD) started task and the BKMINI section **CBTI_DUMP_BACKUP_PROCESS_OPTIONS** token **CHANGED_DATASET_TRACKING** value set to Y.

11. In the **Changed Since Last Backup RC** field, specify whether a return code of 0, 4, or 8 should be generated if, during BKUPEND, any data set identified as critical by Automated Critical Data Identification has an associated backup in Critical Backup Tracking and Inventory but the data set has changed since the last tracked backup.

Specifying Collection options

Collection options specify whether you want to collect SMF or JCL data and how you want the application end function performed at the application cycle end point.

Automated Critical Data Identification uses JCL and SMF data to determine which data sets are used by the application, and of these data sets, which are required for application recovery.

1. In the **COLLECT DATA VIA SMF OR JCL** field, specify how you want to collect data. Choose one of these options:

Option	Description
M	Minimum. (Default) Only the first reference to each data set and the first URD (Update/Rename/Delete) are recorded.
N	RSP does not record SMF records or JCL information for this application. Specify this option when you want to define applications but do not want to collect data yet. When you run BKMJRFSH, it does not update the job file for the application.
A	This option is reserved for use in future releases.

2. In the **CYCLE CONTROL** field, select one of the following options to specify how you want the application end function performed:

Option	Description
A	APPLEND. The application end function is performed by a batch job (refer to the BKMAPLEN member in the product JCL library) added to the scheduled application jobs at the cycle end point. This is the default setting for this option.
J	JOB. The application end function is performed automatically by a batch job submitted by the RSP started task at the cycle end point. The cycle end point is detected when a cycle end job completes.

Specifying RSP options

RSP options specify the amount of RSP data to retain.

Before you specify RSP options for an application, make the following determinations:

- The number of cycles of RSP data you want to save. (The BKMAPLEN process denotes the end of a cycle.)
- How many cycles of RSP data you want to include in the Selection Data Set.
- The SMF and JCL records you want to capture.

RSP options specify the number of cycles of RSP SMF data to retain in the Automated Critical Data Identification Application Data Inventory database and whether you want to delete old cycles automatically.

Note: The Automated Critical Data Identification option to track more than one application cycle only affects how many cycles of SMF and JCL data are saved. (JCL and SMF records are written to the Automated Critical Data Identification Application Data Inventory database as *Data Set Detail records*, or DSDs).

If the number of cycles specified by the **Number of RSP cycles to retain** and the **Number of JCL cycles to retain** options are greater than one, the DSD records for the various cycles are combined and analyzed, and the result is the creation of Data Set Records (DSRs). These records are built for the last completed

cycle and reflect the results of the **SETUP** option settings and filter specifications. DSR records are used to generate a Selection Data Set. At any time, only one version of the DSR records exist.

1. In the **Number of RSP cycles to retain** field, specify the number of cycles of RSP collected data to retain in the Automated Critical Data Identification Application Data Inventory database.

   The number of cycles to be included in the Selection Data Set is controlled by the current Verification option **RSP cycles to include in Sel. DS.**. If you set **RSP cycles to include in Sel. DS.** to a value greater than 1, you must set **Number of RSP cycles to retain** to at least the same value.

   Note: If you want to retain more than one cycle, review the size of Automated Critical Data Identification Application Data Inventory database to determine if the number of cycles you want to retain is appropriate for your installation).
2. In the **AUTOMATICALLY DELETE OLD CYCLES** field, specify whether or not you want to automatically delete old cycles. Specify one of these options:
   - **Y** — (Default) A cycle of data beyond the number of cycles to be retained will be deleted when BKMAPLEN increments the cycle number (unless the BKMAPLEN CYCLENOINC parameter is specified).
   - **N**— Do not delete a cycle of data beyond the **Number of RSP cycles to retain** value. (If you specify **N**, you can change it to **Y** and the excessive cycles will be deleted the next time you run BKMAPLEN.)

Specifying JCL options

JCL options control the amount of JCL data to retain if you use RSP JCL Capture to collect data set information for an application.

Note: The Automated Critical Data Identification option to track more than one application cycle only affects how many cycles of SMF and JCL data are saved. (JCL and SMF records are written to the Automated Critical Data Identification Application Data Inventory database as *Data Set Detail records*, or DSDs).

If the number of cycles specified by the **Number of RSP cycles to retain** and the **Number of JCL cycles to retain** options are greater than one, the DSD records for the various cycles are combined and analyzed, and the result is the creation of Data Set Records (DSRs). These records are built for the last completed cycle and reflect the results of the **SETUP** option settings and filter specifications. DSR records are used to generate a Selection Data Set. At any time, only one version of the DSR records exist.

1. In the **Number of JCL cycles to retain** field, specify the number of cycles of JCL data to retain in the Automated Critical Data Identification Application Data Inventory database. (The number of cycles to be included in the Selection Data Set is controlled by the current Verification option **RSP cycles to include in Sel. DS.**)

   Note: If you want to retain an unusually large number of cycles, review the size of the Automated Critical Data Identification Application Data Inventory database to determine if the number of cycles you want to retain is appropriate for your installation).
2. In the **AUTOMATICALLY DELETE OLD CYCLES** field, specify whether or not to automatically delete old cycles:

Option	Description
Y	(Default) A cycle of data beyond the **Number of JCL cycles to retain** value is deleted when BKMAPLEN increments the cycle number (unless the BKMAPLEN **CYCLENOINC** parameter is specified).
N	Do not delete a cycle of data beyond the **Number of JCL cycles to retain** value. (If you specify **N**, you can change it to **Y** and excessive cycles are deleted the next time you run BKMAPLEN.)

Specifying Batch SMF Scan options

Batch SMF Scan options control how SMF Scan collects data set information.

1. In the **Number of SMF cycles to retain** field, specify the number of cycles of batch SMF data to retain in the Automated Critical Data Identification Application Data Inventory database.

   Note: If you want to retain an unusually large number of cycles, review the size of the Automated Critical Data Identification Application Data Inventory database to determine if the number of cycles you want to retain is appropriate for your installation).
2. In the **AUTOMATICALLY DELETE OLD CYCLES** field, specify whether or not to automatically delete old cycles:

Option	Description
Y	(Default) Specify the number of cycles of data (beyond the **Number of SMF cycles to retain** value) to be deleted when SMF Scan increments the cycle number.
N	Do not delete a cycle of data beyond the **Number of SMF cycles to retain** value. (If you specify **N**, you can change it to **Y** and excessive cycles will be deleted the next time you run SMF Scan.)

Specifying Current Verification options

Current Verification options control the display of data during the generation of the Selection Data Set in the Verification process.

1. In the **APPLY FILTERING DURING VERIFICATION** field, specify whether or not the filter criteria specified using the **FILTERS** option of the **Application AutoBuild Options** panel should be applied during the Verification process. Specify one of these options:

Option	Description
Y	(Default) Apply the filter criteria before the Selection Data Set is displayed. Data filtered out during Verification remains in the Automated Critical Data Identification Application Data Inventory database and can later be included in the Selection Data Set if the filter criteria is changed and the Verification process is performed again.
N	Do not apply filtering criteria.

2. In the **RE-EVALUATE DATASET ENTRIES** field, specify whether Automated Critical Data Identification should re-evaluate each data set entry within the application for classification as INCLUDE, EXCLUDE, ALLOCATE, or ACCOMPANY in the Selection Data Set. Specify one of these options:

Option	Description
Y	(Default) Automated Critical Data Identification re-evaluates the data set entry to determine its status. The only exceptions are external data sets added by filters and data sets whose status has been determined by filters.
N	Automated Critical Data Identification does not re-evaluate the data set entry to determine its status. **CAUTION:** If you specify N, the first presence of an entry for a data set determines the status regardless of subsequent information gathered from JCL or SMF data.

3. (ABARS only) In the **USER TAPE DATASETS AS ACCOMPANY** field, specify whether data sets that reside on tape at the time of Automated Critical Data Identification analysis are to be placed in the ACCOMPANY list or the INCLUDE list when generating aggregate selection control statements. Specify one of these options:

Option	Description
Y	Place tape data sets on the ACCOMPANY list in the Selection Data Set. **Tip:** Specify **Y** the first time a Selection Data Set is created so that all tape files are placed in a concise list for ease of review.
N	(Default.) Place tape data sets in the INCLUDE list in the Selection Data Set. During ABACKUP processing, tape data sets in the INCLUDE list are mounted and the data is copied to the ABARS output tapes.

4. In the **CATEGORY IF 1ST REF. DELETE** field, specify whether to include or exclude a data set from the Selection Data Set if the first reference is DELETE. Specify one of these options:

Option	Description
I	Include data sets with a disposition of DELETE in the Selection Data Set.
E	(Default.) Exclude data sets with a disposition of DELETE from the Selection Data Set (required for forward recovery).

5. In the **CATEGORY IF 1ST REF. RENAME OLD DSN** field, specify if data sets with a disposition of OLD that are RENAMED to a new name should be included or excluded from the Selection Data Set. Specify one of these options:

Option	Description
I	Include data sets with a disposition of OLD that are RENAMED to a new name.
E	(Default) Exclude data sets with a disposition of OLD that are RENAMED to a new name (required for forward recovery).

   Note: This option is ignored if the **EVALUATE ALL DATASETS AS CRITICAL** option is set to Y.

6. In the **CATEGORY IF 1ST REF. RENAME NEW DSN** field, specify if data sets with a disposition of NEW that are RENAMED to a new name should be included or excluded from the Selection Data Set. Specify one of these options:

Option	Description
I	Include data sets with a disposition of NEW that are RENAMED to a new name.
E	(Default) Exclude data sets with a disposition of NEW that are RENAMED to a new name (required for forward recovery).

   Note: This option is ignored if the **EVALUATE ALL DATASETS AS CRITICAL** option is set to Y.

7. In the **USE SMF DATA IN SELECTION DATASET** field, specify if data collected by RSP SMF Capture or by SMF Scan is to be used when the Selection Data Set is built. Specify one of these options:

Option	Description
Y	(Default) The data is used.
N	The data is not used.

8. In the **USE JCL DATA IN SELECTION DATASET** field, specify if data collected by RSP JCL Capture is to be used when the Selection Data Set is built. Specify one of these options:

Option	Description
Y	(Default) The data is used.
N	The data is not used.

9. In the **SELECTION Dsn Format** field, specify the format of the Selection Data Set generated by Automated Critical Data Identification. Valid options are:

Option	Description
ABARS	(Default) Write the records to the Selection Data Set as INCLUDE, EXCLUDE, ALLOCATE, and ACCOMPANY statements. The Selection Data Set is used as input to ABACKUP.
DSS	Write the records to a data set using DFSMSdss logical DUMP control statements. Data sets in the ACCOMPANY or ALLOCATE lists are not supported by DSS and are ignored.
LIST	Generate a list of data set names, including all data sets that meet the INCLUDE, ACCOMPANY, and ALLOCATE criteria.
DMS	Generate Brightstor CA-Disk logical DUMP control statements for data sets in the INCLUDE list. Data sets in the ACCOMPANY and ALLOCATE lists are ignored.
FDR	Generate FDR logical DUMP control statements for data sets in the INCLUDE list. Data sets in the ACCOMPANY and ALLOCATE lists are ignored.
DSSL	Second format for DSS. Use this format when using the FILTERDD ddname for DSS backups.
CA-DISK	Generate Brightstor CA-Disk logical DUMP control statements for data sets in the INCLUDE list. Data sets in the ACCOMPANY and ALLOCATE lists are ignored.

10. In the **RSP cycles to include in Sel. DS.** field, specify the number of cycles of RSP data to include when building the Selection Data Set. When an application data set is referenced in more than one cycle, Automated Critical Data Identification uses the first reference from the first cycle to determine criticality.

    Note: If multiple cycles of RSP data are required, the RSP option **Number of RSP cycles to retain** must be set to a value that is at least equal to the value specified by the **RSP cycles to include in Sel. DS.** option.

11. In the **EVALUATE ALL DATASETS AS CRITICAL** field, specify whether the application should be defined for forward recovery or rerun recovery. Specify one of these options:

Option	Description
Y	Forward recovery. All input and output data sets are classified as critical (with the exception of the data sets excluded by filters) and the names are written to the Selection Data Set as INCLUDE or ACCOMPANY. **Note:** You can override the critical classification using Universal, Global, or Local filtering.
N	(Default) Rerun recovery. Only data sets that are used as input to the application (that is, data sets that existed before running the application) are considered critical (required for rerun recovery).

    Note: Exception to all data sets considered critical: when using the ABARS format for the Selection Data Set, normally the Selection Data Set "Excluded due to Normal Evaluation" category is for the rerun recovery option. When the **EVALUATE ALL DATASETS AS CRITICAL** option is set to support forward recovery and GDG base names (not generation data sets) are read, the base names are placed in the "Excluded due to Normal Evaluation" category because ABARS does not allow GDG base names in the INCLUDE list (only the ALLOCATE list).

12. In the **DETAILED COMMENTS IN SEL. DATASET** field, specify whether you want to generate additional comments in the Selection Data Set to indicate why a data set was classified in a particular category. Specify one of these options:

Option	Description
Y	Generate additional comments in the Selection Data Set.
N	Do not generate additional comments in the Selection Data Set.

Including or excluding data sets from the Selection Data Set

The **CATEGORY IF 1ST REF. DELETE**, **CATEGORY IF 1ST REF. RENAME NEW DSN**, and **CATEGORY IF 1ST REF. RENAME OLD DSN** options provide the ability to include or exclude a data set from the Selection Data Set if the first reference is DELETE or RENAME.

Automated Critical Data Identification looks at the first reference of each data set and determines criticality based on the data set disposition, data set access, Setup options, and filters in the Automated Critical Data Identification Application Data Inventory database.

Most deleted or renamed data sets do not need to be included in the Selection Data Set because they are no longer cataloged at the end of the application.

The exceptions are:

- RENAMED (OLD) data sets — (If you are using rerun recovery.)
- RENAMED (NEW) data sets — (If you are using the forward recovery **EVALUATE ALL DATASETS AS CRITICAL** option.)

However, it is not uncommon for applications to DELETE or RENAME a data set and in the same step, or a subsequent step, create the data set again as NEW. Therefore, you might want to specify **I** (Include) if deleted or renamed data sets resurface during the application, such as those data sets that could result after a reorganization or rebuilding of the data set. Whether the data set after reinstatement has the same contents as at the beginning of the application is not known to Automated Critical Data Identification.

Note: If the first reference of the data set is DISP=(MOD,DELETE), the file exists and the data set is opened for INPUT. The first reference is INPUT, not DELETE and the data set will be placed on the INCLUDE list. (A data set with DISP=(OLD,DELETE) where the file is not opened is, in fact, a DELETE.)

Specifying Verification GDG options

Verification GDG options control how generation data sets (GDSs) are managed at the end of the application (BKMAPLEN) or during Verification (before they are written to the Selection Data Set).

1. In the **GDS NAMES IN SELECTION DATASET** field, specify whether or not generation data set names in the Selection Data Set should be absolute (.G0000V00s), relative (0, -1, and so on) or both. Valid options are:

Option	Description
R (Relative)	Relative names (PROD.DAILY.PARM(0)).
A (Absolute)	Absolute names (PROD.DAILY.PARM.G0000V00s)
B (Both)	(Default) Relative and absolute names

   Generation data set names are derived from database entries that are obtained from RSP JCL Capture or RSP SMF Capture (when the Current VERIFICATION Options **USE SMF DATA IN SELECTION DATASET** and **USE JCL DATA IN SELECTION DATASET** are set to Y).

   If entries exist from JCL and SMF data (relative and absolute names), and the **GDS NAMES IN SELECTION DATASET** option is set to **A** or **R**, the unwanted names are excluded.

   Absolute GDS names generated from relative Automated Critical Data Identification Application Data Inventory database entries are refreshed from the catalog each time BKMAPLEN rebuilds the Selection Data Set or when you run the Verification program.

   Note: It is recommended that you run BKMAPLEN as soon as the application has ended. Doing so ensures correct resolution based on catalog entries at the end of the application (and before additional processing alters the status of a GDG).

   For a correct resolution of absolute names to relative names, the building of the Selection Data Set must complete before any GDG activity alters the GDG status at the completion of the current cycle.

2. In the **EVALUATE GDG ENTRIES FROM JCL DATA** field, specify whether GDG entries from JCL are evaluated. Specify one of these options:

Option	Description
Y	(Default) GDG entries from JCL are evaluated. GDG entries from JCL are added to the Automated Critical Data Identification Application Data Inventory database using the relative name. Use the **GDS NAMES IN SELECTION DATASET** option to determine how GDGs are written to the Selection Data Set.
N	Bypass evaluation of GDG entries from JCL data.

Note: If **USE JCL DATA IN SELECTION DATASET** is set to **N**, it overrides this option.

3. In the **EVALUATE GDG ENTRIES FROM SMF DATA** field, specify whether or not GDG entries from SMF data are evaluated. Specify one of these options:

Option	Description
Y	(Default) GDG entries from SMF are evaluated. GDG entries from RSP or SMF Scan are added to the Automated Critical Data Identification Application Data Inventory database using the absolute name. Use the **GDS NAMES IN SELECTION DATASET** option to determine how GDGs are written to the Selection Data Set.
N	Bypass evaluation of GDG entries from SMF data.

Note: If **USE SMF DATA IN SELECTION DATASET** is set to **N**, it overrides this option.

4. In the **EXPAND BASE ONLY REFERENCES** field, specify whether or not GDG base-only entries from JCL or INCLUDE external filters should be expanded to include absolute GDS names. RSP SMF Capture generates absolute names from a base name in the JCL. The base name is always excluded. Specify one of these options:

Option	Description
Y	Converts the base name to absolute names.
N	(Default) Causes the base-only reference to be excluded because base only references are not supported by ABARS.

5. In the **ADD ENTRIES FOR MISSING GDSES** field, specify if GDS entries that are *not* present in the database when BKMAPLEN is run, should be added. This option prevents "gaps" in the GDS for GDGs that are referenced by the application. For example: if generations '0' and '-2' are referenced by the application, when restored, the '-2' generation becomes '-1' because the missing generation, '-1' was not referenced and therefore not included in the backup. Specify one of these options:

Option	Description
N	Causes no additional GDS names to be added to the database. Specifying this option will produce a backup, which when restored, could cause the application to fail or produce unexpected results if there are any gaps in the GDSs for a GDG referenced by the application. For example, if the only relative generations referenced by the application are '0' and '-2', backing up just these two generations will cause the '-2' generation to be '-1' after recovery (assuming that the original '-1' generation is not recovered from another source).
M	(Minimum for rerun) Causes all generations from the oldest referenced generation through the current generation ('0' before the application has run) at BKMAPLEN to be added to the database if they were not already present. This option is called "Minimum for rerun" because if any gaps exist in the range of generations referenced, the relative generations will not be correct after a restore. **Note:** The default evaluation of new generations (+1) is 'EXCLUDE' (this is consistent with rerun recovery). With rerun recovery, only input data sets are evaluated as critical and are included in the Selection Data Set. If during a rerun, an application is sensitive to the knowledge that new generations were indeed created; it might be necessary for these new

Option	Description
	generations to be restored. See the **ABARS CATEGORY FOR NEW GENERATIONS** option to force the inclusion of new GDSs on the INCLUDE or ALLOCATE list of the Selection Data Set.
F	(Forward recovery) Causes all generations required for forward recovery to be added to the database if they were not already present at the end of the application. Specifying this option positions the application for processing (forward) into the next cycle after recovery. It will cause the same number of oldest generations to be referenced as there are new generations to be excluded. It forces the choice for new generations to INCLUDE and it also forces the **ABARS CATEGORY FOR MISSING GDSES** option from ALLOCATE to INCLUDE (but only for certain generations). **Note:** This option applies to generation data sets only and does not ensure that all data sets required for forward recovery are included in the Selection Data Set. Forward recovery, in theory, is only dependable for generation data sets. If **EVALUATE ALL DATASETS AS CRITICAL** is set, both inputs and outputs are automatically included. However, using the rerun methodology where only inputs are included might provide forward recovery if you use mostly generation data sets. If, however, output files are not generation data sets but are required for forward recovery, the **EVALUATE ALL DATASETS AS CRITICAL** option should be set to include those output data sets.
A	(All cataloged) Causes all generations found in the catalog to be added to the database if not already present at BKMAPLEN. The generations could still be excluded by a filter. **Note:** Use caution when selecting this option for an application. It is common to overstate the LIMIT of the base catalog entry. This option is not required to force all generations into the Selection Data Set. If the base name was only used in the JCL, and JCL data is being used, then if the **EXPAND BASE ONLY REFERENCES** option is set to Y, it occurs automatically.

When data sets are added to the database using the **ADD ENTRIES FOR MISSING GDSES** option, this option provides the ability to place the added entries on the INCLUDE or ALLOCATE list of the Selection Data Set:

- **I** (Include) — (Default) Places the added entries on the INCLUDE list in the Selection Data Set.
- **A** (Allocate) — Places the added entries on the ALLOCATE list in the Selection Data Set.

Example 1: An application references generations -2, -1, 0 and (creates) generation +1. The -2 generation (which becomes -3 after the job creating +1 ends) is excluded. When the next cycle runs, assuming the same JCL, the input generations -2, -1, and 0 are the previous cycle's -1, 0, and +1 generations respectively.

Example 2: Applications reference generations -5, -4, -3, -2, -1, 0, +1, +1 (two new generations). The -5 and -4 generations are dropped. The same number of oldest generations as there are new generations created are excluded.

6. In the **Category for new GDS if new GDS IS ONLY REFERENCE TO GDG** field, specify one of these options:

Option	Description
I	(Default) New GDSs are included in the Selection Data Set. This is the recommended setting for forward recovery.
E	A new GDS is excluded if no existing GDS for that GDG is referenced.
A	A new GDS is placed on the ALLOCATE list in the Selection Data Set.

Note: This option applies only to forward recovery. It applies to new GDSs.

By default, new data sets are excluded unless the option **EVALUATE ALL DATASETS AS CRITICAL** is set to Y. Specify this option to force new GDSs to be added to the Automated Critical Data Identification Application Data Inventory database and include them in the INCLUDE list (**I**), ALLOCATE list (**A**), or Exclude (**E**) list in the Selection Data Set.

Note: This option is ignored if **EVALUATE ALL DATASETS AS CRITICAL** is set to Y. Both input and output data sets are included automatically.

Forward recovery, in theory, is only dependable for generation data sets. If **EVALUATE ALL DATASETS AS CRITICAL** is set to Y, both inputs and outputs are automatically included. However, using the rerun methodology where only inputs are included might provide forward recovery if you use mostly generation data sets. However, if output files are not generation data sets but are required for forward recovery, the **EVALUATE ALL DATASETS AS CRITICAL** option should be set to include those output data sets.

7. To determine the ABARS Selection Data Set category, in the **ABARS CATEGORY FOR MISSING GDSES** field, specify one of these options:

Option	Description
I	(Default) Include. This is the recommended setting for forward recovery.
A	Allocate.

8. In the **ABARS CATEGORY FOR NEW GENERATIONS** field, specify the ABARS Selection Data Set category. New data sets are normally excluded. Use this option to force the category for new GDSs to INCLUDE or ALLOCATE. Specify one of these options:

Option	Description
I	Include. This is the recommended setting for forward recovery.
A	Allocate.
E	(Default) Exclude. This is the recommended setting for rerun recovery.

Note: This option applies to new GDSs.

How generation data sets are added to the Automated Critical Data Identification Application Data Inventory database

Generation data sets are added to the Automated Critical Data Identification Application Data Inventory database from RSP JCL Capture and RSP SMF Capture or from SMF Scan.

GDSs that are added from RSP JCL Capture are added to the Automated Critical Data Identification Application Data Inventory database using relative names (GDGbase(*nn*)). The relative name is adjusted based on the catalog entries when you run BKMAPLEN or Verification and a catalog lookup occurs.

Table 2. Example - GDSs added from RSP JCL Capture

Job name	Relative GDG in JCL	Relative GDG in Automated Critical Data Identification Application Data Inventory database when BKMAPLEN or Verification is run
JA001	PROD.DAILY.PARM(-2)	PROD.DAILY.PARM(-3)
JA001	PROD.DAILY.PARM(-1)	PROD.DAILY.PARM(-2)
JA001	PROD.DAILY.PARM(0)	PROD.DAILY.PARM(-1)
JA001	PROD.DAILY.PARM(+1)	PROD.DAILY.PARM(-0)

GDSs added from RSP SMF Capture or SMF Scan are added using absolute names: gdgbase.G*nnnn*V*nn*.

```
PROD.DAILY.PARM.G0004V00
PROD.DAILY.PARM.G0003V00
PROD.DAILY.PARM.G0002V00
PROD.DAILY.PARM.G0001V00
```

When both RSP JCL Capture and RSP SMF Capture (or SMF Scan) are used to collect application data set records, both the relative and the absolute names are stored in the Automated Critical Data Identification Application Data Inventory database.

The **GDS NAMES IN SELECTION DATASET** option determines if relative names or absolute names (or both) are used in the Selection Data Set.

Specifying Selection Data Sets options

Selection Data Set options specify the Selection Data Set names used by RSP and Verification.

Two Selection Data Set names are stored for each application. One data set name is used by RSP (**RSP/BATCH**) and the other is used by Verification (**VERIFY**). You can specify the same data set name for both options.

Automated Critical Data Identification determines the default Selection Data Set name using one of these methods:

- If you are using ABARS, Automated Critical Data Identification determines whether an aggregate has been associated with the application, and if so, establishes the Selection Data Set name from the ISMF data supplied when the aggregate was defined.
- If no ISMF aggregate name is identified, Automated Critical Data Identification generates a name to be used with non-ABARS formats, or alternatively, you can create your own 80-byte sequential data set.

Note: If you have not selected a Selection Data Set name, or if this is the first time you have accessed the **Application SETUP Options** panel to define a new application, default data set names appear in the **SELECTION DATASETS** fields.

You can change the names so that RSP and Verification builds different Selection Data Sets. Additionally, when Verification is selected, a third Selection Data Set name can be specified for a single build.

Note: If you are using ABARS, the ABARS Selection Data Set name appears in these fields. Otherwise, if you are not using ABARS, or no aggregate exists, a default name is used.

1. In the **VERIFY** field, type the Selection Data Set name you want to use for foreground processing.
2. In the **RSP/BATCH** field, type the Selection Data Set name you want to use for RSP or batch processing.
3. Press Enter and then press PF3 to create the Selection Data Set.

Chapter 10. Performing a job collection

After you define an application to Automated Critical Data Identification, you must identify the jobs that belong to the application.

Job collection is the process of identifying which jobs belong to an application and then placing the information in the Automated Critical Data Identification Application Data Inventory database.

For each initial job collection event (except when you are using job name masks that are stored directly in the Automated Critical Data Identification Application Data Inventory database), Automated Critical Data Identification submits a batch job to interface with a specified job scheduling product or an input data set containing a list of job names. The batch job collects the job elements (job entries and job structures) and saves them in the Automated Critical Data Identification Application Data Inventory database. Optionally, you can define application-specific backup jobs to be used with the Backup End function and application cycle start or cycle end jobs (or both) to be used with the CYCLE CONTROL JOB option.

1. Identify the job elements associated with the application.
2. Determine a starting point for collection.
3. Select a job collection method:
   - An interface to a job scheduler product
   - An input data set (using ISPF Edit, a pre-staged data set, or a user-supplied data set)
4. Follow the appropriate procedure for the selected job collection method.

When job elements are saved to the Automated Critical Data Identification Application Data Inventory database, the Setup options defined for the current application, including job filtering masks, are applied. You can submit the job collection at any time to modify job entries. You can also use ISPF Edit to modify job elements previously collected using a job scheduling product or a pre-staged data set.

Job elements

Job elements are the job entries and job structures associated with a specific application. An application can consist of a single job entry, multiple job entries, or a series of job entries with job structures that control complex scheduling requirements.

Job entries

RSP and SMF Scan use *job entries* during the initial job collection. (SMF Scan uses job entries for SMF record matching purposes.) The data sets referenced during the job collection process are automatically saved in the Automated Critical Data Identification Application Data Inventory database.

Job lists

A *job list* is a set of job names, associated with an application, that do not have predecessor or successor relationships. Automated Critical Data Identification can automatically obtain the job names from the job scheduling product specified for the application, or from a job name list supplied to Automated Critical Data Identification as an input data set.

Job structure

A *job structure* is a set of job names, associated with an application, that have a prescribed order of predecessor or successor dependencies. A job structure has an initial job, called the *anchor job*, that is used to control the scheduling sequence of the succeeding jobs.

Inside of a job structure, any one of the succeeding jobs can be another anchor job for a lower level job structure forming a complete application schedule.

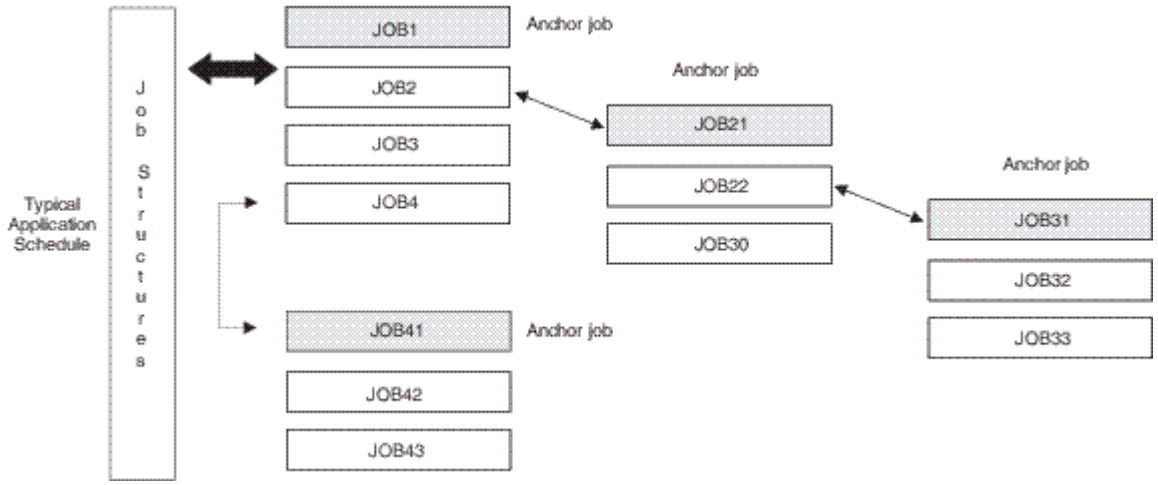


Figure 3. Example job structure

In Figure 3 on page 88, the application consists of four job structures, each with its own specific anchor job points (JOB1, JOB21, JOB31 and JOB41). Using the JOB1 structure as an example, its successor jobs are JOB2 through JOB4. JOB41's predecessor is JOB4 (in the initial JOB1 structure).

The application might be a set of payroll jobs that have both daily and weekly cycle runs. If a daily cycle is used, then only job structures JOB1, JOB21 and JOB31 are processed. If a weekly cycle is used, then only JOB1 and JOB41 structures are processed. Therefore, the JOB1 structure is common to both cycles.

Identifying the job elements associated with an application

Identifying the job elements for a specific application is fundamental to performing a successful job collection.

- To perform a job collection for a specific application, identify these job elements:
  a) Job entries
  b) Job lists
  c) Job structures

Determining a starting collection point

Because job collection allows the specification of one or more job structure names or anchor job names, you must define a starting point for the job collection.

In the example shown in Figure 3 on page 88, if JOB1 is used as the starting collection point, then the resulting application job structure would consist of the four structures and their job names as the job entries. If JOB41 is used as the starting anchor point, then only one structure (the JOB41 elements) would be collected for the application.

In the example, Automated Critical Data Identification allows the application schedule to be separated into two applications (for example, Daily Payroll and Weekly Payroll) by using their unique anchor job name for the starting collection point. Potentially large applications can be separated into smaller sub-applications by using this technique during job collection.

The advantages of this approach are:

- Application backup can be scheduled independently and concurrently with any other application backups.
- You have the capability to back up only that portion of the application as needed, (for example, daily application backups as compared to weekly application backups).

Using a job scheduling interface to perform job collection

When you use a job scheduling interface, Automated Critical Data Identification automatically invokes the job scheduling interface and reads the output contents of a scheduling report to obtain the required application information.

Before you begin, ensure that you perform these tasks:

- Identify the job elements (job structures and job names) associated with the application.
- Determine the starting collection point.

1. Set configuration values for the specific job scheduler you are using in the BKMINI product parameter library member.

   BKMINI contains configuration and global options used by Automated Critical Data Identification during job collection and by other interfaces to job scheduling products.

   Note: To change an option, use ISPF Edit to modify the BKMINI product parameter library member. Any changes you make are implemented during the next job collection request.
2. (Optional) Set up job name filtering.
3. Follow the steps to perform job collection using the appropriate job scheduler interface.

The job collection process using a job scheduling interface

Using a job scheduling interface for the job collection process involves extracting application job name information from the job scheduler database and updating the application job name list in the Automated Critical Data Identification Application Data Inventory database.

When you use a job scheduling interface, the job collection process is performed by running a batch job you create through the **1 (JOB COLLECT)** option. You must update the application job name list generated by this option each time you add a new job to the application.

Tip: If you regularly add new jobs to your applications, you might find it helpful to save the job collection batch jobs for future use.

Updating the job name list

Important: When an application job name list is changed, be sure to refresh the RSP job table by running the job in member BKMJRFSH of the product JCL library.

This example shows a sample job to refresh the job name list for multiple applications using the ZEKE MVS scheduler.

```
//JOBCOLL    JOB .........
//STEP1    EXEC ZEKEUTL
//STEP2    EXEC BKMZI450,ZPGMI=BKMZI510
//JOBZK    EXEC BKMJOBZK,ZPGM=BKMZK450,APPL=appl1
//JOBCOLL    EXEC BKMJOBSQ
//JOBZK    EXEC BKMJOBZK,ZPGM=BKMZK450,APPL= appl2
//JOBCOLL    EXEC BKMJOBSQ
.............................................
.............................................
//JOBZK    EXEC BKMJOBZK,ZPGM=BKMZK450,APPL= appln
//JOBCOLL    EXEC BKMJOBSQ
```

Updating multiple job name lists

If a number of application job name lists must be updated, you can perform the scheduler database extraction once and then build the job name list for each application. (However, you must run the last two steps of the job collection jobs for each application.)

Performing job collection with Unicenter CA-7

To use this option, your installation must have Unicenter CA-7 installed and the jobs that comprise the application must be under the control of Unicenter CA-7.

Before you perform job collection with Unicenter CA-7, verify that values have been set for the Unicenter CA-7 keywords in the BKMINI product parameter library member's ACDI section:

- **CA7ALTSCHID**
- **CA7COLLECTOPTIONS**
- **CA7LOGONNAME**
- **CA7LOGONPASS**
- **CA7PROCNAME**
- **CA7PROCSTEPNAME**
- **CA_INCLUDE_ALTSCHID**
- **JOBCOLL_CA7_UID**
- **JOBCOLL_DROP_ALL_UNRESOLVED_PREDECESSORS**

Automated Critical Data Identification obtains information about the structure of an application from CA-7 by submitting a batch job that logs onto the CA-7 batch terminal processor to issue the **CA-7 FSTRUC** (Forecast Structure) command and **LJOB** command.

1. From the **Application AutoBuild Options** panel, select the **JOB COLLECT** option.

   The **APPLICATION JOB COLLECTION OPTIONS** panel is displayed.
2. From the **APPLICATION JOB COLLECTION OPTIONS** menu, select the **CA7** option.

   The **Supply Application CA7 Information** panel is displayed. The CA-7 anchor job names you specify on this panel are used to determine which jobs belong to the application.
3. Type the Unicenter CA-7 Scheduling ID (1–255) for this collection event in the **Current (Default) SCHID** field. The default is 1.
4. In the **CA7 UserID Name** field, type the user ID (maximum of eight characters) to be used by Automated Critical Data Identification when logging on to the Unicenter CA-7 batch terminal processor.

   Note: You must specify a user ID. The CA-7 user ID is defined in BKMINI as CA7LogonName.
5. In the **CA7 Password** field, type the password (maximum of eight characters) to be used by Automated Critical Data Identification when logging on to the Unicenter CA-7 batch terminal processor.

   Note: You must specify a password. The CA7 password is defined in BKMINI as CA7LogonPass.
6. In the **Supply ANCHOR Job Names Below** area, specify a maximum of 15 job names, or job name masks, for which Unicenter CA-7 is to return the job structure to Automated Critical Data Identification.

   Note: You must specify at least one job name or mask.
7. Press Enter to continue or press F3 to exit.
8. From the **JOB Statement Verification For Batch Submit** panel, review the job card specification and make any changes, if needed.
9. Press Enter to continue or press F3 to exit.
10. Select from these options to process the batch job:
    - To edit the JCL prior to submitting the job, type S in the **EDIT JCL Prior to Submit** field and press Enter.
    - To submit the batch job, type S in the **SUBMIT JOB For Execution** field and press Enter.
    - Press F3 to exit without submitting the job.

Sample Unicenter CA-7 batch collection job

This information provides an example of the job collection procedure generated through the **CA7** job collection option on the **APPLICATION JOB COLLECTION OPTIONS** panel.

```
  //APPLJRFF JOB (ACCT),,'ABC
  R',CLASS=A,MSGCLASS=X,NOTIFY=&SYSUID
  //*
  //*
  //JCLL JCLLIB ORDER=(prefix.value.SBKMJCL )
  //  EXEC CA73BTI
  //BTERM.SYSPRINT DD DISP=OLD,DSN=PDUSER.APPL.CA7OUTPT,
  //      DCB=BLKSIZE=1330
  //BTERM.SYSIN DD *
  /LOGON UCC7JOBS,UCC7JOB
  //*
  //*   APPL NAME = APPL
  //*    JOB TYPE = CA7 JOB COLLECTION
  //*     SUBMIT BY PDUSER  ON 11/17/2009 14:20:52
  //*
  // IF (RC < 5 ) THEN
  //JOBCA7  EXEC BKMC7JOB,APPL=APPL,
  // RPT=PDUSER.APPL.CA7OUTPT
  // ENDIF
  // IF (RC < 5 ) THEN
  //JOBCOLL EXEC BKMJOBSQ
  // ENDIF
```

Parameters

APPL ***appl***
The application name.

RPT
The location of the report output.

Sample Unicenter CA-7 job collection results

This example shows a batch JOBRPT output produced from a Unicenter CA-7 job collection event.

```
        ORIGINAL JOBS          FOR APPLICATION: ABTEST1  2004/12/15 08:50:28

 JOBNAME       PREDECESSOR
 --------      -----------

 ABCDAILY
   ABCD1000    ABCDAILY
   ABCD1100    ABCD1000
   ABCD1150    ABCD1100
   ABCD2000    ABCD1150
   ABCD2200    ABCD2000
   ABCD2400    ABCD2200
ABCDAILY HAS      6 SUCCESSOR JOBS

ABCREQ1
ABCREQ1  HAS NO SUCCESSOR JOBS

ABCREQ2
ABCREQ2  HAS NO SUCCESSOR JOBS
```

Performing job collection with CONTROL-M

To use this job collection method, your installation must have CONTROL-M installed and the jobs that comprise the application must run under the control of CONTROL-M.

Before you perform job collection with CONTROL-M, verify that values have been set for the CONTROL-M keywords in the BKMINI product parameter library member's :ACDI section:

- **CTLM_INPUTUTLJCL**
- **CTLM_SCHEDFILES**

- **JOBCOLL_CTLM_VERSION**
- **JOBCOLL_DROP_ALL_UNRESOLVED_PREDECESSORS**
- **JOBCOLL_EXCLUDE_DEP**

Automated Critical Data Identification uses the CTMRFLW CONTROL-M utility EXEC to create a job flow report based upon input scheduling files and table member names specified on the **Application SETUP Options** panel. Automated Critical Data Identification automatically constructs the appropriate job to complete the job collection event.

The scheduling files and their table member names are defined by the CONTROL-M product coordinator. They can contain multiple scheduling files (as input) for any application defined to Automated Critical Data Identification. You can also define standard scheduling files through definitions in the BKMINI product parameter library member.

1. From the **Application AutoBuild Options** panel, select the **JOB COLLECT** option.

   The **APPLICATION JOB COLLECTION OPTIONS** panel is displayed.
2. From the **APPLICATION JOB COLLECTION OPTIONS** menu, select the **CONTROL-M** option.

   The **Supply Application CTL/M Information Fields** panel is displayed.
3. In the **CTL/M Scheduling File Names** and the **Table Member Names** fields, type the names of the scheduling files and table members that correspond to the CONTROL-M job entries you want to collect. You can specify a total of 15 entries.

   Note: If this is the initial job collection event, the scheduling file names are either blank or contain the names of the data sets defined using the **CTLM_SCHEDFILES** value specified in the BKMINI product parameter library member. If job entries have been previously collected, the scheduling file names and table members shown are the current and previous values specified for the application.
4. Press Enter to save and activate your entries.
5. From the **JOB Statement Verification For Batch Submit** panel, review the job card specification and make any changes, if needed.
6. Press Enter to continue or press F3 to exit.
7. Select from these options to process the batch job:
   - To edit the JCL prior to submitting the job, type S in the **EDIT JCL Prior to Submit** field and press Enter.
   - To submit the batch job, type S in the **SUBMIT JOB For Execution** field and press Enter.

     Note: The JCL has two parts:

     a. First, the CTMRFLW CONTROL-M collection utility is executed. The scheduling file names and table members supplied from the input panel are concatenated as input under the DAJOB DD name. The output file is pre-allocated and supplied as the DAREPORT DD name value.

     b. Automated Critical Data Identification then uses the output generated by CTMRFLW as input to continue job collection processing.
   - Press F3 to exit without submitting the job.

Sample CONTROL-M batch collection job

The CONTROL-M job scheduling product requires you to submit an Automated Critical Data Identification constructed job, (generated through the **CONTROL-M** job collection option on the **APPLICATION JOB COLLECTION OPTIONS** panel) which produces a CONTROL-M job flow report. The output of the job flow report is used as input to the second step to save the job entries in the Automated Critical Data Identification Application Data Inventory database.

Sample

```
//JCLL JCLLIB ORDER=(prefix.value.SBKMJCL)
//         INCLUDE MEMBER=IOASET
//CTMRFLW  PROC
//STEP1    EXEC PGM=CTMRFLW,REGION=&REG
```

```
//         INCLUDE MEMBER=&IOAENV
//DAALOCIN DD DISP=SHR,DSN=&DAALOCIN(ALCMRFLW)
//SCHOUT   DD SPACE=(TRK,(300,300)),UNIT=&WORKUNIT
//DAJOB    DD DDNAME=&SOURCE
//SCHEDULE DD DISP=SHR,DSN=&OLPREFM..SCHEDULE(&FLOWMEM)
//CKP      DD DISP=SHR,DSN=&DBPREFM..CKP
//SYSPRINT DD SYSOUT=&OUT
//DAFLWPRM DD DDNAME=SYSIN
//DAREPORT DD SYSOUT=&OUT            REPORTS SYSOUT
//DACHART  DD SYSOUT=&OUT            FLOW CHART SYSOUT
//DACHART1 DD SYSOUT=&OUT,DCB=(OPTCD=J),
//            CHARS=(&CHARSET,&CHARST2)
//DACHART4 DD SYSOUT=&OUT,DCB=(OPTCD=J),
//            CHARS=GT24             VERY SMALL LETTERS
//DACHART9 DD SYSOUT=&OUT,DCB=(OPTCD=J),
//            CHARS=(&CHARSET,FM99)  BOLD BOX ELEMENTS
//SYSOUT   DD SYSOUT=&OUT
//DATRACE  DD SYSOUT=&OUT
//SYSABEND DD SYSOUT=&OUTDUMP
// PEND
//          EXEC  CTMRFLW
//DAREPORT DD DISP=SHR,DSN=PDUSER.APPL.CLMOUTPT
//DAJOB    DD DISP=SHR,DSN=FILE1(NAME1)
//         DD DISP=SHR,DSN=FILE2(NAME2)
//         DD DISP=SHR,DSN=FILE3(NAME3)
//*
//*   APPL NAME = APPL
//*    JOB TYPE = CLM JOB COLLECTION
//*     SUBMIT BY PDUSER  ON 12/01/2009 09:30:29
//*
// IF (RC < 5 ) THEN
//JOBCM   EXEC BKMCM606,APPL=APPL,
// MPGMI=BKMCM606,
// RPT=PDUSER.APPL.CLMOUTPT
// ENDIF
// IF (RC < 5 ) THEN
//JOBCOLL EXEC BKMJOBSQ
// ENDIF
```

In the example, three scheduling files and their table members have been concatenated as the DAJOB DD name. The pre-allocated output file is assigned to the DAREPORT DD name.

Note: The CTMRFLW utility JCL used should not contain the DAREPORT and DAJOB DD name statements. If they exist, they will be overridden by the Automated Critical Data Identification constructed statements shown above.

Parameters

APPL ***appl***
The application name.

MPGMI
The job collection procedure.

RPT
The location of the job collection output.

Sample CONTROL-M application job list

This example shows a sample CONTROL-M application job list. You can display the application job list using the **STATUS** option on the **Application AutoBuild Options** panel.

When you select the **Display Current JOB List** option from the **APPLICATION Display Selections** panel, the **Application JOB Sequence List** panel is displayed. The **Misc Descriptive Info** field on the **Application JOB Sequence List** panel shows the contents of the CONTROL-M Table Job Flow report output as extracted for each job entry listed on the left. The job name entries in the first column are the current contents of the scheduling files and table members used as input and generated from the CTMRFLW utility program. For example:

```
Row 1 to 14 of 582
   Command ==>                  Application ==> CTLMAPPL
                          Collection Method ==> CONTROL-M Table Entries


  -- Jobname -- -- Misc Descriptive Info -------------

     ADABKUAT    1 ADABKUAT          ODATE 010725   TIME FROM 1900 UNTIL
     ADABKUCJ    1 ADABKUCJ          ODATE 010725   TIME FROM 0400 UNTIL
     ADABKUHS    1 ADABKUHS          ODATE 010725   TIME FROM 0200 UNTIL
     BACKUPM     1 BACK UPM          ODATE 010723
     BECIDOWN    3 BECIDOWN BECISHUT ODATE 010725
     BECIJRN4    2 BECIJRN4 BECISHUT ODATE 010725
     BECISHUT    1 BECISHUT          ODATE 010725   TIME FROM 2209 UNTIL
     BECIUP      1 BECIUP            ODATE 010725   TIME FROM 0601 UNTIL
     BELLAS01    1 BELLAS01          ODATE 010725   TIME FROM 0130 UNTIL
     BELLAT00    1 BELLAT00          ODATE 010725   TIME FROM 0745 UNTIL
     BEMASSCT    2 BEMASSCT BECISHUT ODATE 010725
     BEMASSS1    1 BEMASSS1          ODATE 010725   TIME FROM 0159 UNTIL
     BEMASS1T    1 BEMASS1T          ODATE 010722   TIME FROM 0603 UNTIL
```

Sample CONTROL-M application anchor job structures list

This example shows a sample application anchor job structure list. You can display the anchor job structure list using the **STATUS** option on the **Application AutoBuild Options** panel.

When you select the **Display Current JOB Anchor Chains** option from the **APPLICATION Display Selections** panel, the **Application ANCHOR Job Structures** panel is displayed. The **Job Structure Elements** field on the **Application ANCHOR Job Structures** panel lists the successor jobs for the job named in the **JOB/Anchor** field. This information is extracted from the CTMRFLW output report using the **DEPEND ON** field.

Note: Job entries that have the same name as a successor are assigned a name by Automated Critical Data Identification.

```
Command ==>                                          Row 183 to 196 of 625
   Valid Commands => NOte, SAVE and SORT(c)  APPLICATION ==> CTLMAPPL

   - JOB/Anchor - - Type - -- JOB Structure Elements -----------------

      FINETD        CTL/M   FINETD
      FIPANBKP      CTL/M   FIPANBKP
      FIPAYCD       CTL/M   FIPAYCD
      FIPAYEXC      CTL/M   FIPAYEXC
      FIPDTANK      CTL/M   FIPDTANK
      FIREOLED      CTL/M   GFSBGNDF
      FISCHED       CTL/M   FISCHED
      FISHOUT       CTL/M   FISHOUT
      FISTD         CTL/M   FISTD
      FIVERIFY      CTL/M   FIBKUPD2 FIBKUPD3 FIBKUPD4 FIBKUPD5 FIBKUPD6
                            FIBKUPD7
      FIVERIF1      CTL/M   FIBKUPD0 FIBKUPD1 XIBKUPD1
      FIVERIF2      CTL/M   FIBKUPI0 FIBKUPI1
      FIVP8J02      CTL/M   FILISTD2
```

Sample CONTROL-M batch SYSPRINT report

This example shows sample job entries in the CONTROL-M batch SYSPRINT file.

Sample job entries

An example of the job entries in the batch SYSPRINT file from the last collection job is shown below. Information contained in the file has been extracted from the CTMRFLW output report.

```
CURRENT DATE = 11/15/11  CURRENT TIME = 09:46:32                PAGE = 2

                     J O B S C A N   Output  Summary:


 Locating JOB List For Application: CTLMAPPL
 Job Collection Type: REPLACE
             Options: NONE
   Job Entries Saved: By ALL Jobs
    Filtering Status: ENABLED
  Initial Anchor Job: CDINV04

JOB NAMES REMOVED BY FILTERING: WEU00617 WEU00301 WEU00461 WEU0461Y


  Job Name:    CONTROL-M Scheduling ( Job Flow Report ) Data:
  ---------    ---------------------------------------------

  HLPCO041     1 HLPCO041          ODATE 010421   TIME FROM 0040 UNTIL 0500
  SWTOM13      1 SWTOM13           ODATE 010723   TIME FROM 0700 UNTIL
  SWTOM14      1 SWTOM14           ODATE 010723   TIME FROM 0700 UNTIL
  LMCS         1 LMCS              ODATE 010725   TIME FROM 1100 UNTIL
  CCRF         1 CCRF              ODATE 010725   TIME FROM 1130 UNTIL
  CASCHED      1 CASCHED           ODATE 010725
  CDSCHED      1 CDSCHED           ODATE 010725
  CELSCHED     1 CELSCHED          ODATE 010725
  CRSCHED      1 CRSCHED           ODATE 010725
  BESCHED      1 BESCHED           ODATE 010725
  DBSCHED      1 DBSCHED           ODATE 010725
  DPACFSCH     1 DPACFSCH          ODATE 010725
  DPCOTSCD     1 DPCOTSCD          ODATE 010725
  DPJSCHED     1 DPJSCHED          ODATE 010725
  FICTMSCH     1 FICTMSCH          ODATE 010725
  SWSCHED      1 SWSCHED           ODATE 010725
  DPJHABND     1 DPJHABND          ODATE 010725
  TESTJOB2     1 TESTJOB2          ODATE 010725   TIME FROM 1000 UNTIL
  SMSBKLRG     1 SMSBKLRG          ODATE 010725   TIME FROM 0300 UNTIL
  WEDB2WEP     1 WEDB2WEP          ODATE 010725
  TESTJOB3     1 TESTJOB3          ODATE 010725   TIME FROM 1105 UNTIL
  SMSBKSTD     1 SMSBKSTD          ODATE 010725   TIME FROM 0300 UNTIL
```

Sample anchor structure entries

An example of the anchor structure entries in the batch SYSPRINT file from the last collection job is shown below. Information contained in the file has been extracted from the CTMRFLW output report.

```
J O B  SCHEDULING - Anchor/DEP Job List

 DEP Job:     Successor Job/Member Name List:
 ---------    -------------------------------------------

 CDINV04     CDLOC03
 DPPM2CAT    DPPM2DLY
 HLPNITLY    HLPCS007
 BYSERIAL    SORTSERL
 SORTSERL    SER#LIST
 SER#LIST    FTPJCLT
 FTPJCLT     EISEDIT
 BECISHUT    BECIJRN4 BE14RQPO BE22R2BA BEMASSCT BECIDOWN
 BEMASSCT    BECIDOWN
 CDCICSCL    CDDAILY
 CDDAILY     CDCICSOP
 COTMSDLY    VAULT
 KMDVAL      KMDUJIGF
 CIMSDCO2    CIMSBULK CIMSDISK
 KMDADST9    KMDADSTM
 KMDADSTM    KMDADSTG
 KMDLZR      KMDDAMS
 KMDCCIR     KMDBKUPW
 KMDBKUPW    KMDWKRPT ADAROLL
 KICKOFF     FIREOLED
 STARTUP     XIMIS    XISHOUT  XILDPAJV XILDEMP  XILDVINT XILDERIS XIGS8JW0
             XFSEPNY  XFSINDR  XFSLEAV  XFSSHOP  XFSNCPR  XIAD1JIC XIADVSM
             XIADCGM  XFSFADP  XFSX002  XFSFALS  XFSX003  XIBKUPI2 XFSEQPT
             XFSIN91  XFSINF3  XFSINRP  XIBKUPMB XFSMBILL XFSRBILL XFSSR50
             XFSSR51  XIBKUPFB XFSFBILL XFSFHWT  XFSSR52  XFSSR54  XFSAM82
             XFSA656  XFSA601  XFSIN40  XFSIN70  XIIATADD FIBKUPD1 FIMIS
             FISHOUT  FILDSUSF FILDPAJV FILDEMP  FILDVINT FILDERIS FIIATADD
             FIGS8JW0 GFSINR2  GFSEPNY  GFSINDR  GFSLEAV  GFSSHOP  GFSRECJ
             GFSARCH  GFSNCP   GFSNCPR  FIAD1JIC FIADVSM  FIADCGM  GFSFADP
             GFSX002  GFSFALS  GFSX003  FIBKUPI2 GFSEQPT  GFSRESJ  GFSINB2
             GFSIN91  GFSINF3  GFSINRP  FIBKUPMB GFSMBILL GFSRBILL GFSSR50
             GFSSR51  FIBKUPFB GFSFBILL GFSFHWT  GFSSR52  GFSSR54  GFSAM82
             GFSA656  GFSA601  GFSIN40  GFSIN70
 FIREOLED    GFSBGNDF
```

Sample CONTROL-M job collection reports

This information shows an example of CONTROL-M job collection reports which display the structure of the jobs.

In this example, job ABCJOB03 is succeeded by the ABCJOB02, ABCJOB04, and ABCJOB01 jobs. Jobs ABCJOB13 and ABCJOB23 are independent from ABCJOB03 and have successor jobs:

```
        ORIGINAL JOBS FOR APPLICATION: CONTROLM1  2012/05/13 08:31:18

 JOBNAME          CTLM GROUP           PREDECESSOR
 -------          ----------           -----------

 ABCJOB03         XXXDMN1
   ABCJOB02       XXXDMN1              ABCJOB03
   ABCJOB04       XXXDMN1              ABCJOB02
   ABCJOB01       XXXDMN1              ABCJOB04
 ABCJOB03 HAS        3 SUCCESSOR JOBS

 ABJOB13          XXXDMN2
   ABCJOB14       XXXDMN2              ABCJOB13
 ABJOB13  HAS        1 SUCCESSOR JOBS

 ABJOB23          XXXDMN3
   ABCJOB24       XXXDMN3              ABCJOB23
 ABJOB23  HAS        1 SUCCESSOR JOBS
```

If job filters exist for the application, a report is also created. The following figure shows an example of this type of report.

```
       JOBS AFTER FILTERING FOR APPLICATION: CONTROLM1  2012/05/13 08:31:18

JOBNAME          CTLM GROUP           PREDECESSOR
-------          ----------           -----------

ABCJOB03         XXXDMN1
  ABCJOB02       XXXDMN1              ABCJOB03
  ABCJOB04       XXXDMN1              ABCJOB02
ABCJOB03 HAS          2 SUCCESSOR JOBS

ABJOB13          XXXDMN2
  ABCJOB14       XXXDMN2              ABCJOB13
ABJOB13  HAS          1 SUCCESSOR JOBS

ABJOB23          XXXDMN3

  ABCJOB24       XXXDMN3              ABCJOB23
ABJOB23  HAS          1 SUCCESSOR JOBS
```

Performing job collection with Cybermation ESP

To use this job collection method, your installation must have Cybermation ESP installed and the jobs that comprise the application must run under the control of Cybermation ESP.

Before you perform job collection with Cybermation ESP, verify that values have been set for the Cybermation ESP keywords in the BKMINI product parameter library member's :ACDI section:

- **JOBCOLL_ESP_VERSION**

Automated Critical Data Identification uses the ESP LSAR (List Scheduled Activity Report) to obtain job names from Cybermation ESP.

Note: The LSAR report must have the following DCB attributes:

- RECFM=FB
- LRECL=133
- BLKSIZE=27930

1. From the **Application AutoBuild Options** panel, select the **JOB COLLECT** option.
   The **APPLICATION JOB COLLECTION OPTIONS** panel is displayed.
2. From the **APPLICATION JOB COLLECTION OPTIONS** menu, select the **ESP** option.
   An ISPF Edit session is displayed.
3. Specify the job collection criteria:
   - ESP_PREFIX — Event prefix or mask.
   - ESP_EVENT — Event name or mask.
   - ESP_APL— ESP application or mask.
   - JOB — Job name or mask.

   For example:

   ```
   ESP_PREFIX  PFX1 PFX2
   ESP_EVENT   EVENT_001 EVENT_002 EVENT_1*
   ESP_APL     AP*
   JOB         AC1*E* AC1*F* AC1*G*
   JOB         AC2*
   ```
4. Press F3.
5. From the **JOB Statement Verification For Batch Submit** panel, review the job card specification and make any necessary changes.
6. Press Enter to continue or press F3 to exit.
7. Select from these options to process the batch job:
   - To edit the JCL prior to submitting the job, type S in the **EDIT JCL Prior to Submit** field and press Enter.

- To submit the batch job, type S in the **SUBMIT JOB For Execution** field and press Enter.
- Press F3 to exit without submitting the job.

Sample Cybermation ESP batch collection job

This information provides an example of the job collection procedure generated through the **ESP** job collection option on the **APPLICATION JOB COLLECTION OPTIONS** panel.

Sample

```
 //APPLJRFH JOB (ACCT),,'ABC
R',CLASS=A,MSGCLASS=X,NOTIFY=&SYSUID
 //*
 //*
 //JCLL  JCLLIB ORDER=prefix.value.SBKMJCL
 //STEP1   EXEC BKMESPEX,EPGMI=BKMEI530
 // IF (RC < 5 ) THEN
 //ESPJB   EXEC BKMESJOB,APPL=APPL
 // ENDIF
 // IF (RC < 5 ) THEN
 //JOBCOLL EXEC BKMJOBSQ
 // ENDIF
```

Parameters

EPGMI
Specifies the program name to pass to PROC BKMESPEX.

APPL ***appl***
The application name.

Sample Cybermation ESP job collection report

This information provides an example of a Cybermation ESP job collection report.

```
ORIGINAL JOBS FOR APPLICATION: ESPTST 2010/02/08 13:59:20 Page: 1

JOBNAME ESP PREFIX ESP EVENT PREDECESSOR
-------- ---------- ---------------- -----------

CN1911IX STCUCC7 CSSPDAN

CN1911PC STCUCC7 CSSPDAN

CN1913IX STCUCC7 CSSPDAN

CN1913PC STCUCC7 CSSPDAN

CN1915IX STCUCC7 CSSPDAN

CN1915PC STCUCC7 CSSPDAN

CN1917IX STCUCC7 CSSPDAN
```

Performing job collection with Unicenter CA-Jobtrac

To use this job collection method, your installation must have Unicenter CA-Jobtrac installed and the jobs that comprise the application must run under the control of Unicenter CA-Jobtrac.

Before you perform job collection with Unicenter CA-Jobtrac, verify values have been set for the Unicenter CA-7 keywords in the BKMINI product parameter library member's :ACDI section:

- **JOBTRACSCHEDOUT**

Automated Critical Data Identification obtains information about the structure of an application by submitting a batch job that queries the Unicenter CA-Jobtrac history file for applications scheduled for the current day.

The Unicenter CA-Jobtrac job collection is controlled by the **JOBTRACSCHEDOUT** BKMINI keyword. It provides the flexibility of running the GJTRMAIV utility to collect the current Jobtrac schedule before the job collection and directing Automated Critical Data Identification to process this file instead of generating the GJTRMAIV utility statements.

1. From the **Application AutoBuild Options** panel, select the **JOB COLLECT** option.

   The **APPLICATION JOB COLLECTION OPTIONS** panel is displayed.
2. From the **APPLICATION JOB COLLECTION OPTIONS** panel, select the **JOBTRAC** option.

   The **JobTRAC Collection Information** panel is displayed. The **Pre-Defined Output File** field displays the name of the file specified for the **JOBTRACSCHEDOUT** value in BKMINI. Automated Critical Data Identification uses this file as input for all applications requiring the Unicenter CA-Jobtrac job collection event.
3. In the **Supply JobTRAC ANCHOR Job Names Below** area, specify a maximum of nine job names, or job name masks, for which Unicenter CA-Jobtrac is to return the job structure to Automated Critical Data Identification.

   Note: You must specify at least one job name or mask.
4. Press Enter to continue or press F3 to exit.
5. From the **JOB Statement Verification For Batch Submit** panel, review the job card specification and make any changes, if needed.
6. Press Enter to continue or press F3 to exit.
7. Select from these options to process the batch job:
   - To edit the JCL prior to submitting the job, type S in the **EDIT JCL Prior to Submit** field and press Enter.
   - To submit the batch job, type S in the **SUBMIT JOB For Execution** field and press Enter.
   - Press F3 to exit without submitting the job.

Sample Unicenter CA-Jobtrac batch collection job

This information provides an example of the job collection procedure generated through the **JOBTRAC** job collection option on the **APPLICATION JOB COLLECTION OPTIONS** panel.

Sample

```
//APPLJRFF JOB (ACCT),,'ABC
R',CLASS=A,MSGCLASS=X,NOTIFY=&SYSUID
//*
//*
//JCLL JCLLIB ORDER=(prefix.value.SBKMJCL)
//*
//*   APPL NAME = APPL
//*    JOB TYPE = JBT JOB COLLECTION
//*     SUBMIT BY PDUSER  ON 11/17/2009 14:37:35
//*
//BKMJBT EXEC BKMJBTCV,APPL=APPL,
// JBTJOBS=
// IF (RC < 5 ) THEN
//JOBCOLL EXEC BKMJOBSQ
// ENDIF
```

Parameters

APPL ***appl***
The name of the application.

JBTJOBS
The Jobtrac schedule output file specified in BKMINI.

Sample Unicenter CA-Jobtrac job collection results

This information shows an example of a batch SYSPRINT file from a successful Unicenter CA-Jobtrac job collection event. Based upon the input anchor jobs supplied, the output listing displays both the anchor structures and any non-succeeding job structures.

Wherever possible, the SYSPRINT output includes the current status, the SYSID values, and the list of all succeeding job structures (if any) for the anchor jobs. If the job was successful, then the last run date and time values are also included.

```
Entries Displayed By Collection Order:

                                          Sched        Run
-Job Name- -Status-SYSID---Bus Group-- -Time/Date- -Time/Date- --- Successors ---

 CA3050AP  WAIT    N/A       ALACA        08/11 21:45 N/A   N/A   CAXTRTAP CAXTR45
                                                                  CA4000AP CA4100AP
                                                                  CA1500AP CABDF56I
                                                                  MI4000AP MI2000AP
 CA4100AP  WAIT    N/A       ALACA        08/11 22:15 N/A   N/A   CA4151AP
 CA4600AP  WAIT    N/A       ALACA        08/11 22:00 N/A   N/A   CA4410AP CA3075AP
 CAXTRTAP  WAIT    N/A       ALACA        08/11 23:06 N/A   N/A   CA4100AP CA456M
 CA3070AP  WAIT    N/A       ALACA        08/11 22:25 N/A   N/A   CA3075AP CA6620AP
 CA3065AP  WAIT    N/A       ALACA        08/11 22:45 N/A   N/A   CA3066AP
 CA3100AP  WAIT    N/A       ALACA        08/11 22:30 N/A   N/A   ***NONE***
 CA3850AP  WAIT    N/A       ALACA        08/11 22:38 N/A   N/A   CA3853AP CA6620AP
 CA4000AP  WAIT    N/A       ALACA        08/11 22:20 N/A   N/A   CA4008AP
 CA4475AP  WAIT    N/A       ALACA        08/11 22:30 N/A   N/A   CAOC40AP CA3070AP
 CA6590AP  WAIT    N/A       ALACA        08/11 22:10 N/A   N/A   ***NONE***
 CA6620AP  WAIT    N/A       ALACA        08/11 22:15 N/A   N/A   ***NONE***
 MI2000AP  WAIT    N/A       ALACA1       08/11 23:00 N/A   N/A    MI2005AP MI45BB0
 SB1600AP  WAIT    N/A       ALACA1       08/11 22:07 N/A   N/A   CICA10AP
 CA1500AP  COMPLETE MVSC     ALACA        08/11 22:50 08/12 23:45 CAMRBDAP CA3066AP
 CA9701AP  WAIT    N/A       ALACA        08/11 22:47 N/A   N/A   ***NONE***
```

Use the **STATUS** option from the Application AutoBuild Options menu to view the job list display and any job structure entries collected. When job collection is complete, the Unicenter CA-Jobtrac job entries and job structures are retained in the Automated Critical Data Identification Application Data Inventory database.

Performing job collection with TWS

To use this job collection method, your installation must have TWS installed and the jobs that comprise the application must run under the control of TWS.

Before you perform job collection with TWS, verify these values have been set for the TWS keywords in the BKMINI product parameter library member's :ACDI section.

- **JOBCOLL_DROP_ALL_UNRESOLVED_PREDECESSORS**

Automated Critical Data Identification obtains information about the structure of an application from TWS by analyzing the report output of the job scheduling product. Automated Critical Data Identification uses the TWS batch reporting utility, EQQBATCH, to obtain the job names associated with a TWS application name.

1. From the **Application AutoBuild Options** panel, select **1 (JOB COLLECT)**.
   The **APPLICATION JOB COLLECTION OPTIONS** panel is displayed.
2. Select **5 TWS (OPC)**.
3. Press Enter.
   An ISPF Edit session is displayed.
4. Specify the job collection values:
   - TAPL — TWS application or mask.

   For example:

   ```
   TAPL APPL1* APPL2* APPL3
   ```

5. Press F3.

   The **TWS Job Collection** panel is displayed.
6. In the **SPECIFY TWS Valid Date** field, type the TWS date in the format: MM/DD/YYY (the default is the current date) and press Enter.
7. From the **JOB Statement Verification For Batch Submit** panel, review the job card specification and make any changes, if needed.
8. Press Enter to continue or press F3 to exit.
9. Select from these options to process the batch job:
   - To edit the JCL prior to submitting the job, type S in the **EDIT JCL Prior to Submit** field and press Enter.
   - To submit the batch job, type S in the **SUBMIT JOB For Execution** field and press Enter.
   - Press F3 to exit without submitting the job.

Sample TWS batch collection job

This information provides an example of the job collection procedure generated through the **TWS (OPC)** job collection option on the **APPLICATION JOB COLLECTION OPTIONS** panel.

Sample

```
//APPLJRFV JOB (ACCT),,'ABC
R',CLASS=A,MSGCLASS=X,NOTIFY=&SYSUID
//*
//*
//JCLL JCLLIB ORDER=(prefix.value.SBKMJCL)
//BKMTWS  EXEC BKMTWJOB,
// APPL=APPL,
// TWSDATE=11/17/2009
//*
//*   APPL NAME = APPL
//*    JOB TYPE = TWS JOB COLLECTION
//*     SUBMIT BY PDUSER  ON 11/17/2009 14:04:28
//*
// IF (RC < 5 ) THEN
//JOBCOLL EXEC BKMJOBSQ
// ENDIF
```

Parameters

APPL ***appl***
The name of the application.

TWSDATE ***date***
Specifies the TWS date in the format MM/DD/YYYY.

Sample TWS job collection results

This information provides an example of a batch JOBRPT output produced from a TWS job collection event.

```
       ORIGINAL JOBS          FOR APPLICATION: TWTEST1  2014/02/03 08:50:28

JOBNAME       PREDECESSOR
--------      -----------

SMFDAILY
  SMFD1000    SMFDAILY
  SMFD1100    SMFD1000
  SMFD1150    SMFD1100
  SMFD2000    SMFD1150
  SMFD2200    SMFD2000
  SMFD2400    SMFD2200
SMFDAILY HAS      6 SUCCESSOR JOBS

SMFREQ1
SMFREQ1  HAS NO SUCCESSOR JOBS

SMFREQ2
SMFREQ2  HAS NO SUCCESSOR JOBS
```

Performing job collection with ASG-Zeke

To use this job collection method, your installation must have ASG-Zeke installed and the jobs that comprise the application must run under the control of ASG-Zeke.

Before you perform job collection with ASG-Zeke, verify that values have been set for the ASG-Zeke keywords in the BKMINI product parameter library member's :ACDI section.

- **JOBCOLL_ZEKE_JCLLIB**
- **JOBCOLL_ZEKE_JCLLIB02**
- **JOBCOLL_ZEKE_JCLLIB03**
- **JOBCOLL_ZEKE_VERSION**
- **JOBCOLL_ZEKE_ZEKEUTL**
- **JOBCOLL_ZEKE_ZEKEUTL02**
- **JOBCOLL_ZEKE_ZEKEUTL03**

In the :JCL_DEFAULTS section, verify that values have been set for these keywords:

- **JOBSSPCE**
- **ZEKEOUTF**
- **ZEKEOUTP**
- **ZEKEJOBS**
- **ZEKESPCE**

1. From the **Application AutoBuild Options** panel, select the **JOB COLLECT** option.
   The **APPLICATION JOB COLLECTION OPTIONS** panel is displayed.
2. Select the **ZEKE MVS** option from the **JOB COLLECTION OPTIONS** menu.
   An ISPF Edit session is displayed to enable you to create the job collection criteria.
3. Specify the job collection criteria:
   a) For the **JOB** criteria, specify a job name or job name mask.
   b) For the **ENAME** criteria, specify an event name or event name mask.
   c) For the **ZAPL** criteria, specify the Zeke application or a mask.
   d) For the **GROUP** criteria, specify a group ID or a group ID mask.
   For example:

```
JOB      AC1*E* AC1*F* AC1*G*
JOB      AC2*
ENAME    DLYAC*
ZAPL     AP*
GROUP    GP1
```

4. Press F3.

   If it is applicable, a **Zeke Extraction File** panel indicates if ASG-Zeke interface data is available from a previous extraction and the date the data was created. To use the existing extraction file press Enter; otherwise, press F3 to re-extract ASG- Zeke jobs.

   Note: If updates have been made for the application since the last extraction, or if the ASG-Zeke interface has not been used for the current version of ASG-Zeke, press F3 to use the ASG-Zeke interface to extract ASG-Zeke jobs.
5. If multiple **JOBCOLL_ZEKE_JCLLIB** token values were specified in the BKMINI member, specify the JCL library to use.
6. If multiple **JOBCOLL_ZEKE_ZEKEUTL** token values were specified in the BKMINI member, specify the utility to use.
7. From the **JOB Statement Verification For Batch Submit** panel, review the job card specification and make any changes, if needed.
8. Press Enter to continue or press F3 to exit.
9. Select from these options to process the batch job:
   - To edit the JCL prior to submitting the job, type S in the **EDIT JCL Prior to Submit** field and press Enter.
   - To submit the batch job, type S in the **SUBMIT JOB For Execution** field and press Enter.
   - Press F3 to exit without submitting the job.

Sample ASG-Zeke batch collection job

This information provides an example of the job collection procedure generated through the **ZEKE MVS** job collection option on the **APPLICATION JOB COLLECTION OPTIONS** panel.

Sample

```
//APPLJRF4 JOB (ACCT),,'ABC
R',CLASS=A,MSGCLASS=X,NOTIFY=&SYSUID
//*
//*
//JCLL JCLLIB ORDER=(prefix.value.SBKMJCL)
//STEPA   EXEC PGM=IEFBR14
//DD02     DD  DSN=ZEKEUTL.AMPZEKE.FILE,
//    DISP=(MOD,DELETE),
//    UNIT=SYSALLDA,SPACE=(0,0)
//STEP1   EXEC ZEKEUTL
//SYSIN    DD  DISP=SHR,
//  DSN=prefix.value.SBKMPARM(ZEKEUTIL)
//AMPZEKE  DD  DISP=CATLG,
//    DSN=ZEKEUTL.AMPZEKE.FILE,
//    UNIT=SYSALLDA,
//    SPACE=(CYL,(10,10),RLSE)
//*
//*   APPL NAME = ABCRTST1
//*   JOB TYPE = ZEKE JOB COLLECTION
//*     SUBMIT BY PDUSER  ON 11/17/2009 13:22:16
//*
//STEP2   EXEC BKMZI450,ZPGMI=BKMZI510
// IF (RC < 5 ) THEN
//JOBZK   EXEC BKMJOBZK,
//  ZPGM=BKMZK450,APPL=ABCRTST1
// ENDIF
// IF (RC < 5 ) THEN
//JOBCOLL EXEC BKMJOBSQ
// ENDIF
```

Parameters

ZPGM
Specifies the program name to pass to PROC BKMJOBZK.

APPL ***appl***
The name of the application.

Sample ASG-Zeke job collection reports

This information provides examples of ASG-Zeke job collection reports. The ASG-Zeke job collection report indicates the structure of the jobs.

In this example, job ABJOB03 is succeeded, in order, by jobs ABJOB02, ABJOB04 and ABJOB01. The ABJOB13 and ABJOB23 jobs are independent from ABJOB03 and have succeeding jobs:

```
ORIGINAL JOBS FOR APPLICATION: ABJOBS   2011/10/26 20:17:34       PAGE:  1

 JOB NAME      ZEKE APPLICATION  ZEKE GROUPID  ZEKE EVENT NAME
 -------      ----------------  ------------  ---------------

 ABJOB03          ABJOBS1           AB1       AB1 ABJOB03
   ABJOB02        ABJOBS1           AB1       AB1 ABJOB02
   ABJOB04        ABJOBS1           AB1       AB1 ABJOB04
   ABJOB01        ABJOBS1           AB1       AB1 ABJOB01
 ABJOB03  HAS       3 SUCCESSOR JOBS

 ABJOB13          ABJOBS1           AB2       AB1 ABJOB13
   ABJOB12        ABJOBS1           AB2       AB1 ABJOB12
   ABJOB11        ABJOBS1           AB2       AB1 ABJOB11
 ABJOB13  HAS       2 SUCCESSOR JOBS

 ABJOB23          AB2               AB2       AB2 ABJOB23
   ABJOB22        AB2               AB2       AB2 ABJOB22
   ABJOB21        AB2               AB2       AB2 ABJOB21
 ABJOB23  HAS       2 SUCCESSOR JOBS
```

If job filters exist for the application, the JOBS AFTER FILTERING FOR APPLICATION report is also created. For example:

```
JOBS AFTER FILTERING FOR APPLICATION: ABJOBS   2011/10/26 20:17:34   PAGE:  1

 JOB NAME      ZEKE APPLICATION  ZEKE GROUPID  ZEKE EVENT NAME
 -------      ----------------  ------------  ---------------

 ABJOB03          ABJOBS1           AB1       AB1 ABJOB03
   ABJOB02        ABJOBS1           AB1       AB1 ABJOB02
   ABJOB01        ABJOBS1           AB1       AB1 ABJOB01
 ABJOB03  HAS        2 SUCCESSOR JOBS

 ABJOB23          AB2               AB2       AB2 ABJOB23
   ABJOB22        AB2               AB2       AB2 ABJOB22
   ABJOB21        AB2               AB2       AB2 ABJOB21
 ABJOB23  HAS        2 SUCCESSOR JOBS

 ABJOB12          ABJOBS1           AB2       AB1 ABJOB12
   ABJOB11        ABJOBS1           AB2       AB1 ABJOB11
  ABJOB12  HAS        1 SUCCESSOR JOBS
```

Using an input data set to perform job collection

You can use ISPF Edit to create a job entries list, or you can use a pre-staged data set or a user-supplied data set as input to a job collection.

- Choose one of these input data set methods based on your installation's needs:

Option	Description
ISPF Edit	Use ISPF Edit to create a job entries list as input to the job collection. Consider using this option if your installation does not use a job scheduling system or if there are only a few job names or job name masks to specify.

Option	Description
Pre-staged data set	Select this option if you want Automated Critical Data Identification to locate and validate job pre-staging data sets and submit a batch job for the job collection event. *Pre-staged* means that you place the job names into an external file. The pre-staged input file can then be used either as the only job collection input source, or it can be merged with job input from job scheduling products or input data set job entries.
User-supplied data set	Select this option if your installation does not use a job scheduling system or if there are a large number of jobs associated with the application.

Formatting job entries for input data sets

If you use ISPF Edit, a pre-staged data set, or a user supplied data set to perform job collection, you must format the job entries according to specific rules.

- Format job entries using these rules:
  - A line may contain a maximum of 80 characters.
  - Lines beginning with an asterisk (*) are ignored and can be used as comment statements.
  - Job names can be any valid job name, with a maximum length of eight characters.
  - A trailing asterisk (*) indicates a job name mask. All job names that match the character string that precedes the asterisk are selected.
  - Each job name or job name mask must be placed on a separate line.
  - It is not necessary that job names begin in the first column — however, they are blank delimited. (Automated Critical Data Identification considers the first non-blank field to be the job name and ignores any characters after the next blank.)

Defining job structure for input data sets

To assign unique job anchor structures for supplied job names, use the optional keyword :**STRUC**. Using this technique, you can indicate that unique jobs have a successor relationship to a previous job (the anchor job), even though they are not part of a job scheduling system.

When you use the :**STRUC** statement, Automated Critical Data Identification saves the anchor job entries according to your structure specification. If you do not specify the :**STRUC** statement, Automated Critical Data Identification places the entries in alphanumeric order. You can include multiple job structure statements to completely define each structure.

- If multiple successor job names are needed within :**STRUC** entries, you can continue the :**STRUC** statement entries using a plus sign (+). For example:

```
   JOB346TR
 :STRUC
   JOB346TR  JOBA1 JOBA2 JOBA3 JOBA4
   JOBWEEK2  JOB341 JOB342 JOB343 JOB344 +
             JOBWEEK3 JOBWEEK4 JOBEND
   JOB999T   JOB661 JOB662
```

  In this example, the anchor job: JOBWEEK2 contains seven succeeding job entries (JOB341, JOB342, JOB343, JOB344, JOBWEEK3, JOBWEEK4, and JOBEND).

Default job structure

If you create an application's jobs using ISPF Edit (no other job collection was performed), Automated Critical Data Identification creates a default structure when the edited data set is saved.

Example

For example, if you specify the jobs in this order:

1. JOB2

2. JOB4
3. JOB1

and you do not use the :**STRUC** keyword, Automated Critical Data Identification creates the structure:

```
JOB2  JOB4  JOB1
```

The first job becomes the predecessor for all the subsequent jobs.

Sample - 1 Job structure

In this sample, all of the job names supplied were assigned as one job anchor structure.

Automated Critical Data Identification automatically assigns the anchor starting point JOB346TR even though the subsequent jobs might not have a relationship to the job name JOB346TR. Automated Critical Data Identification assigns the starting point in this manner so SMF Scan can locate all of the job names.

```
JOB346TR  Anchor Structure JOB  - Normal Payroll Execution.
   JOBA1     Payroll Job-1 (Daily)
   JOBA2     Payroll Job-2 ( ... )
   JOBA3     Payroll Job-3 ( ... )
   JOB999T   Anchor Structure JOB- Special
   JOB341    Payroll Job-4 (Weekly)
   JOB342    Payroll Job-5 ( ...  )
   JOB343    Payroll Job-6 ( ...  )

   JOB346TR
 :STRUC
   JOB346TR  JOBA1 JOBA2 JOBA3
   JOB999T   JOB341 JOB342 JOB343
```

The first job name represents the anchor name to be assigned. The second and subsequent job names in the blank delimited string are the successor job entries.

Notice the job name (JOB346TR) included at the top of the example before the **:STRUC** statement. For the **:STRUC** statement to work, it is necessary to include a job name before the **:STRUC** statement.

The resulting application anchor job structure is:

```
     (1)         (2)                      (3)
   - JOB/Anchor - - Type - -- JOB Structure Elements -----------------

      JOB346TR      USR     JOBA1 JOBA2 JOBA3
      JOB999T       USR     JOB341 JOB342 JOB343
```

Sample - 2 Job structure

In this sample, the application has only a few jobs that are dependent on an anchor job. The remaining jobs have no structure.

```
JOBA    Quarterly Reconciliation Job A
   JOBB    Quarterly Reconciliation Job B
   JOBC    Quarterly Reconciliation Job C

   JOB1
 :STRUC
   JOB1  JOB2 JOB3
   JOBX
   JOBY
   JOBZ
```

If there is a job entry listed before the **:STRUC** statement that is not in the **:STRUC** list, Automated Critical Data Identification creates a job anchor structure record for it. If there is a job in the **:STRUC** list that is not in the job entries before the **:STRUC**, Automated Critical Data Identification builds a job entry record for it.

The resulting application anchor job structure is shown below.

```
     (1)         (2)                      (3)
   - JOB/Anchor - - Type - -- JOB Structure Elements -----------------
```

```
JOBA          USR       JOBA
JOBB          USR       JOBB
JOBC          USR       JOBC
JOB1          USR       JOB2 JOB3
JOBX          USR       JOBX
JOBY          USR       JOBY
JOBZ          USR       JOBZ
```

Using ISPF Edit to perform job collection

Select this option if you want to use ISPF Edit to specify the job names and job structure associated with an application.

When you use ISPF Edit to perform job collection, you must know the job name conventions.

Tip: Consider using a job name mask that matches the job naming convention. When you use a job name mask, any new jobs that are added to the schedule are captured automatically and you do not need to run additional job collections to capture the new jobs.

Note: You can also use ISPF Edit to add, change, or delete entries previously collected using a job scheduling product.

1. From the **Application AutoBuild Options** menu, select the **JOB COLLECT** option and press Enter.

   The **APPLICATION JOB COLLECTION OPTIONS** panel is displayed.
2. Select the **ISPF EDIT MODE** option.

   An ISPF Edit session is displayed.
3. Create and edit job entries list and job structure using ISPF Edit. Do not remove the comments. Type each job name in column 1.
   For example:

   ```
   JOB346TR  Anchor Structure JOB  - Normal Payroll Execution.
   JOBA1     Payroll Job-1 (Daily)
   JOBA2     Payroll Job-2 ( ... )
   JOBA3     Payroll Job-3 ( ... )
   JOB999T   Anchor Structure JOB- Special Payroll Execution.
   JOB341    Payroll Job-4 (Weekly)
   JOB342    Payroll Job-5 ( ...  )
   JOB343    Payroll Job-6 ( ...  )
   ```

   In the example, eight job names were defined. Each job name has an optional comment description that is saved with the job entries.
4. Press F3 to save your changes.

   The **JOB Statement Verification For Batch Submit** panel is displayed.
5. From the job statement verification panel, review your entries and make any changes to the JCL, if needed.
6. Press Enter to submit the job.

   The job submission panel is displayed.
7. Select one of these options to process the batch job:
   - To submit the batch job, type S in the **SUBMIT JOB For Execution** field and press Enter.
   - To edit the JCL prior to submitting the job, type S in the **EDIT JCL Prior to Submit** field and press Enter.
   - Press F3 to exit without submitting the job.
8. Refresh the RSP job table by submitting the BKMJRFSH job from *prefix.value*.SBKMJCL (where *prefix.value* represents one or more qualifiers of the installed product libraries). Performing this step enables RSP to update the job table and begin collecting JCL and SMF data for the application.
9. Run the production application.
10. Signal the end of the production application cycle by editing a copy of BKMAPLEN. Specify the name of the application in the **APPL** parameter. For example:

```
//BKMAPLEN EXEC BKMAPLEP,APPL=PAYROLL
```

The BKMAPLEN job analyzes the RSP JCL and SMF records and builds the Selection Data Set that is input to a backup. The list of data sets is written in the format specified by the Setup options.

Using a pre-staged data set to perform job collection

A pre-staged data set enables you to supply an alternate input file to the job collection function outside of the normal Automated Critical Data Identification job scheduling product support. Automated Critical Data Identification uses the contents of the pre-staged data set as input to the job collection.

Before invoking the Automated Critical Data Identification job pre-staging facility, you must define a pre-staged data set to Automated Critical Data Identification. This input file can be either a PDS member or a sequential file.

When you create a pre-staged data set, you create input to the job collection by placing job names into an external file. A pre-staged input file can be used either as the only job collection input source, or it can be merged with job input from either job scheduling products or input data set job entries.

1. Specify the configuration values for a pre-staged data set job collection in the BKMINI product parameter library member.

   BKMINI contains configuration and global options used by Automated Critical Data Identification during the job collection event and other interfaces to the job pre-staged job scheduling product. The unique pre-staged input file for each application must be specified in BKMINI so Automated Critical Data Identification can locate and allocate the data set.

   To change any job pre-staged related option, use ISPF Edit to edit the BKMINI member in the product parameter library data set. Any changes made will take effect during the next job pre-staged job collection request.
2. Create the pre-staged data set and format the job entries.
3. Assign the job structure for the input data set entries using the **:STRUC** statement.
4. From the **Application AutoBuild Options** menu, select **JOB COLLECT** and press Enter.

   The **APPLICATION JOB COLLECTION OPTIONS** panel is displayed.
5. Select the **JOB PRE-STAGED DATASET** option.

   The **USER CREATED DATA SET** panel is displayed.
6. In the **PRE-STAGED OR USER DATA SET** field, specify the pre-staged data set.
7. Press Enter.

   The **JOB Statement Verification For Batch Submit** panel is displayed.
8. From the job statement verification panel, review your entries and make any necessary changes to the JCL.
9. Press Enter to submit the job and display the job submission panel.
10. Select one of these options to process the batch job:
    - To submit the batch job, type S in the **SUBMIT JOB For Execution** field and press Enter.
    - To edit the JCL prior to submitting the job, type S in the **EDIT JCL Prior to Submit** field and press Enter.
    - Press F3 to exit without submitting the job.
11. Refresh the RSP job table by submitting job BKMJRFSH from *prefix.value*.SBKMJCL. This enables RSP to update the job table and begin collecting JCL and SMF data for the application.
12. Run the production application.
13. Signal the end of the production application cycle by editing a copy of BKMAPLEN. Specify the name of the application in the **APPL** parameter. For example:

    ```
    //BKMAPLEN EXEC BKMAPLEP,APPL=PAYROLL
    ```

    The BKMAPLEN job analyzes the RSP JCL and SMF records and builds the Selection Data Set that is input to a backup. The list of data sets is written in the format specified by the Setup options.

Using a user-supplied data set to perform job collection

To collect job entries and job structure by providing a user-supplied data set, complete these steps.

1. Create the user-supplied data set and format the job entries.
2. Assign the job structure for the input data set entries using the **:STRUC** statement.
3. From the **Application AutoBuild Options** menu, select the **JOB COLLECT** option.

   The **APPLICATION JOB COLLECTION OPTIONS** panel is displayed.
4. Select the **USER SUPPLIED DATASET** option. Type the name of the user-supplied data set in the **PRE-STAGED OR USER DATA SET** field.
5. Press F3 to display the **JOB Statement Verification For Batch Submit** panel.
6. From the job statement verification panel, review your entries and make any necessary changes to the JCL.
7. Press Enter to submit the job and display the job submission panel.
8. Select one of these options to process the batch job:
   - To submit the batch job, type S in the **SUBMIT JOB For Execution** field and press Enter.
   - To edit the JCL prior to submitting the job, type S in the **EDIT JCL Prior to Submit** field and press Enter.
   - Press F3 to exit without submitting the job.
9. Refresh the RSP job table by submitting job BKMJRFSH from *prefix.value*.SBKMJCL. This enables RSP to update the job table and begin collecting JCL and SMF data for the application.
10. Run the production application.
11. Signal the end of the production application cycle by editing a copy of BKMAPLEN. Specify the name of the application in the **APPL** parameter. For example:

    ```
    //BKMAPLEN EXEC BKMAPLEP,APPL=PAYROLL
    ```

    The BKMAPLEN job analyzes the RSP JCL and SMF records and builds the Selection Data Set that is input to a backup. The list of data sets is written in the format specified by the Setup options.

Modifying a job scheduling product job list or structure

If you collect an application's jobs using a job scheduling product and then subsequently choose ISPF Edit for job collection for that application, Automated Critical Data Identification creates an edit data set using the jobs previously collected and builds the **:STRUC** statement. You can then add, change, or delete the jobs and job structures as needed.

Creating a backup job list

Complete these steps to create or edit a backup job list to identify application-specific backup job names. The backup job list is used in conjunction with application setup Backup End option, **Application Only Backup**.

1. From the **Application AutoBuild Options** panel, select the **BACKUP JOB LIST** option and press Enter to open an ISPF Edit session.
2. Type one job name or job name mask per line, starting in any column before 65. Use as many lines as needed.

Creating a cycle control job list

Complete this task to create or edit a cycle control job list that identifies application-specific cycle start and cycle end job names.

The cycle control job list is used in conjunction with application setup CYCLE CONTROL option **J** (JOB). If you specify both cycle start and cycle end jobs, the RSP started task tracks application jobs between the

cycle start and end jobs only. The RSP started task does not track application jobs executing outside the cycle start and end window.

When you use the Cycle Control function, an APPLEND job and a WTO message are submitted automatically when a cycle has completed. The WTO message contains the application name, job name, cycle number, date, and time. Cycle Start or Cycle End jobs (or both) can be registered to specific applications so that application-specific action may be taken when cycles complete. The RSP started task detects cycle completion (using an SMF type 30 record), submits an APPLEND job, and uses the WTO command to produce message BKMK1050I on the system log. You can use a network management product to monitor console messages and take the appropriate actions whenever the WTO messages are detected.

Note: For application-scheduled APPLEND jobs, you can control the execution of next cycle application jobs so that they can be delayed until after the APPLEND job completes. This is not necessarily the case for the RSP-submitted APPLEND job because it is not part of the application schedule. In this situation, next cycle application jobs could be run before the RSP-submitted APPLEND job completes so that the critical data set list and BKUPEND might not accurately reflect the true cycle end point. When a cycle end job completes, the cycle end WTO is issued and the APPLEND job is submitted.

1. From the **Application AutoBuild Options** panel, select the **CYCLE CONTROL** option to display a panel where you can specify cycle control job names.
2. On the cycle control job name specification panel, you must specify at least one cycle start or cycle end job name.

   You may specify one cycle start job name, or up to eight cycle end job names, or both one cycle start job name and as many as eight cycle end job names.

Viewing job collection results

Review the results of job collection events using one of these methods.

- From the **Application AutoBuild Options** menu, select the **STATUS** option to view the job elements and job anchor structures. From the **APPLICATION Display Selections** menu, you can choose to display the current job list or current job structures:

Option	Description
Display Current JOB List	Display the current job collection list entries.
Display Current JOB Anchor	Display the current job structures, or anchor chains.

- From the Automated Critical Data Identification Main Menu, select the **LIST APPLICATIONS** option.

Chapter 11. Identifying application data sets using RSP

Use RSP to identify JCL and SMF data to determine which data sets are used by an application and which of these data sets are required for backup and recovery.

The Real-time Selection Process (RSP) dynamically collects and analyzes SMF and JCL data for those applications that are defined to Automated Critical Data Identification. RSP collects data as the applications run and continuously updates the Automated Critical Data Identification Application Data Inventory database to keep it current with the previously collected application critical data set information.

1. Define the Selection Data Set name and type and then create the Selection Data Set.

   Note: This step is required for the initial build only.
2. Do one of the following:
   - If the RSP started task has never been issued, run BKMBLDJB to create the initial job flat file and then issue the start command BKMSTSKR to start RSP.
   - If the RSP started task has already been issued, run BKMJRFSH to rebuild the job flat file and refresh the job table.
3. Run the application.

   When you run the application, you enable Automated Critical Data Identification to record the data set information in the Automated Critical Data Identification Application Data Inventory database.
4. Run BKMAPLEN (step or job).

   Running BKMAPLEN updates and rebuilds the application's Selection Data Set entries. This allows the applications defined to Automated Critical Data Identification to be backed up at any time subsequent to processing completion. If RSP detects any changes for an application, it automatically reconstructs the Selection Data Set using the most current critical data set entries encountered through the end of BKMAPLEN processing.

RSP components and processing

Use the RSP components RSP SMF Capture and RSP JCL Capture to collect SMF and JCL data.

RSP SMF Capture

RSP SMF Capture (which uses the IEFU83 and IEFU84 SMF exit facilities) collects SMF record types 14, 15, 30, 61, 64, 65, and 66. These records supply data to RSP for application job cycle control job completion and data sets that are opened and closed, deleted, renamed, and allocated. RSP SMF Capture detects cycle control job completions and captures the first reference of an application data set. It does not matter if the reference is OPEN, or CLOSE, or URD (updated, renamed, or deleted).

Note: RSP SMF Capture does not collect SMF data for data sets that do not have references before BKMAPLEN is run (for example, CICS®).

In this situation, if CICS is started before RSP is started and the data sets are not closed before BKMAPLEN is run, RSP does not know they existed.

RSP JCL Capture

RSP JCL Capture (which uses the IEFUJI JCL exit facility) identifies all data sets in the JCL, even those data sets referenced in steps that are not processed or those data sets that are not opened— including concatenated libraries.

The database update task (ACDIUPDT)

A database update task (ACDIUPDT) runs on each image where you want to collect data.

Each job submitted to the system is scanned by RSP JCL Capture to identify each data set in the JCL. As jobs run, system SMF records are created for each data set that record whether or not a data set has been read, updated, renamed, or deleted. RSP intercepts each SMF record before the record is written to the system SMF MANX and MANY data sets. These JCL and SMF records are written to the Automated Critical Data Identification Application Data Inventory database as "raw" records, or *data set detail records* (DSD), and can be viewed through the Automated Critical Data Identification ISPF interface.

The records intercepted by RSP SMF Capture and RSP JCL Capture are stored temporarily in buffers that are created in a data space. The database update task unloads the data space buffers and updates the Automated Critical Data Identification Application Data Inventory database. Before it unloads the data space buffers, the database update task waits until the number of loaded buffers reaches a threshold that maximizes I/O efficiency.

If the database update task slows and fills most of the buffers in the data space, due to the volume of data being processed, the task enlarges the data space. The data space can be enlarged to twice its initial size. If more room is required, the export task unloads the data space to an overflow file. When buffers are available, the import task reloads the data space buffers. The export task also unloads the data space if the database update task is not running and the buffers are full. The unloaded records are not reloaded until the database update task is processing normally again.

System configuration and RSP tasks

RSP can collect data for applications defined to Automated Critical Data Identification in single image or multi-image (SYSPLEX) environments. This capability allows you to schedule and run applications, or unique application jobs, on any processor image and still maintain continuous collection of application file usage.

Creating the job table

Before an application can be tracked by RSP or SMF Scan, you must create the job table. All data collection methods read the job table.

Automated Critical Data Identification creates job reference tables from a sequential file, called the *job flat file*, which is extracted from the Automated Critical Data Identification Application Data Inventory database. You must run BKMBLDJB to create the initial job table prior to starting RSP for the first time.

- Run BKMBLDJB, a member of the Automated Critical Data Identification product JCL library, to perform the extraction and create the initial job file.

To populate the job table with the new jobs to track, run BKMJRFSH, a member of the Automated Critical Data Identification JCL library, each time you perform a job collection.

Updating the job table

Update the job table after you create the initial job table using BKMBLDJB, each time you perform a job collection, and when any application job list changes.

Automated Critical Data Identification does not automatically update the job table therefore, you must run BKMJRFSH each time you perform a job collection so Automated Critical Data Identification knows which jobs to track.

- When any application job list changes, run BKMJRFSH to rebuild the job table and notify RSP on each image that a new job table should be loaded. You can run BKMJRFSH on any image of a complex — all associated images are updated automatically.

  Tip: You might want to schedule BKMJRFSH to run at a specific time interval (for example, once a day) to ensure that application job list changes are reflected in the tables of each image running RSP.

Running the application

When you run the application, you enable Automated Critical Data Identification to record the data set information in the Automated Critical Data Identification Application Data Inventory database.

Using RSP, each job submitted to the system is scanned by RSP JCL Capture to identify each data set in the JCL, even the procedure libraries. As jobs run, system SMF records are created for each data set that records whether a data set has been read, updated, renamed, or deleted. RSP intercepts each SMF record before the record is written to the system SMF MANX and MANY data sets.

JCL and SMF records are written to the Automated Critical Data Identification Application Data Inventory database as data set detail records (DSD).

Running BKMAPLEN

BKMAPLEN builds a Selection Data Set if you are tracking the application using realtime JCL and SMF entries collected by RSP.

Note: You cannot use BKMAPLEN to build a Selection Data Set from batch SMF data. You must use Verification.

1. Modify the selection criteria.

   Automated Critical Data Identification saves any changes you make in the Automated Critical Data Identification Application Data Inventory database for reference and reporting purposes.
2. Run BKMAPLEN at the end of the application cycle.

   Automated Critical Data Identification automatically builds the Selection Data Set.

   The Selection Data Set is then immediately available to the backup.

How BKMAPLEN builds the Selection Data Set

Automated Critical Data Identification builds Selection Data Set entries from the application entries in the Automated Critical Data Identification Application Data Inventory database.

Automated Critical Data Identification Application Data Inventory database entries are generated by the following:

- SMF Scan
- RSP SMF Capture
- RSP JCL Capture
- External type filters: INCL_EXTDSN, ALLOC_EXTDSN, and ACCOMP_EXTDSN. (You can use these filters to specify data set names in the Automated Critical Data Identification Application Data Inventory database that the JCL or SMF data collection might not have captured.)
- Custom programs

The sources of data set names are employed depending upon on the application. RSP JCL Capture is not necessary if you want to use only data set names captured by RSP SMF Capture. An application could also be comprised of only data set names specified with external filters.

Building the Selection Data Set is a two step process:

1. The data set entries from the Automated Critical Data Identification Application Data Inventory database are evaluated for criticality using evaluation filters.
2. The Selection Data Set is built in the chosen format.

BKMAPLEN performs these actions when building the Selection Data Set:

1. BKMAPLEN combines the SMF and JCL records by date and time stamp (required if the application has run on more than one system).
2. BKMAPLEN categorizes the data sets as critical or noncritical based on each application's settings specified through the **0 (SETUP)** option.

3. BKMAPLEN converts the data set detail records to data set records (DSR analyzed records) and writes the Selection Data Set.

Automated Critical Data Identification uses the Current Verification settings, specified for the application through the **0 (SETUP)** option, when reconstructing the application's Selection Data Set entries.

Note:

- ABARS rejects a Selection Data Set that does not have an INCLUDE, ALLOCATE, or ACCOMPANY statement. In this case, the Selection Data Set build program produces the error message: `Selection Data Set Built in Error`. This error can be caused by all data sets being excluded by filters, no application jobs tracked, and so forth. Review the comments at the end of the Selection Data Set. (The INCL_EXTDSN filter can be used to force an INCLUDE for a dummy data set.)
- If BKMAPLEN produces a nonzero return code, the associated errors are written to the console log and to the SYSPRINT DD.
- If a specific SMF record was not captured by RSP, the reason could be that the data set was opened, but not closed, before BKMAPLEN was encountered. SMF record types 14 and 15 are generated when a data set is closed.

Modifying the BKMAPLEN JCL

Insert the BKMAPLEN JCL as the last job or as a step within the last job.

The necessity of using BKMAPLEN depends upon the structure of the job stream and the application backup plan:

- If the exit point of a job stream varies due to runtime parameter options or conditional execution based on program return codes, you can use BKMAPLEN to explicitly notify RSP where the application ends.
- If a backup step will be included immediately after the application completes, RSP requires that you place the BKMAPLEN step prior to the backup step.

1. Change the job statement to local standards.
2. Specify the `JCLLIB`.
3. Specify the name of the application on the EXEC statement:

   ```
   //BKMAPLEN EXEC BKMAPLEP,PARM='APPL(APPL0001),CYCLEINC'
   ```

   where APPLNAME specifies the name of the Automated Critical Data Identification application.

Examples

In this example, the application APPL0001 cycle number is incremented by one and the Selection Data Set is rebuilt:

```
//BKMAPLEN EXEC BRMAPLEP,APPL=APPL0001,CYCLEINC
```

In this example, the application PRODPAY cycle is incremented by one and the Selection Data Set is *not* rebuilt:

```
//BKMAPLEN EXEC BKMAPLEP,PARM='APPL(PRODPAY),NOSELREBUILD'
```

BKMAPLEN JCL parameters

These BKMAPLEN parameters specify how to rebuild an application's Selection Data Set entries, control the cycle number, and how to manage DSD records.

APPL= (*applname*), [SELREBUILD | NOSELREBUILD, CYCLEINC | CYCLENOINC, DELETEDETAIL | NODELETEDETAIL]

Parameters

APPL (*applname*)
(Required) The name of the application.

SELREBUILD|NOSELREBUILD
SELREBUILD and **NOSELREBUILD** are optional parameter keywords that you can use to rebuild an application's Selection Data Set entries.

SELREBUILD — (Default) Rebuild the application's Selection Data Set after the Automated Critical Data Identification Application Data Inventory database update process has been completed.

NOSELREBUILD — Bypass rebuilding of the application's Selection Data Set entries.

CYCLEINC|CYCLENOINC
CYCLEINC and **CYCLENOINC** are optional parameter keywords that are used to increase the cycle number.

CYCLEINC — (Default) Increase the cycle number by one. This option is appropriate for most applications.

CYCLENOINC — Keep the cycle number the same.

DELETEDETAIL|NODELETEDETAIL
DELETEDETAIL — Cause the DSD records for the application to be deleted after the Selection Data Set has been successfully created and BKMAPLEN has ended with a return code zero. If the BKMAPLEN return code is greater than zero, the DSD records will not be deleted. If the DSD records are removed, a full cycle of data will not be available.

NODELETEDETAIL — (Default) Do not delete the DSD after the Selection Data Set is created.

Using VDR filter lists during BKMAPLEN processing

Automated Critical Data Identification allows you to use the OpenTech Virtual Data Recovery (VDR) filter list to exclude selected tape data sets. The exclusion list is processed after the normal Automated Critical Data Identification filters.

- Add a DD statement to the BKMAPLEN job or PROC. For example:

  ```
  //VDRPATT DD DISP=SHR,DSN=VDR filter DSN
  ```

  Note: There is no Automated Critical Data Identification Setup option for this optional filter. You must use a DD statement to invoke it. Additionally, you cannot invoke VDR filters in Automated Critical Data Identification Foreground or Background Verify.

Using RSP Monitor

RSP Monitor provides information about RSP processing and allows you to issue selected RSP operator commands.

- To access RSP Monitor, select the **RSP MONITOR** option from the Automated Critical Data Identification Main Menu.

  The **RSP MONITOR** panel is displayed.

Displaying RSP status

Displaying the status of RSP tasks can provide status information to help you to determine why a task is down.

- Select one of these options:
  - Select the **Display RSP Status** option to display the status of all RSP tasks on the current image.
  - If a task is down, use the line command S to view additional status information.

Displaying RSP statistics

Complete these steps to display statistics for a specific system ID and date.

1. From the **RSP MONITOR** panel, select the **Display RSP Statistics** option.

   The **SELECT SYSID AND DATE** panel is displayed.
2. In the **ENTER SYSID DEFAULT IS CURRENT SYSID** field, type the system ID (maximum of four characters).

   The default is the current system ID.
3. In the **CHANGE DATE TO VIEW MONITOR STATISTICS FROM PREVIOUS DAYS DEFAULT IS TODAY'S DATE :** field, specify the date in the format: MM/DD/YY.

   The default is the current date.
4. Press Enter.

   The **RSP PROCESSING STATISTICS** panel displays statistics for the specified system ID and date.

   Tip: To view additional information about the data, move the cursor to a red or yellow data field and press F1.

RSP processing statistics

The **RSP PROCESSING STATISTICS** panel provides statistics related to RSP processing.

Note: Press Enter to refresh the current query.

U83 EXIT/UJI EXIT RECORD PROCESSING

SMF/UJI RECORDS MOVED TO RSP DATA SPACE
The number of exit records that matched the selection criteria in the U83 exit program (BKMRU83T) or the UJI exit program (BKMUJISE).

These records are reformatted and stored in the RSP data space. The database task (BKMRUPDT) removes them from the data space and updates the Automated Critical Data Identification Application Data Inventory database.

SMF/UJI RECORDS DROPPED - NO MATCH IN JOB TABLE
The number of records that were not moved to the RSP data space because the job name associated with these records did not match a job name in the job table.

SMF/UJI RECORDS CURRENTLY IN BUFFER OVERFLOW DATASET
The number of records that were moved to the overflow data set because there were no available buffers in the RSP data space. When space becomes available, these records are moved back to the RSP data space.

SMF/UJI RECORDS LOST - OVERFLOW DATASET NOT AVAILABLE
The number of records that were lost because the overflow data set was not available or unusable.

NUMBER OF TIMES LOSSES OCCURRED
The number of times that all of the buffers in the RSP data space were filled and new records could not be moved to the overflow data set.

ARE LOSSES OCCURRING NOW?
The current buffer overrun status:

- Y (Yes) indicates that record losses are currently occurring.
- N (No) indicates no current record loss. (If the **NUMBER OF TIMES LOSSES OCCURRED** field show losses, previous losses have occurred.)

Note: If you see losses, increase the buffers only on the system where the losses are occurring. Make sure the overflow file is large enough to accommodate the transfers from the data space to the overflow file.

RSP DATA SPACE BUFFERS

TOTAL BUFFERS
The total number of buffers created when RSP was started.

AVAILABLE BUFFERS
The number of buffers not in use.

HIGHEST NUMBER OF BUFFERS USED
The highest number of buffers in use at one time.

LONGEST TIME OF CONTINUOUS BUFFER USE IN MINUTES
The longest amount of time, in minutes, that the database task (BKMRUPDT) was processing data from the buffers in the RSP data space, without clearing all of the buffers before the U83 exit task or UJI exit task added a new record.

NUMBER OF TIMES 80 TO 99 PERCENT OF BUFFERS IN USE
NUMBER OF TIMES 50 TO 79 PERCENT OF BUFFERS IN USE
NUMBER OF TIMES 20 TO 49 PERCENT OF BUFFERS IN USE
NUMBER OF TIMES 00 TO 19 PERCENT OF BUFFERS IN USE
These fields count the number of the times the buffer use percentage was within the stated ranges. When the statistics record is updated (approximately every 30 seconds), the database task computes the percentage and adds 1 to the appropriate count.

RSP DATABASE PROCESSING

NUMBER DSD RECORDS WITH NO HITS IN FILTER TABLE
The number of DSD records that did not receive any hits against the filter table.

NUMBER DSD RECORDS WITH FILTER TABLE HITS
The number of DSD records that received hits against the filter table.

NUMBER OF HITS ON KEEP FILTER
The number of DSD records that received hits on a Keep filter.

NUMBER OF HITS ON DISCARD FILTER
The number of DSD records that received hits on a Discard filter.

MAXIMUM NUMBER OF ENTRIES IN DATABASE CACHE
The number of entries the database cache can hold.

NUMBER OF CACHE ENTRIES REUSED WHEN FULL
When the cache is full, the oldest entry is replaced by the new entry. This field shows the number of times the oldest entry was replaced by a new entry.

NUMBER OF DSD RECORDS THAT MATCHED CACHE ENTRIES
The number of DSD records from the RSP data space that matched a cache entry.

NUMBER OF CACHE MATCHES WITH DATABASE I/O
The number of DSD records that matched cache entries and Automated Critical Data Identification Application Data Inventory database I/O was required.

NUMBER OF CACHE MATCHES THAT AVOIDED DATABASE I/O
The number of DSD records that matched cache entries and Automated Critical Data Identification Application Data Inventory database I/O was not required.

NUMBER OF ACDI DATABASE READS
The number of DSD records read for update. If a record does not exist, it is added. See the **NUMBER OF ACDI DATABASE ADDS AND UPDATES** field for updates.

NUMBER OF ACDI DATABASE ADDS AND UPDATES
The number of DSD records updated or added to the Automated Critical Data Identification Application Data Inventory database. If the record does not exist, it is added. An existing record is only updated if the time stamp for new data is earlier than the existing time stamp.

RSP DATABASE TASK TUNING

BUFFER QUEUE THRESHOLD
The database update task does not process buffers until there are at least this number waiting, unless BKMAPLEN is in progress or a user has requested a flush. The database task optimizes

its processing so that it maximizes efficiency when updating the Automated Critical Data Identification Application Data Inventory database.

AVERAGE NUMBER OF BUFFERS PROCESSED
The average number of buffers processed for each database access request when the threshold is set at the current value.

AVERAGE NUMBER OF RECORDS PROCESSED
The average number of DSD records created in processing the average number of buffers.

AVERAGE NUMBER OF DATABASE I/O'S
The average number of database I/Os required to process the average number of records.

AVERAGE TIME IN SECONDS TO PROCESS BUFFERS
Each time the database update task requests database access it will have 30 seconds to perform I/O. The task attempts to accumulate the number of buffers it can process in 30 seconds before requesting database I/O.

Displaying the current job table

This information describes how to display the job table members that RSP is currently using.

- From the **RSP MONITOR** panel, select the **Display Current Job Table** option.
  The members of the job table that RSP is currently using are displayed by application.

Displaying the current filter table

This information describes how to display the members of the retention filter table that RSP is currently using.

- To display the current filter table, select the **Display Current Filter Table** option.
  The members of the retention filter table that RSP is currently using is displayed.

Current RSP Retention Filter Table fields

The fields on the **CURRENT RSP RETENTION FILTER TABLE** panel provide information about Retention filters.

Fields

APPLNAME
The name of the application.

FILTER ACTION
The filter action. For example: KEEP, DISCARD, or GLOBAL.

FILTER TYPE
The type of filter. For example: DSN, JOB, SYSID, UNIT, or VOLUME.

FILTER SCOPE
The scope of the filter. For example: UNIVERSAL, GLOBAL, or LOCAL.

FILTER
The filter.

RSP commands

Selecting the **RSP Commands** option lists the operator commands to stop or start the RSP tasks. You must have SAF authority to issue these commands. To select a command, type the number of a listed RSP command next to the **CMD** prompt and press Enter.

The available command options on the **RSP COMMANDS** panel are:

FLUSH BUFFERS
The RSP data space buffers are flushed immediately.

STOP RSP ACDIDB UPDATE TASK
F RSP,ACDIUPDT(STOP)

START RSP ACDIDB UPDATE TASK
F RSP,ACDIUPDT(START)

KILL RSP ACDIDB UPDATE TASK
F RSP,ACDIUPDT(KILL)

STOP RSP DATASPACE OVERFLOW EXPORT TASK
F RSP,OVFEXTSK(STOP)

START RSP DATASPACE OVERFLOW EXPORT TASK
F RSP,OVFEXTSK(START)

STOP RSP DATASPACE OVERFLOW IMPORT TASK
F RSP,OVFIMTSK(STOP)

START RSP DATASPACE OVERFLOW IMPORT TASK
F RSP,OVFIMTSK(START)

SUBMIT JOB TO UPDATE RSP JOB TABLE
Submits job BKMJRFSH directly from the panel.

Viewing the RSP message log

RSP maintains and continuously updates a message log file with event and diagnostic information for each application on every processor image. The message log file is a required file that is allocated using supplied JCL members.

- To view the RSP message log, use ISPF Browse. The first header record contains a LAST and NEXT block pointer (or line number) value you can use to locate the last series of recorded RSP messages.

Viewing captured SMF and JCL records

Complete these steps to view the SMF and JCL records captured by RSP.

1. From the Automated Critical Data Identification Main Menu, select the **SELECT/CREATE APPL** option.
2. Select the application.
   The **Application AutoBuild Options** menu is displayed.
3. Select the **STATUS** option.
4. Select the **Display Dataset Detail Records** option.
   The **DATASET DETAIL RECORDS** panel is displayed.

   Note: If no records are displayed, it is because they reside in the data space buffers or overflow file. You can flush the buffer using the **RSP Monitor RSP Commands** option.

RSP operational considerations

This information describes RSP operational considerations.

Reducing the amount of data to be processed by RSP

Reducing the amount of data processed by RSP can help reduce the number of records that BKMAPLEN must process.

- Use Discard Retention filters instead of Exclude Evaluation filters. Using Discard Retention filters eliminates records at RSP collection time, which results in fewer records for BKMAPLEN to process.
- Be careful when using data sets that were collected using RSP JCL Capture. Data sets that are not used during processing might be included in the BKMAPLEN process. You can determine whether this is the case is to display the detail records.

1. From the Automated Critical Data Identification Main Menu, select the **SELECT/CREATE APPL** option.

2. Specify an application.
3. From the **Application AutoBuild Options** menu, select the **STATUS** option.
4. Select the **Display Dataset Detail Records** option.

   Note: If a data set is captured by both the RSP SMF Capture and RSP JCL Capture, the two entries are displayed together.

RSP cycle considerations

Follow these guidelines when working with RSP cycles.

BKMAPLEN causes the RSP cycle to be incremented unless the parameters passed to BKMAPLEN are not valid, or the **CYCLENOINC** parameter is passed. If errors occur while the Selection Data Set is being built, the cycle will have been incremented.

Important: If you are rebuilding the Selection Data Set, do not rerun BKMAPLEN unless the you specify the **CYCLENOINC** parameter. Alternatively, you can build the Selection Data Set using Verification. If you rerun BKMAPLEN without specifying the **CYCLENOINC** parameter, the cycle will increment and the current cycle will contain nothing or you will obtain data set records from the current run of the application.

It is helpful to set up the JCL ahead of time to use if BKMAPLEN fails. If the normal BKMAPLEN JCL is a job containing BKMAPLEN followed by the backup, you could set up a rerun PROC that is a batch verification followed by the backup. Because a failing BKMAPLEN frequently takes some work to resolve, consider rebuilding the Selection Data Set and then starting the backup.

Note: Do not rerun the backup job if it also contains the BKMAPLEN step.

Another approach is to use the BKMBKEND process to control execution of the BKMAPLEN process. This approach allows you to rerun the job without changes. To use this method, perform these steps:

1. Select the **Setup** option from the Automated Critical Data Identification Main Menu to specify that you want BKMBKEND executed after every execution of the BKMAPLEN program.
2. Add a conditional BKMBKEND step to the job, following the backup step. The job then contains three steps:
   - BKMAPLEN
   - BACKUPS
   - BKMBKEND

When the job executes and the BKMAPLEN step is successful but the BACKUPS step fails, the BKMBKEND step is not performed. After correcting the issues with the BACKUP step, you can rerun the job. In the job's subsequent execution, the BKMAPLEN step does not process because the BKMBKEND was not run after the previous BKMAPLEN. The BACKUPS step runs using the results of the previous BKMAPLEN followed by the BKMBKEND. The next execution of the job executes all three steps. The following figure shows a sample job:

```
//APPLRR   JOB (ACCT),'APPL END',CLASS=A,MSGCLASS=X,
//              NOTIFY=&SYSUID,REGION=8M
//PROCLIB  JCLLIB ORDER=prefix.value.SBKMJCL
//*
//* APPLEND, BACKUP AND BKUPEND
//*
//BKMAPLEN EXEC BKMAPLEP,APPL=APPLRR
//BACKUP   EXEC PGM=ADRDSSU,REGION=8M
//SYSPRINT DD   SYSOUT=*
//TAPE     DD   DSN=APPLRR.BKUP(+1)
//         UNIT=SYSDA,DISP=(CATLG),SPACE=(TRK,(5,5),RLSE)
//SYSIN    DD   DSN=APPLRR.SELECT,DISP=SHR
//BKMBKEND EXEC BKMBPROC,BPARM='APPL(APPLRR)',COND=(0,NE,BACKUP)
//BKMBATCH.SYSIN  DD  *
 BKUPEND CHECK-CHANGED-SINCE-LAST-BKUP(N)   +
    NO-BKUP-FOUND-RC(4)
    BAD-BKUP-RC(8)
```

Consider setting the RSP retain cycles to a value greater than 1 even though the cycles to be included in the selection data is set to 1. This way, if you accidentally run BKMAPLEN twice, resulting in the current cycle containing nothing, you can then set the cycles to be included in the Selection Data Set to 2 and pick up the correct data.

If BKMAPLEN is run again accidentally, and there are new records in the Automated Critical Data Identification Application Data Inventory database because the application is currently running, there is no way to build the Selection Data Set from just the -1 generation of RSP cycles.

It might be beneficial to retain multiple RSP cycles for comparison purposes, but you should balance this benefit against the room required in the Automated Critical Data Identification Application Data Inventory database.

Increasing the RSP buffer size

If a system is losing records, you need to increase the number of buffers. To increase the buffer size, complete these steps.

1. Stop RSP.
2. Increase the buffers by 5000 on the system that is losing records.
3. Restart RSP.

RSP IPL considerations

When RSP is shut down normally, it flushes the data space to the overflow file. The records remain in the overflow file until RSP is started, at which time the records are written to the Automated Critical Data Identification Application Data Inventory database.

- Shutdown — While RSP is running, the possibility exists that it is collecting data; therefore, the process for a scheduled IPL is as follows:
  1. Drain the initiators.
  2. Shut down the job scheduler.
  3. Shut RSP down.
- Startup — RSP data collection is critical to ensure that all possible data is collected; therefore, RSP should be one of the first products started as soon as the system starts.

RSP and system outages

This information describes RSP and system outages.

If you experience a system outage, you lose all of the RSP data in the data space. However, RSP is never far behind in either updating the Automated Critical Data Identification Application Data Inventory database, or writing the records to the overflow file (generally, less than 30 seconds). Steps that represent that portion of data are most likely regenerated to place the application at a good starting point. If STEPA completes just before the outage and the application continues with STEPB, then those records are lost.

If you experience a system outage and you are using Record Level Sharing, the records could be backed up in the coupling facility. There is a failover mechanism: if one coupling facility fails, it switches to another.

RSP and the overflow data set

The overflow data set is used when the buffers fill up either as part of normal processes during busy times, when the buffers are too small, or when the RSP update task has been stopped to reorganize the Automated Critical Data Identification Application Data Inventory database.

Data written to the overflow data set could signal a problem if the overflow is not large enough to accommodate the data collected during a reorganization, during a busy period, or because the buffers have not been increased appropriately, causing data to spill over to the overflow more often. If you

want to monitor when data goes to the overflow data set, signaling a possible overflow file out-of-space condition, two messages signal this event:

- Message BKMK3810I OVERFLOW DATASET LOAD STARTED appears when the buffers are about to be unloaded and written to the overflow data set.
- Message BKMK3811I OVERFLOW DATASET LOAD FINISHED appears when all the buffer data has been written to the overflow data set.

The records intercepted by RSP are stored temporarily in buffers created in a data space. The database update task unloads the data space buffers and updates the Automated Critical Data Identification Application Data Inventory database. If the database update task lags behind due to a large number of records, the buffers expand. If still more room is required, the records overflow to the overflow data set.

For performance reasons, buffer sizes must be monitored and enlarged periodically if Automated Critical Data Identification is collecting more data. Performance is lost if only the overflow data set is increased. You can display RSP statistics using the **RSP Monitor** menu's **Display RSP Statistics** option.

A B37 could occur for several reasons:

- The buffers and overflow data set are not large enough to accommodate the amount of data you are collecting. If this occurs, increase the buffer size by 5000 only on the system where the problem is occurring, and use the RSP Monitor to ensure this amount is large enough to support the data collected.
- It is possible that the RSP update task, ACDIUPDT is not running. This can occur if the update task is not running to reorganize the Automated Critical Data Identification Application Data Inventory database. In this case, Automated Critical Data Identification continues to collect and write data to the buffer space and then to the overflow data set. If this occurs, increase both buffer size and overflow size to support a regular reorganization.
- It is possible that the RSP update task, ACDIUPDT, is not running for another reason.

If the overflow data set already encountered a B37, you have already lost data. If the ACDIUPDT task is not running, start it. If you determine you need to increase the size of the overflow data set, stop RSP and increase the overflow data set space in the BKMINI Overflow_Parms parameter. For example:

```
Overflow_Parms   = BLKSIZE(0) UNIT(SYSALLDA) SPACE(x) CYLINDERS +
                   RACCHECK(YES)
```

To prevent this from happening:

1. If the ACDIUPDT task was not running, determine why it was not running.
2. If the ACDIUPDT task was running, check RSP Monitor. It is possible that the data space buffers are no longer large enough to support the amount of data you are collecting. Check the current day and previous day's information to see if data has been lost.
3. Check the **RSP PROCESSING STATISTICS** panel and review these items:
   - ARE LOSSES OCCURRING NOW? – Is it set to Y?
   - NUMBER OF TIMES LOSSES OCCURRED – Is this value greater than zero?
   - RSP DATA SPACE BUFFERS – Note the number of buffers.
   - HIGHEST NUMBER OF BUFFERS USED – Does the number equal the total number of buffers?

RSP JCL members

These JCL members are used by RSP.

BKMRSPSU
The RSP file allocation job that allocates and initializes the message log and job flat data sets used by RSP.

BKMSTSKR
The started task that installs the IEFU83 exit and starts the subtask that writes records to the Automated Critical Data Identification Application Data Inventory database.

BKMSTSKR controls the RSP tasks. There is one BKMSTSKR task for each image.

The BKMSTSKR task is started with a job name of RSP.

BKMJSWAP
Signals to RSP to refresh the job tables from the job flat file. You need to run BKMJSWAP from only one image.

The program that is run is also the second step of BKMJRFSH. Normally this function is accomplished by running BKMJRFSH as it also rebuilds the job flat file.

BKMBLDJB
Builds the job flat file from the Automated Critical Data Identification Application Data Inventory database. You need to run BKMBLDJB only once before RSP is started for the first time and it needs to be run on only one image. For subsequent changes to application job lists, this function is normally accomplished by running BKMJRFSH.

BKMJRFSH
Rebuilds the job flat file, and then initiates a refresh of the internal RSP job reference table on each image from the job flat file.

BKMJRFSH is a combination of BKMBLDJB and BKMJSWAP. You need to run BKMJRFSH on only one image.

Chapter 12. Identifying application data sets using SMF Scan

Identifying application data sets through SMF Scan is an optional step that you can perform after job collection and after you refresh the RSP job table. The SMF records are written to the Automated Critical Data Identification Application Data Inventory database.

While it is possible to identify most of the application data by examining the JCL that is used to run the application, the data that is collected by RSP JCL Capture provides incomplete information about the application. (JCL data provides information about the disposition of data sets — however, it does not indicate whether the data sets were used for input or output, nor does it provide enough information about the size of the data set or whether or not the data set is dynamically allocated.) To generate an accurate Selection Data Set, Automated Critical Data Identification must obtain this information from the system SMF records.

SMF data can be collected dynamically, using RSP SMF Capture (the recommended method), or in batch using SMF Scan. SMF Scan is intended for installations that are not ready to implement RSP to collect SMF data — or for those installations that want to obtain an historical analysis of an application.

You can submit the SMF Scan at any time because the Automated Critical Data Identification data set information is merged with existing entries for the application.

To initiate an SMF Scan, specify the SMF history files and any filtering options and then submit a batch job to examine SMF record information. The data set information is saved in the Automated Critical Data Identification Application Data Inventory database.

Note: SMF Scan reads the job flat file for a list of eligible jobs.

SMF Scan guidelines

Review these SMF Scan guidelines to help you determine the method you should use to identify application data sets.

Determining the amount of data to collect

The data SMF Scan collects should reflect at least one complete processing cycle for the application. For example, if the application runs weekly, then at least one week of SMF data should be collected.

Note: If a sample includes more than one processing cycle, an incorrect analysis could result.

SMF record types

Various SMF record types are used by Automated Critical Data Identification during SMF analysis. Before using Automated Critical Data Identification, make sure that you are collecting the required record types (these same record types are also used by RSP).

Considerations and limitations

Note these SMF Scan considerations and limitations:

- SMF Scan data is not recommended for rerun recovery because it can be difficult to determine the start of the application. This is because rerun recovery relies on the first reference of a data set, but what SMF Scan determines as the first reference might not be the first reference at all, thus, unpredictable results can occur.
- SMF Scan data is more reliable for forward recovery using the **EVALUATE ALL DATASETS AS CRITICAL** option because both inputs and outputs are considered critical. But again, it might be difficult to determine the start of the application.

- Because JCL data cannot be collected in addition to SMF data, unopened data sets and steps that are skipped within a job will be excluded from the SMF Scan process.
- Typically, the GDGs referenced in SMF will be out of date — or they will no longer reside on the system.
- Typically, many non-GDG data sets will have been deleted.

In summary, while SMF Scan will give you an idea of the data that has been used, it is recommended that you not rely on it exclusively as foolproof data for analysis.

SMF record requirements

Automated Critical Data Identification uses SMF records to determine whether or not a data set belongs to an application, the size of a data set, and whether or not it is a critical data set required for backup.

If no SMF records are available for a data set, Automated Critical Data Identification must depend upon information gathered from RSP JCL Capture (which can be less accurate because it includes only JCL data). Therefore, you should ensure that SMF records encompassing at least one full cycle of the application are available for processing.

Required SMF record types

Table 3 on page 126 shows the SMF record types required by SMF Scan:

Table 3. Required SMF records and descriptions

Record Type	Description
14	INPUT or RDBACK Activity
15	OUTPUT Activity
30	End of Step or End of Job
61	ICF Define Activity
64	VSAM Component or Cluster Status
65	ICF Delete Activity
66	RENAME (alter)

Although processing is faster if the data sets specified for the scan contain only SMF record types 14, 15, 30, 61, 64, 65 and 66, they can contain any SMF data. Automated Critical Data Identification ignores any records that are not of the correct SMF type, or those that cannot be used due to invalid span bits or record lengths.

Collecting the required SMF records

You can use the SMF dump program (IFASMFDP) or the system sort to extract records from the SYS1.MANx SMF data sets. The following JCL example shows how to use IFASMFDP to collect the SMF records required for SMF Scan:

```
//STEP1   EXECPGM=IFASMFDP
//INDD1   DD  DSN=SYS1.MANX,DISP=SHR
//OUTDD1  DD  DSN=SMF.RECS.FOR.ACDI,DISP=SHR
//SYSPRINT  DD  SYSOUT=*
//SYSIN  DD *
INDD(INDD1,OPTIONS(DUMP))
OUTDD(OUTDD1,TYPE(14,15,30,61,64,65,66))
```

Creating required batch SMF data sets

Complete these steps to create the batch SMF data sets required for SMF Scan.

1. Allocate these files for batch SMF use:
   - RSP MSGLOG — See the value specified for the INI token **RSP_LOG**.
   - RSP JOBFLAT — See the value specified for the INI token **RSP_JOBFLAT**.

   Note: If you have decided not to start RSP on your system at this time, you must allocate these files for batch SMF use.
2. Ensure the load library has been APF authorized.
3. Edit the *prefix.value*.SBKMJCL member (BKMRSPSU).
4. Delete the last step: RSPMQBLD (this step is not required for batch SMF Scan).
5. Submit the job.

Using SMF Scan to collect SMF data

To use SMF Scan to collect SMF data, complete these steps.

1. From the Automated Critical Data Identification Main Menu, select the **SMF SCAN SETUP** option and press Enter.

   The **SMF SCAN SETUP** panel is displayed.
2. In the **SMF DSN** field, specify the names of the SMF batch files.

   You can specify a maximum of four data set names. The data sets you specify can be any valid SMF data sets, including generation data sets. All generations of a GDG can be specified by entering the GDG base name without a relative generation number.

   Note:
   - You must specify at least one data set name.
   - If you need to specify additional data set names, you can do so by editing the JCL prior to submission.
3. In the **APPLICATION** field, specify the name of the application whose SMF data you want to collect and press Enter.

   Automated Critical Data Identification examines only those SMF records with job names that match those belonging to the specified applications, as determined during the job name collection step.

   Note: Repeat this process for each application whose SMF data you want to collect using SMF Scan.
4. To specify optional parameters, type Y in the **OPTIONAL PARAMETERS** field.

   The **SMF SCAN OPTIONAL PARAMETERS** panel is displayed.
5. Type S to select these options:
   - **DISPLAY LIST OF PREVIOUSLY ENTERED APPLICATIONS** — Display a list of the applications that have been specified in this session— and their optional parameters.
   - **DISPLAY CURRENT JOBS COLLECTED** — Display a list of the job names in the Automated Critical Data Identification Application Data Inventory database for the application.
6. When you are ready to schedule the job, type S in the **Schedule SMF SCAN JOB** field and press Enter.

   The job card is displayed.
7. Press Enter to submit the job.

   The job submission panel is displayed.
8. Select one of these options to process the job:
   - To submit the job, type S in the **SUBMIT Job For Execution** field and press Enter.
   - To edit the JCL prior to submitting the job, type S in the **EDIT JCL Prior To Job Submit** field and press Enter.

Note: If the SMF files are located on tape and you are reading many SMF tapes, you might want to select **EDIT JCL Prior To Job Submit** and modify the JCL to use the '**UNIT=AFF=SMFFILE**' parameter on the tape files so you only use one drive at a time. (This parameter is not included automatically because the SMF files might be located on disk.)

- Press F3 to exit without submitting the job.

Note: All data collected is not saved after you leave **SMF SCAN SETUP** panel. Therefore, you must save the JCL if you want to avoid re-specifying application data.

Ensure the job completes with return code 0 and the message at the bottom of the SYSOUT indicates that records have been added to the Automated Critical Data Identification Application Data Inventory database. For example:

```
ACDIDB RECORDS ADDED/UPDATED 37
```

Optional SMF Scan parameters

The optional SMF Scan parameters on the **SMF SCAN OPTIONAL PARAMETERS** panel limit the amount of data collected for each application.

Parameters

STARTING DATE
The starting date (YYYYDDD) of the initial record to be examined. SMF records written before this date are not analyzed.

STARTING TIME
The starting time (HH.MM) of the initial record to be examined.

Note: You cannot specify a starting time without specifying a starting date.

ENDING DATE
The date (YYYYDDD) of the last SMF record to be analyzed. Records written after this date are not used.

ENDING TIME
The ending time (HH.MM) of the last SMF record to be analyzed.

Note: You cannot specify an ending time without specifying an ending date.

STARTING JOBS
A list of jobs that initiate SMF analysis whenever any SMF record matches any of the jobs. When a match is found, Automated Critical Data Identification analyzes SMF records for that job until it reads that job's type 30 record.

ENDING JOBS
A list of jobs that stop SMF analysis when any record matches any of the jobs and a type 30 record is read for that job. The first job that matches is used.

Note: The starting date and ending date parameters take precedence over the starting job and ending job parameters. For example, if starting date and starting job are specified, but the first record that matches the starting job was created before the date in the starting date parameter, then that record and any others that have dates before the starting date are not analyzed.

Viewing the results of the batch SMF Scan

View the results of the SMF Scan from the SYSPRINT file or by using the **STATUS** option on the **Application AutoBuild Options** menu.

- Use ISPF to view the SYSPRINT file which contains summary information about the processing of each application.
- Use the **STATUS** option on the **Application AutoBuild Options** menu to view results as follows:

- To view the data sets collected, from the **APPLICATION Display Selections** menu, select the **Display Dataset Detail Records** option.

  The Combined Data Set List is displayed when you select the **Display Dataset Detail Records** option from the **APPLICATION Display Selections** menu.

  Rows indicating a REC TYP of SMF are records created from SMF Scan. Each time you run SMF Scan, the SMF cycle number increments. If more than one cycle is retained, the current cycle is indicated by both a cycle number and an asterisk in the **CC** (current cycle) column.
- To view the data set records for inclusion in the SDS, From the **APPLICATION Display Selections** menu, select the **Display Current JCL/SMF DSN Entries** option. The **DATASET RECORDS FOR INCLUSION IN SDS** panel is displayed.

Generating diagnostic information during SMF Scan

You can produce diagnostic information during SMF Scan when SMF records are encountered that are considered bad or missing data.

Additional diagnostic information will be produced only if the job step JCL contains a DD statement with DDNAME ERRFILE.

- For example:

```
//SMFSCAN  EXEC PGM=BKMSMFSC
//STEPLIB  DD   DISP=SHR,DSN=prefix.value.SBKMLOAD
//SYSOUT   DD   SYSOUT=*
//SYSPRINT DD   SYSOUT=*
//INI      DD   DISP=SHR,DSN=prefix.value.SBKMPARM(BKMINI)
//RSPLOG   DD   DISP=SHR,DSN=ASPX.PRDLIB.MSGLOG
//SISLOAD  DD   DISP=SHR,DSN=prefix.value.SBKMLOAD
//ACDIDB   DD   DISP=SHR,DSN=prefix.value.ACDIDB
//SMFFILE  DD   DISP=SHR,DSN=PDABC.SMF.EXTRACT1
//ERRFILE  DD   DSN=PDABC.SMF.ERRFILE,DISP=(,CATLG,DELETE),
//         UNIT=SYSALLDA,
//         SPACE=(CYL,(5,5),RLSE)
//TEMPFILE DD   DSN=&&SMFTEMP,
//         DISP=NEW,
//         UNIT=SYSALLDA,
//         DCB=(LRECL=4096,BLKSIZE=27998,RECFM=VB),
//         SPACE=(CYL,(5,5),RLSE)
//SYSIN    DD  *
  APPL (BK000001)
//
```

Sample output

The additional diagnostic information provided during SMF Scan includes an SMF record rejection reason message and the raw SMF record. The SMF record rejection reason message is written to the SYSPRINT file and contains the TYPE/CREATION DATE/TIME/JOBNAME/SYSTEMID/REASON for the bad SMF record. For example:

```
SMF15 2009.279/07.40.03.45/DUMPSMF /DVLP DROPPED – UNKNOWN TYPE
```

The bad SMF record is written to the ERRFILE (in this example: PDABC.SMF.ERRFILE).

Chapter 13. Using Verification to create a Selection Data Set

Use the Verification program to build the Selection Data Set when you use SMF Scan to gather data.

Before you use Verification, note the following:

- Verification processing is partially controlled by global options and settings contained in the BKMINI product parameter library member.

  You can specify values in the BKMINI member to force, or alter, Verification options for all applications defined to Automated Critical Data Identification.
- You must use BKMAPLEN to build the Selection Data Set at least once before you build it using Verification.

The options that you select for Verification are used during the RSP process of reconstructing the application Selection Data Set entries. You can change any options at a later time and the new values are used the next time the Selection Data Set is rebuilt.

1. From the Automated Critical Data Identification Main Menu, select the **SELECT/CREATE APPL** option and press Enter.

   The **SELECT Current/NEW Application Name** panel is displayed.
2. Specify the application.
3. From the **Application AutoBuild Options** panel, select the **VERIFICATION** option.

   The **Selection Data Set Build** panel is displayed.
4. Choose one of these processing options:

Option	Description
F	Build the Selection Data Set in the foreground. **Note:** Select **F** if this is the first time Verification has been used for this application.
B	Submit a batch job to build the Selection Data Set. (This option is recommended if you have 1500 or more records to process.)

5. Follow the steps according to the processing option you selected in step 4.

Verification extracts the records from the Automated Critical Data Identification Application Data Inventory database, processes any filters, and builds the Selection Data Set.

Building the Selection Data Set in the foreground

When you select this option, the **SELECTION DATASET BUILD FOREGROUND OPTIONS** panel enables you to change the application Setup options for the selected application and specify any filtering or Verification changes.

1. In the **SMF DATA FROM RSP OR BATCH SMF SCAN?** field specify the source of the SMF data:
   - **R** —RSP data.
   - **S** — Batch SMF data.
2. To change the application Setup options, type Y in the **CHANGE APPLICATION SETUP OPTIONS** field; otherwise, specify N.
3. In the **SELECTION DATASET NAME FOR THIS BUILD** field, type the name of the Selection Data Set for this build.

   Note: The Selection Data Set name you specify is used for the current build only.

4. If you specified Y in the **CHANGE APPLICATION SETUP OPTIONS** field, the **Application SETUP Options** panel is displayed. (This is the same panel that is displayed when you select the **SETUP** option from the **Application AutoBuild Options** panel.) Make any filtering or Verification changes and press Enter.

Building the Selection Data Set in batch

When you select this option, the **SELECTION DATASET BUILD BATCH OPTIONS** panel enables you to change the Selection Data Set options for the selected application and build the Selection Data Set in batch.

1. To change the application configuration settings, type Y in the **CHANGE SELECTION DATA SET OPTIONS** field.

   Note: With batch processing, you cannot change the Selection Data Set name. However, if you specify Y in the **CHANGE SELECTION DATA SET OPTIONS** field, you can change the Selection Data Set names in the Automated Critical Data Identification Application Data Inventory database when the **Application SETUP Options** panel is displayed.
2. Press Enter.
3. Select one of these options to process the job:
   - To submit the job, type S in the **SUBMIT Job For Execution** field and press Enter.
   - To edit the JCL prior to submitting the job, type S in the **EDIT JCL Prior To Job Submit** field and press Enter.
   - Press F3 to exit without submitting the job.

Ensure the job completes with a return code of 0.

Verification output selection formats and data set types

Verification separates an application's Selection Data Set entries into separate lists based on data set type.

The Selection Data Set entries are separated into three lists:

- INCLUDE — For primary DASD data sets.
- ALLOCATE — For primary DASD data set catalog information only (data sets with no data).
- ACCOMPANY — For user tape data sets.

Not all data set types are supported by all output selection formats. Table 4 on page 132 summarizes the data set entries that are automatically constructed based upon the output format specified for a Selection Data Set.

Table 4. Output selection format and supported data set types

Output selection format	**Supported data set types**
DFSMShsm (ABARS)	Primary DASD, ML1, and ML2. ALLOCATE and user tapes as ACCOMPANY are all supported.
DMS/OS (CA-Disk)	Only primary DASD data sets are supported. User tape data sets or ALLOCATE is not supported.
DFSMSdss (DSS)	Only primary DASD data sets are supported. Any ALLOCATE type entries will be processed and placed in the INCLUDE list for full backup.
FDR	Both primary DASD data sets and ALLOCATE entries are both supported. Any user tape data set entries will be eliminated.

Table 4. Output selection format and supported data set types (continued)	
Output selection format	**Supported data set types**
LIST	No distinct separation is made for data set entries. All data sets are listed using the same format.

ABARS Selection Data Sets

Automated Critical Data Identification supports ABARS as a default for the final output format of the Selection Data Set. Unless you change the default, ABARS is used. Because this format requires a defined ABARS aggregate you should review the information presented in this topic before using Verification.

During the process of defining an application to Automated Critical Data Identification, you can associate the application with an existing ABARS aggregate definition contained in your active SMS configuration.

When the application is verified to create the output Selection Data Set critical list, Automated Critical Data Identification uses the following process to locate the associated aggregate name:

1. Automated Critical Data Identification attempts to locate the associated aggregate name in your SMS configuration.
2. If the associated aggregate name is found, then its defined Selection Data Set name (or member) is used as the final selection container data set.
3. If the associated aggregate name does not exist, then Automated Critical Data Identification constructs a default Selection Data Set name. This default name can then be used when the final aggregate entry is constructed using ISMF (Interactive System Management Facility).

   Note: In order for ABARS to use a Selection Data Set, you must specify the name of the data set when the SMS aggregate group is defined. The ISMF application is used to define or update SMS aggregate group definitions.

For ABARS output formats, Automated Critical Data Identification does not require that the final aggregate name exist when Verification is used. However, you must define the aggregate to SMS before any ABARS ABACKUP scheduling. In this way, you can create future ABARS applications using Automated Critical Data Identification, and then define the aggregate definition at a later time.

If you change the output format of the application Selection Data Set using the **0 (SETUP)** option, you must verify the application to ensure the correct format and Selection Data Set name will be assigned.

ABARS Selection Data Set contents

An aggregate's Selection Data Set for ABARS is an 80-character fixed-length file that indicates which data sets are processed during ABARS aggregate backup. The aggregate's Selection Data Set provides the selection criteria to be used by ABACKUP.

These selection statements in an aggregate's Selection Data Set determine how data sets are processed during ABACKUP:

INCLUDE
Specifies the data sets to be included during any backup processing. The data sets specified by an INCLUDE statement are part of the critical data set list and are placed in the Selection Data Set.

Note: A Selection Data Set must contain at least one INCLUDE statement.

EXCLUDE
Specifies the data sets to be excluded from any backup processing. The data sets specified by an EXCLUDE statement are maintained in the Automated Critical Data Identification Application Data Inventory database for the application but are not placed in the Selection Data Set.

ACCOMPANY
(Used by ABARS only.) Specifies which user tape data sets should be considered as included in the application backup and which should be cataloged at recovery time.

ALLOCATE
(Used by ABARS only.) Specifies DASD only data sets (that do not contain data) to include in the backup critical list. At recovery time, these data sets will be created, but no data will be restored to them.

Note: The use of ACCOMPANY and ALLOCATE selection statements is unique to the ABARS backup platform. If not supported by other backup products, data sets assigned with these selection criteria cannot be placed in the Selection Data Set list.

Automated Critical Data Identification uses all available information to determine, for each data set, whether the data set should be placed in the INCLUDE, EXCLUDE, ALLOCATE, or ACCOMPANY lists. Data set name masks are not used in these lists. (The ACCOMPANYEXCLUDE and ALLOCATEEXCLUDE lists are empty because no data set name masking is used.)

ABARS output format example

This example shows a sample of the ABARS output format.

```
/*                                                                  */
/*  APPL NAME: CA7                                                  */
/*       DESC: CA7 APPLICATION       CREATED ON 01/10/2004          */
/*                                                                  */
/*                    DSN FORMAT: ABARS                             */
/*                          SYSID: ****                             */
/*                     EXEC MODE: DIALOG (OPTION-4)                 */
/*   RE-EVALUATE DATASET ENTRIES: YES                               */
/*       DATASETS EVALUATED FROM: SMF AND JCL                       */
/*    TAPE DATASETS AS ACCOMPANY: NO                                */
/*    USE SIZE FOR INC/ACC TAPES: NO                                */
/*                     CTLG CHK: ENABLED                            */
/*               PROCESS FILTERS: YES                               */
/*          NAME TYPE FOR GDSES: ABSOLUTE                           */
/*         GDSES EVALUATED FROM: SMF AND JCL                        */
/*     EXPAND BASE GDG REFERENCE: NO                                */
/*            ADD MISSING GDSES: MIN FOR FWD RECOVRY                */
/*    CATEGORY FOR MISSING GDSES: INCLUDE                           */
/*        CATEGORY FOR NEW GDSES: INCLUDE                           */
/*                   VDR FILTERS: ACTIVE=NO                         */

          INCLUDE(
USER1.ABARSMGR.JCLPROF                       /*DASD(SMS004) 2001.325*/ +
AMP.ACDIDB                                   /*DASD(TSO002) 2001.325*/ +
CA7.TESTGDG.G0004V00                         /*DASD(SMS003) *********/ +
CA7.TESTGDG.G0005V00                         /*DASD(SMS003) *********/ +
                 )
/*                                                                  */
         ALLOCATE(
CA7.ZOUT.LIST                                /*DASD(TSO002)         */ +
               )
/*                                                                  */
        ACCOMPANY(
CA7.ZTAPE                                   /*TAPE(021212) 2001.267*/ +
               )

/*           EXCLUDED DUE TO NORMAL EVALUATION:                      */
/*                                                                   */
/* DSN=USER1.LEAVE.THIS.OUT                    DASD */
```

Non-ABARS Selection Data Sets

If Automated Critical Data Identification identifies data that at some point during batch processing is either migrated or archived, non-ABARS formats cause the backup to fail because they cannot back up migrated or archived data.

If you are not using ABARS, ensure filters are set up to exclude migrated or archived data.

Non-ABARS Selection Data Set contents

For non-ABARS formats such as DFDSS, FDR, DMS, the contents of the entries are unique to the backup product being used. Automated Critical Data Identification creates these entries in the proper format based upon the selection data format specified during application setup.

DSS output format example

This example shows a sample of the DSS output format.

```
  /*  APPL Name:BELLALT                                                 */
  /*       Desc:APPLICATION CREATED BY:MSIS04T ON 01/10/04              */
  /*                                                                    */
  /*                    DSN FORMAT: DSS                                 */
  /*                         SYSID: ****                                */
  /*                     EXEC MODE: DIALOG (OPTION-4)                   */
  /*   RE-EVALUATE DATASET ENTRIES: YES                                 */
  /*       DATASETS EVALUATED FROM: SMF AND JCL                         */
  /*    TAPE DATASETS AS ACCOMPANY: NO                                  */
  /*    USE SIZE FOR INC/ACC TAPES: NO                                  */
  /*                     CTLG CHK: ENABLED                              */
  /*               PROCESS FILTERS: YES                                 */
  /*          NAME TYPE FOR GDSES: ABSOLUTE                             */
  /*         GDSES EVALUATED FROM: SMF AND JCL                          */
  /*     EXPAND BASE GDG REFERENCE: NO                                  */
  /*            ADD MISSING GDSES: MIN FOR FWD RECOVRY                  */
  /*    CATEGORY FOR MISSING GDSES: INCLUDE                             */
  /*        CATEGORY FOR NEW GDSES: INCLUDE                             */
  /*                                                 */
         DUMP DATA SET(INCLUDE(    -
PBACR.CR932M.PTP1006 /* DASD (A4PRAB) */ -
PCRAB.AB3906M.REPORT /* DASD (A4PRAB) */ +
PCRAB.AB3907M.REPORT /* DASD (A4PRAE) */ +
PCRAB.AB3908M.REPORT /* DASD (A4PRAE) */ +
SUFYP.SEND.BFTCNTL /* DASD (A4SUAL) */ -
SUOPP.PROD.DMSSLLB /* DASD (A4SUAY) */ -
SUOPP.PROD.LLBWK1 /* DASD (A4SUBH) */ -
SUOPP.PROD.PROC /* DASD (S1SYP3) */ -
                )) -
                INDYNAM((......)) -
                OUTDD(TAPE)
```

Adding data sets to the Selection Data Set during Verification

Add additional data set entries, such as tool box data sets, to the Selection Data Set during Verification by using filters.

- Use the INCL_EXTDSN, the ACCOMP_EXTDSN and ALLOC_EXTDSN filters to add additional data set entries to the Selection Data Set. Data set entries that are added using this process are evaluated in the same manner as all other data set entries. The exception to this is that these data set entries are not affected by the re-evaluate option.

  Note: ABARS rejects a Selection Data Set that does not have at least one INCLUDE statement.

Viewing the results of the Verification

Complete these steps to review the results of the latest Verification of an application Selection Data Set contents.

1. From the **Application AutoBuild Options** menu, select the **STATUS** option.

   The **APPLICATION Display Selections** menu is displayed.
2. To display information for the last Verification, type B (**B VERIFY (Database)**)

   An ISPF Browse session displays the current Selection Data Set.

Scheduling application backups

After Verification is complete and the Selection Data Set has been built, the application is ready to be backed up. The Selection Data Set can then be used as the input to the backup, or it can be processed prior to the backup.

- For example, if the ABARS format is the current output option for the application, you can use Advanced Backup and Recovery for z/OS to schedule ABACKUP through DFSMShsm.
- For non-ABARS formats, you must create and schedule the backup, using any data mover utility (DSS, FDR, and so on) for the Selection Data Set format created.

Using Verification to re-evaluate data

You can use Verification to re-evaluate data. For example, if you run BKMAPLEN and you want to create some filters or change some settings but want to determine the impact of your changes before you actually change the settings.

1. Modify filters and settings as needed.
2. Run Verification.

   Verification uses the last cycle of data to rebuild your Selection Data Set.

Chapter 14. Obtaining application information

You can obtain application related information through the Automated Critical Data Identification user interface and batch facilities.

Displaying a list of application information

For any application defined to Automated Critical Data Identification, you can view information about the application backup job, filters used, job collection, and data identified by RSP.

1. From the Automated Critical Data Identification Main Menu, select the **LIST APPLICATIONS** option.
   The **CREATE APPLICATION LIST** panel is displayed.
2. In the **ENTER APPLICATION OR MASK** field, do one of the following:
   - Press Enter or type an asterisk (*) to list all applications.
   - Specify an application name mask to display a list of selected applications. For example, you can specify BK00* to list all applications that begin with BK00.

   For each application, the **APPLICATION LIST** panel displays the application name, user ID of the user that last updated the application, the most recent event, the application description, and the date the application was created.

Displaying the status of an application

Complete these steps to display the status of an application.

1. From the Automated Critical Data Identification Main Menu, select the **SELECT/CREATE APPL** option.
   The **SELECT Current/NEW Application Name** panel is displayed.
2. Specify the application whose status you want to view.
3. Select the **STATUS** option from the **Application AutoBuild Options** menu.
   The **APPLICATION Display Selections** menu is displayed.
4. Select from these options:
   - **Display Current JOB List**
   - **Display Current JOB Anchor Chain**
   - **Display Dataset Detail Records**
   - **Display Current JCL/SMF DSN Entries**
   - **Display Current JCL/SMF DSN Entries with URD Information**

Displaying the current job list

Job list information is obtained during job collection. The **Application JOB Sequence List** panel displays the individual job element names for the application, in alphanumeric order.

- Select the **Display Current JOB List** option from the **APPLICATION Display Selections** menu to display a list of all jobs identified as belonging to this application.
  This information is obtained during the first step in the data collection sequence — job collection. The **Application JOB Sequence List** panel lists only the individual job element names for the application, in alphanumeric sequence order.

Application JOB Sequence List fields

The fields on the **Application JOB Sequence List** panel provide information about the application job sequence.

Fields

Note: Depending upon the job scheduling product that is used for collection, the display columns and headings might be different for your installation.

Application
The name of the current application.

Collection Method
The name of the job collection process used:

- TWS (OPC)
- CA7
- CONTROL-M
- ESP
- JOBTRAC
- ZEKE MVS

Jobname
The application job name.

Misc Descriptive Info
Information such as the date and time the job collection occurred.

Displaying current job anchor chains

The **Application ANCHOR Job Structures** panel lists each application anchor job, or anchor synchronization point, with the associated (succeeding) job structure entries.

- From the **APPLICATION Display Selections** menu, select the **Display Current JOB Anchor Chains** option to display job anchor structure for all previously collected jobs.

Application ANCHOR Job Structures fields

The fields on the **Application ANCHOR Job Structures** panel provide information about anchor job structures.

Fields

Note: Depending upon the job scheduling product used for collection, the display columns and headings might be different for your installation.

APPLICATION
The name of the current application.

JOB/Anchor
The names of the anchor jobs in the application. These jobs are generally the first job in a series.

Type
The type of MVS job scheduler used.

- TWS (OPC)
- CA7
- CONTROL-M
- ESP
- JOBTRAC

- ZEKE MVS

JOB Structure Elements
The remaining jobs in the series.

Displaying data set detail records

Complete these steps to display information about the data sets identified with a specific application during RSP JCL Capture, RSP SMF Capture, or SMF Scan.

- From the **APPLICATION Display Selections** menu, select the **Display Dataset Detail Records** option.

  The **DATASET DETAIL RECORDS** panel shows all data set detail records for this application currently located in the Automated Critical Data Identification Application Data Inventory database.

  Note:

  – If no records are displayed, it is because none of the application jobs have been executed or if they have, the records still reside in the data space buffers or overflow file. In this case, use RSP Monitor to flush the buffer.
  – Records captured by RSP might show a cycle number that is one greater than the current cycle. If this occurs, RSP has written a record for the next cycle but the BKMAPLEN process to close out the cycle has not occurred yet. Current cycle records are those that are used in creating a Selection Data Set. For records captured by RSP, note that from the **Setup** panel you can choose to include more than one cycle's worth of data in the Selection Data Set.
  – RSP SMF Capture, RSP JCL Capture or SMF Scan records information for the first reference to a data set within the job stream.

    RSP JCL Capture or RSP SMF Capture also records the first URD of a data set.

    Tip: To display URD information, select the **Display Current JCL/SMF Entries with URD Information** option from the **APPLICATION Display Selections** menu.

Data Set Detail Records fields

The fields on the **Data Set Detail Records** panel provide information about data set detail records.

Fields

APPL
The application name.

CURRENT CYCLES
The current cycle number for RSP (RSP SMF Capture), JCL (RSP JCL Capture) and SMF (SMF Scan).

CM (command)
Specify S for a full screen display of the selected data set.

DATASET NAME
The name of the data set.

REC TYP
The source of the record:

- RSP (SMF records captured by RSP SMF Capture using IEFU83)
- JCL (interpreted JCL captured by RSP JCL Capture using IEFUJI)
- SMF (records collected by SMF Scan)

Cycle
The cycle number of RSP SMF Capture, RSP JCL Capture or SMF Scan.

CC
The current cycle. An asterisk (*) indicates that the record is in the current cycle.

JOBNAME
The job name from RSP JCL Capture.

DDNAME
The ddname from RSP JCL Capture.

PGM
The name of the program executed for the step that referenced the data set.

DSP
The disposition from the SMF record.

SYS ID
The system ID.

DATE
The date the data set was referenced by RSP JCL Capture.

TIME
The time the data set was referenced by RSP JCL Capture.

JCL NDP
JCL normal disposition. The second subparameter of the JCL DISP parameter.

JCL ADP
JCL abnormal disposition. The third subparameter of the JCL DISP parameter.

JOBSTEP
The job step name.

PROCSTEP
The proc step name.

SMF TYP
The SMF record type.

SMF UNIT
The device UNIT from the SMF record.

SMF VOLUME
The volume from the SMF record.

SMF ACCESS
The type of data set access from the SMF record.

SMF DSORG
The data set organization from the SMF record.

SMF DSN-SIZE
The size of the data set in kilobytes.

Displaying current JCL and SMF DSN entries

You can view current JCL and SMF DSN entries to determine why a data set was placed in a given Selection Data Set category. The Selection Data Set category is shown, determined by evaluating the data set information, and from filter overrides.

- From the **APPLICATION Display Selections** menu, select the **Display Current JCL/SMF DSN Entries** option to display a combined data set list containing information about the data sets identified with this application from RSP SMF Capture, RSP JCL Capture, or SMF Scan.

  The **DATASET RECORDS FOR INCLUSION IN SDS** panel displays records created as a result of running Verification (either from ISPF or from an RSP BKMAPLEN step).

  The first step of Verification selects data set detail records based on cycles, and combines information from RSP SMF Capture, RSP JCL Capture, or SMF Scan into a single record per data set. Data set records in this display could have been created from filters and from GDG rules (additional generations might have been added). Important in this display is the filter, which might have influenced the Selection Data Set category.

Data Set Records for Inclusion in SDS fields

The fields on the **Dataset Records for Inclusion in SDS** panel provide information about the data set records selected for inclusion in the Selection Data Set.

Fields

APPL
The application name.

CM (command)
Specify S for a full screen display of the selected data set.

DATA SET NAME
The name of the data set.

SDS STAT
The status of the Selection Data Set. This field displays the category chosen for the Selection Data Set:

- INCLUDE
- EXCLUDE
- ALLOCATE
- GDGBASE

SMF UNIT
The device UNIT from the SMF record.

SMF VOLSER
The volser from the SMF record.

JCL SDP
The JCL scheduler disposition. The first subparameter of the JCL DISP parameter.

JCL NDP
The JCL normal disposition. The second subparameter of the JCL DISP parameter.

JCL ADP
The JCL abnormal disposition. The third subparameter of the JCL DISP parameter.

JCL JOB NAME
The job name from RSP JCL Capture.

JCL CYCLE
The cycle number of RSP JCL Capture.

SMF SYID
The SMF system ID.

SMF RTY
The SMF record type.

SMF CYCLE
The cycle of the SMF data (RSP SMF Capture or Batch SMF Scan).

SMF JOBNAME
The job name from the SMF record.

SMF DDNAME
The ddname from the SMF record.

SMF ACCESS
Type of data set access from the SMF record.

SMF DSP
The disposition from the SMF record.

SMF ORG
The data set organization from the SMF record.

SMF DSN-SIZE
The size of the data set in kilobytes.

Displaying expanded current JCL and SMF DSN data

This information describes how to display expanded data for a selected data set.

- Type an S in the **CM** (command) field on the **DATASET RECORDS FOR INCLUSION IN SDS** panel.
  The **DSR DATA** panel appears.

For example:

```
                                  DSR DATA
 Command ==>                                              APPLICATION ==> APPL0001

  DATASET NAME: BK.XYZ.GDG3.G1004V00
  STATUS: INCLUDE     REASON:
  FILTER HIT: INCLUDE    FILTER SCOPE: LOCAL      FILTER TYPE: FORCE INCLUDE
  FILTER: NEW GDS REQUESTED AS INCLUDE
  GDS: NEW GDG = INCL/ALLOC
                                                                        More:     +
  SMF DATA   RECORD TYPE: 00
  SYSID   CYCLE NUMBER DATE      TIME
  ****    0000000000   0000.000  00.00.00.00
  JOBNAME   DDNAME    PROGRAM   ACCESS    DISP  UNIT  VOLSER  ORG  SIZE /K
  ********  ********  ********  ********  ***   DASD  TSO002  **   00000000
  JOBSTEP   PROCSTEP  JCLGEN   SMFSRCE
  ********  ********            *****
  JCL DATA
  SYSID   CYCLE NUMBER RDRDATE   RDRTIME
  DVLP    0000000117   2003.045  10.46.52.08
                               SCHD  NORM  ABNM
  JOBNAME   DDNAME    PROGRAM   DISP  DISP  DISP JOBID
  ASPTST02  SYSUT2    IEBGENER  NEW   CAT   CAT  JOB20050

  JOBSTEP   PROCSTEP  JCLGEN    INDIRECT REFERENCE
  S5                  +1        NO
  URD DATA   RECORD TYPE: 00
  SYSID   CYCLE NUMBER DATE      TIME
  ****    0000000000   0000.000  00.00.00.00
  JOBNAME   DDNAME    PROGRAM   ACCESS    DISP  UNIT  VOLSER  ORG  SIZE /K
  ********  ********  ********  ********  ***   ****  ******  **
00000000
```

Figure 4. DSR data

This display includes filter information and Update, Rename, and Delete (URD) information. If the reason for the status is not EVALUATION, a filter caused the Selection Data Set category to be other than what normal evaluation might have produced.

GDS DSR record action

GDS DSR record action information helps provide insight to the eventual status of GDS DSR records when you display expanded information for current JCL/SMF DSN entries.

Possible GDS actions

ADDED TO SATISFY MISSING GDS OPTION
GDSs were added because of switch setting for missing GDS.

ONLYNEW GDS REQUESTED AS INCLUDE
When forward recovery and new GDS only reference switch set to include.

ONLYNEW GDS REQUESTED AS EXCLUDE
When forward recovery and new GDS only reference switch set to exclude.

NEW GDS REQUESTED AS INCLUDE
When no forward recovery and new GDS switch set to include.

NEW GDS REQUESTED AS ALLOCATE
When no forward recovery and new GDS switch set to allocate.

REPLACED BY NEW GDS (ADDMISS=F)
When forward recovery all not new GDSs excluded when new GDSs exist.

REL GDS IGNORED BY GDG RULES
Relative GDS created using SMF data but no JCL info in DSR record.

ABS GDS IGNORED BY GDG RULES
Absolute GDS created using JCL data but no SMF info in DSR record.

RELATIVE GDS NAMES NOT SELECTED
Only absolute GDSs wanted.

ABSOLUTE GDS NAMES NOT SELECTED
Only relative GDSs wanted.

Note: Because the comment is limited to 44 characters, it cannot be detailed.

Displaying current JCL and SMF DSN entries with URD information

This information describes how to display the data sets that were found by RSP to have been updated, renamed or deleted.

- From the **APPLICATION Display Selections** menu, select the **Display Current JCL/SMF DSN Entries wit URD Information** option to display data sets that were found by RSP to have been updated, renamed or deleted (URD).

  The **DATASET RECORDS WITH URD INFORMATION** panel appears.

Data Set Records with URD Information fields

The fields on the **Dataset Records with URD Information** panel provide information about data set records with Update, Rename, and Delete (URD) information.

Fields

APPL
The application name.

DATASET NAME
The name of the data set.

URD JOBNAME
The job name from the URD record.

URD ACCESS
The type of data set access from the URD record.

URD DSP
The disposition of the data set from the URD record.

URD DDNAME
The ddname from the URD record.

URD DATE
The date the data set was referenced by the application job step (from the URD record).

URD TIME
The time the data set was referenced by the application job step (from the URD record).

REC TYPE
The record type.

SMF TYPE
The SMF record type.

Browsing a Selection Data Set

There will be occasions where you need to browse the contents of a Selection Data Set.

1. From the **APPLICATION Display Selections** panel, select one of these options:
   - **VERIFY (One time build)**.
   - **VERIFY (Database)**
   - **(RSP)**

   Note: N/A indicates there is no cataloged Selection Data Set for that category.

   An ISPF Browse session displays the data sets associated with the application and the selection criteria status: INCLUDE, EXCLUDE, ACCOMPANY, or ALLOCATE. The content and format of the application's Selection Data Set is specific to the application's output format. Each output Selection Data Set contains both the active entries and commented entries.

   - The active entries and keywords are specific to the application's output format (ABARS, FDR, and so on).
   - The commented entries, listed below the active entries, represent data set names excluded by Verification due to either filtering, noncritical and (or) other conditions.

   If the data set is in use or cannot be found, the message `DATASET NOT FOUND OR DATASET IN USE` is displayed.
2. When you are finished browsing the data set, press F3.

   The **APPLICATION Display Selections** menu is displayed.

Chapter 15. Filters and controls

Filters and controls limit the job name and data set entries Automated Critical Data Identification collects in the Selection Data Set and the Automated Critical Data Identification Application Data Inventory database.

Types of information that can be filtered

For each application, Automated Critical Data Identification collects and saves two types of information:

- Job entries and job anchor structures
- Data set entries

Filters enable you to limit the scope of the application job entries, job anchor structures, and the application data set entries saved in the Automated Critical Data Identification Application Data Inventory database. When filters reside in the Automated Critical Data Identification Application Data Inventory database, data set names will be included in, or excluded from, the Selection Data Set each time the Selection Data Set is built.

When to define filters

You can define filters before you collect any data so that the analysis includes them when you run BKMAPLEN or Verification. Alternatively, you can define them after data collection and use Verification to analyze the results and rebuild the Selection Data Set.

Note: Verification uses the DSD records for the last successful cycle. These records can be reused to determine the results that would be achieved by adding new filters — without affecting the current cycle.

Processing of filters for each collection process can be enabled or disabled for each application by selecting **3 (FILTERS)** from the **Application AutoBuild Options** menu.

Filter levels

There are three levels of filtering: *Universal*, *Global*, and *Local*. Data set filtering is controlled by all three levels of filters. Job collection filtering is controlled through the use of Local filters only.

Figure 5 on page 145 provides an overview of filtering levels.

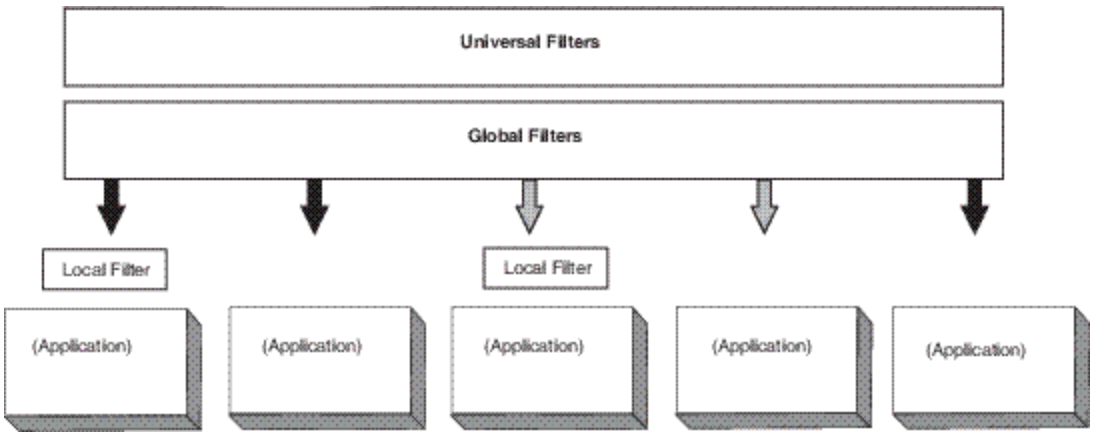


Figure 5. Levels of filtering

Universal filters

Universal filters are always processed.

Universal filters apply to all Automated Critical Data Identification applications and override any Automated Critical Data Identification application. Universal filters are processed prior to any other filters.

Universal filters can be used for any data set, for example:

- SYS1 data sets
- Data sets backed up by full volume dump
- Data sets you do not want any application defined to Automated Critical Data Identification to back up

Usage

Universal filters are useful if you are backing up system libraries and third-party product libraries and you want to ensure that no application can inadvertently back up a data set of this nature (and subsequently restore it on top of the version restored by the system backup). Universal filters can also be useful if you are missing a data set (you can check the Universal filters for a match).

Note:

- An application name of UNIVERSAL is not permitted.
- You cannot override Universal filters.
- SAF controls are allowed for Universal filters.

Global filters

Automated Critical Data Identification processes Global filters when appropriate, and processes them in a relative sequence to local application filters. There is one list of Global filters for each filter category and each category is either enabled or disabled for a specific application.

Usage

Global filters are useful, if for example, you have PROD.PROCLIB and PROD.JCLLIB files and you want most of your Automated Critical Data Identification applications to exclude these files, but you want one of them to back up these files. By defining a Global filter, you can enable Global filters for all Automated Critical Data Identification applications except for the application you want to back up these files.

Note: Global filters must be enabled for an application for them to take effect.

With Local application filters, the GLOBAL_xxx keywords are used without values to indicate where global filtering should occur relative to Local application filters. Typically, all Global filters are specified first. It is an application's option whether or not Global filters are processed, and if so, whether they are honored before, or after, local filters.

Note: Note:

- An Automated Critical Data Identification application name of GLOBAL is not permitted.
- Use caution when using Universal and Global filters, especially with Retention filters. Make sure that any masks you specify do not include more data than you intend.
- Global filters can be overridden by including the Global filters after a Local filter.

Local filters

Local filters apply only to a specific application and are typically processed after any Universal and Global filters.

Usage

Note the following:

- To override a Global filter, place it after a Local filter.
- Universal and Global Exclude filters are usually comprised of masks that identify groups of data sets that should *not* be included in an application's Selection Data Set. Therefore, it is recommended that you review the current Universal and Global filters when you are defining new applications to Automated Critical Data Identification.

Filter types

Within the Universal, Global, and Local filter levels, *filter types* control the contents of the Automated Critical Data Identification Application Data Inventory database and the Selection Data Set.

The type of filter determines whether data is placed in the Automated Critical Data Identification Application Data Inventory database.

Data set Retention filters

Data set Retention filters determine whether data set names are added to the Automated Critical Data Identification Application Data Inventory database from RSP JCL Capture, RSP SMF Capture, or SMF Scan.

Note: Retention filters do not apply to job collection.

Entries that match the data sets specified by the Retention filter are not added to the Automated Critical Data Identification Application Data Inventory database. This means BKMAPLEN is not required to sort through additional records to exclude data sets that you do not want an application to back up. For example, if you have SYS1 data sets that you do not want any application to back up, define them using Universal Retention filters. This will reduce the BKMAPLEN processing time and keep your Automated Critical Data Identification Application Data Inventory database at a maintainable size.

Note: Occasionally, data is missing in an SMF record. This most often occurs for the volume field. If a discard retention field is referencing a field with missing data, a match does not occur. Subsequent catalog look up processing might add the missing data to the DSR record, so it is not immediately clear why the record was not discarded. To circumvent this situation, add the discard filter information as an EXCLUDE to the evaluation filters.

Alternatively, if you want the entries to be written to the Automated Critical Data Identification Application Data Inventory database but excluded from the Automated Critical Data Identification applications, use Evaluation filters.

Using Local Retention filters can be more risky in that any data set name specified by a Local Retention filter is not added to the Automated Critical Data Identification Application Data Inventory database for that application and therefore cannot be analyzed for criticality.

Data set Evaluation filters

Data set Evaluation filters determine whether data set names in the Automated Critical Data Identification Application Data Inventory database are evaluated for criticality and whether they are included in the backup Selection Data Set.

Note: Evaluation filters do not apply to job collection.

All entries that match the data sets specified by an Evaluation filter are added to the Automated Critical Data Identification Application Data Inventory database. When BKMAPLEN or Verification performs processing, the entries are placed in the correct category based on the filter you specified.

For example, if you specify an Exclude data set statement in the Local Evaluation filter for 'PAYB.**' which belongs to the application PAYROLL, all 'PAYB.**' data sets found during the Payroll application cycle will be written to the Automated Critical Data Identification Application Data Inventory database for the Payroll application, but they will be excluded in the Selection Data Set. By using an Evaluation filter and writing them to the Automated Critical Data Identification Application Data Inventory database, you have a record of the data you excluded.

Job collection filters

Job collection filters alter the job list that constitutes an application. Job collection filters only apply to job names collected from a job scheduling product and are local to the application.

Note:

- Job name filters do not apply if job names are collected using ISPF Edit.
- Global and Universal filter options do not exist for job name filtering.

Filter categories

Filtering patterns, consisting of various *filter categories*, can be established through option **3 (FILTERS)** on the **Application AutoBuild Options** menu.

A filtering pattern is established using various keyword statements in these filter categories:

- Job Names (JOB)
- Data Set Names (DSN)
- External Data Set Names (DSN)
- Volume Names (VOL)
- Unit Names (UNIT)
- SYSID Names (SYSID)
- Data Set Organization types (DSORG)
- Program name (PGM)

Defining Universal and Global filters

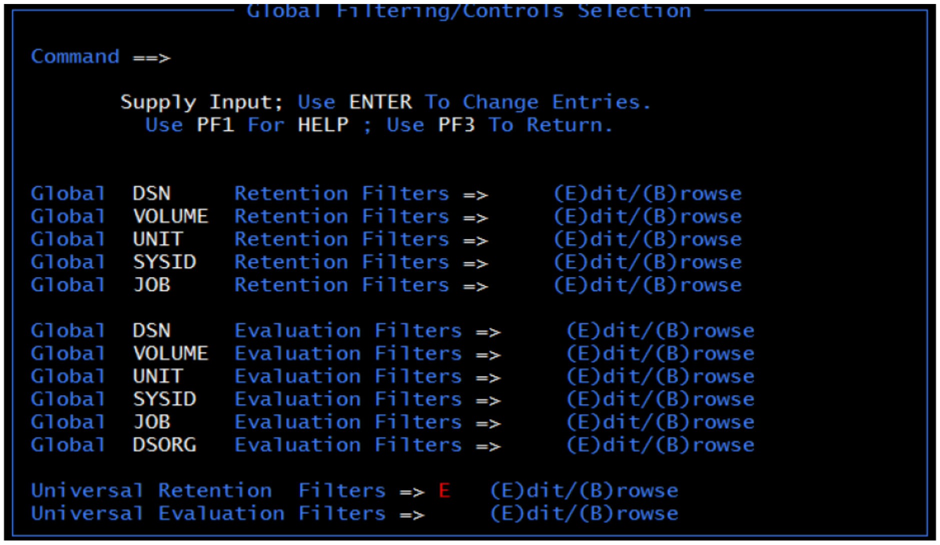
These steps describe how to define Retention and Evaluation filters at the Universal or Global level.

Defining Universal filters

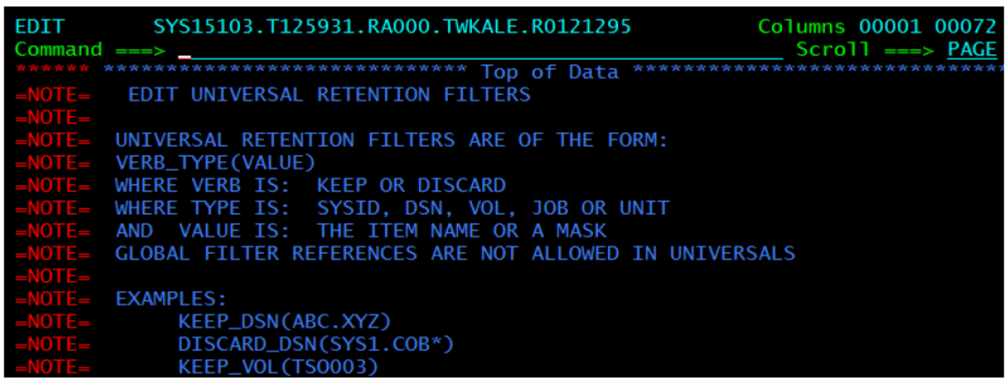
These steps describe how to define Retention and Evaluation filters at the Universal level.

1. On the Advanced Backup and Recovery for z/OS Main Menu, type 3 in the **Command** line and press Enter to select Automated Critical Data Identification.
2. On the Automated Critical Data Identification menu panel, type 5 in the **Option** line and press Enter to select **Filters/Controls**.
3. On the Global Filtering/Controls Selection panel, type E in the appropriate field and press Enter to select a universal retention filter.

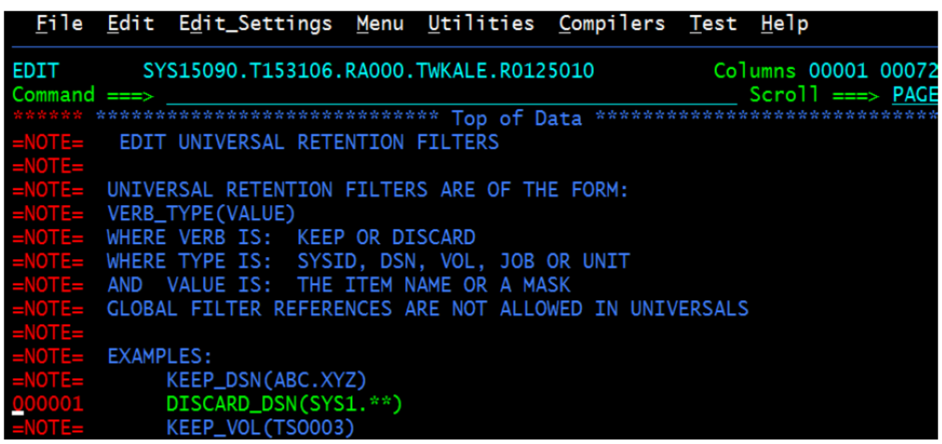
   The following figure shows slection of a universal retention filter:


Global Filtering/Controls Selection
Command ==>
Supply Input; Use ENTER To Change Entries.
Use PF1 For HELP ; Use PF3 To Return.
Global DSN Retention Filters => (E)dit/(B)rowse
Global VOLUME Retention Filters => (E)dit/(B)rowse
Global UNIT Retention Filters => (E)dit/(B)rowse
Global SYSID Retention Filters => (E)dit/(B)rowse
Global JOB Retention Filters => (E)dit/(B)rowse
Global DSN Evaluation Filters => (E)dit/(B)rowse
Global VOLUME Evaluation Filters => (E)dit/(B)rowse
Global UNIT Evaluation Filters => (E)dit/(B)rowse
Global SYSID Evaluation Filters => (E)dit/(B)rowse
Global JOB Evaluation Filters => (E)dit/(B)rowse
Global DSORG Evaluation Filters => (E)dit/(B)rowse
Universal Retention Filters => E (E)dit/(B)rowse
Universal Evaluation Filters => (E)dit/(B)rowse

4. When the Edit Filters panel appears, it contains an example filter, such as the one shown in the following figure:


EDIT SYS15103.T125931.RA000.TWKALE.R0121295 Columns 00001 00072
Command ===> Scroll ===> PAGE
****** ***************************** Top of Data ******************************
=NOTE= EDIT UNIVERSAL RETENTION FILTERS
=NOTE=
=NOTE= UNIVERSAL RETENTION FILTERS ARE OF THE FORM:
=NOTE= VERB_TYPE(VALUE)
=NOTE= WHERE VERB IS: KEEP OR DISCARD
=NOTE= WHERE TYPE IS: SYSID, DSN, VOL, JOB OR UNIT
=NOTE= AND VALUE IS: THE ITEM NAME OR A MASK
=NOTE= GLOBAL FILTER REFERENCES ARE NOT ALLOWED IN UNIVERSALS
=NOTE=
=NOTE= EXAMPLES:
=NOTE= KEEP_DSN(ABC.XYZ)
=NOTE= DISCARD_DSN(SYS1.COB*)
=NOTE= KEEP_VOL(TSO003)

You can use the ISPF MD line command to change a =NOTE= line into a data line. For this exercise, use the DISCARD_DSN keyword to exclude all SYS1.** data sets that are to be backed up by any Automated Critical Data Identification application.


File Edit Edit_Settings Menu Utilities Compilers Test Help
EDIT SYS15090.T153106.RA000.TWKALE.R0125010 Columns 00001 00072
Command ===> Scroll ===> PAGE
****** ***************************** Top of Data ******************************
=NOTE= EDIT UNIVERSAL RETENTION FILTERS
=NOTE=
=NOTE= UNIVERSAL RETENTION FILTERS ARE OF THE FORM:
=NOTE= VERB_TYPE(VALUE)
=NOTE= WHERE VERB IS: KEEP OR DISCARD
=NOTE= WHERE TYPE IS: SYSID, DSN, VOL, JOB OR UNIT
=NOTE= AND VALUE IS: THE ITEM NAME OR A MASK
=NOTE= GLOBAL FILTER REFERENCES ARE NOT ALLOWED IN UNIVERSALS
=NOTE=
=NOTE= EXAMPLES:
=NOTE= KEEP_DSN(ABC.XYZ)
000001 DISCARD_DSN(SYS1.**)
=NOTE= KEEP_VOL(TSO003)

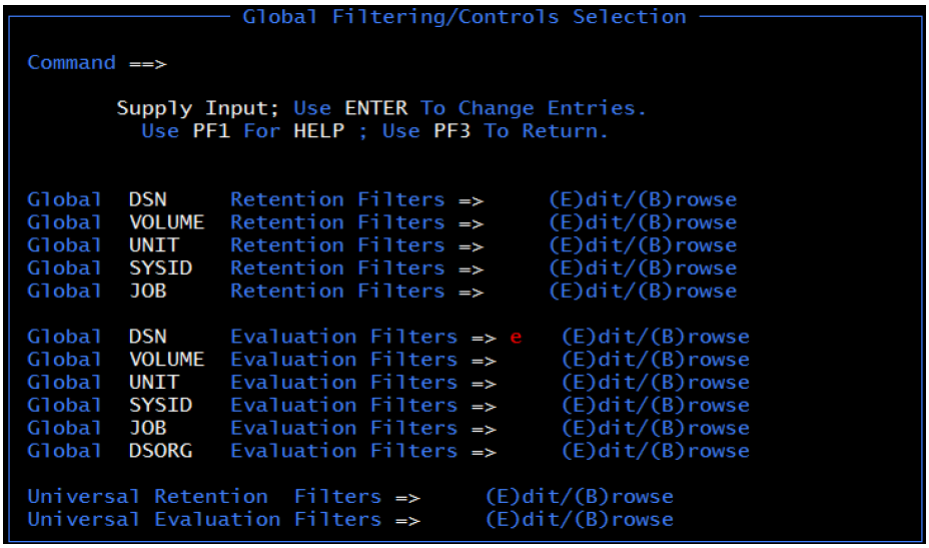
5. Press PF3 to save your filter definition and display the Global Filtering/Controls Selection panel.
6. On the Global Filtering/Controls Selection panel, type E in the **Universal Evaluation Filters** field and press Enter, then repeat steps 4 and 5 to define a universal evaluation filter.
7. Press PF3 until the Advanced Backup and Recovery for z/OS Main Menu panel appears.

Defining Global filters

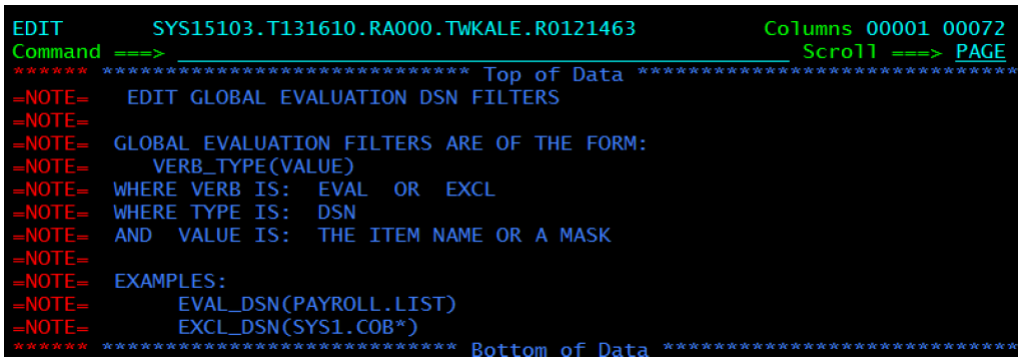
These steps describe how to define Retention and Evaluation filters at Global level.

1. On the Advanced Backup and Recovery for z/OS Main Menu, type 3 in the **Command** line and press Enter to select Automated Critical Data Identification.
2. On the Automated Critical Data Identification menu panel, type 5 in the **Option** line and press Enter to select **Filters/Controls**.
3. On the Global Filtering/Controls Selection panel, type E in the appropriate field and press Enter to select a global evaluation filter on a data set name (DSN).

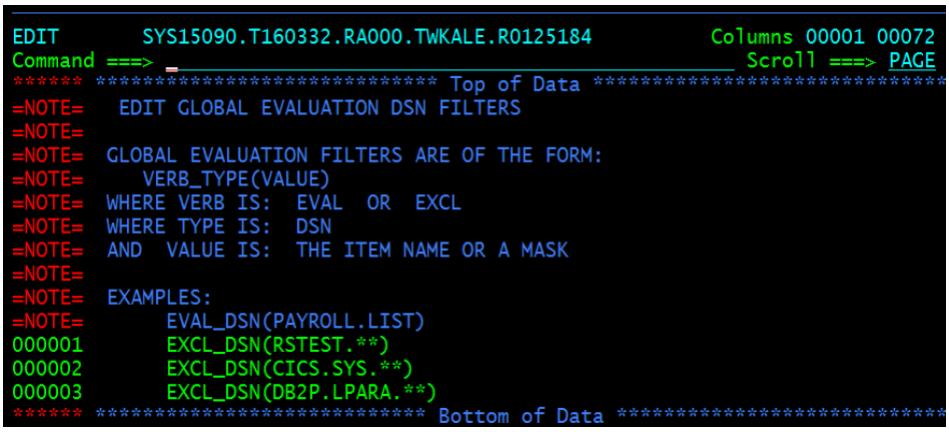
   The following figure shows slection of a global evaluation filter based on data set name:



4. When the Edit Filters panel appears, it contains an example filter, such as the one shown in the following figure:



   You can use the ISPF MD line command to change a =NOTE= line into a data line. For this exercise, use the EXCL_DSN keyword to exclude a set of files, such as software libraries, CICS system data sets, and so forth. This filter can be enabled or disabled for each individual Automated Critical Data Identification user application. The following figure shows the filter definition:



5. Press PF3 to save your filter definition and display the Global Filtering/Controls Selection panel.
6. Press PF3 again to display the Automated Critical Data Identification menu panel.

Data set Retention filter syntax

Data set Retention filters determine whether data set names are added to the Automated Critical Data Identification Application Data Inventory database from RSP JCL Capture, RSP SMF Capture, or SMF Scan. Retention filters do not apply to job collection.

Control statement syntax

{KEEP |DISCARD }_{DSN |VOL |SYSID |UNIT |JOB |PGM}(*value*)

Keywords

KEEP_(*value*)
Where *value* can be explicitly stated for the category of filter specified or an extended ACS filter mask.

DISCARD_(*value*)
Where *value* can be explicitly stated for the category of filter specified or an extended ACS filter mask.

Typically, all data set records associated with an application are included in the Automated Critical Data Identification Application Data Inventory database (KEEP is the default). Therefore, DISCARD is used almost exclusively to limit records written to the Automated Critical Data Identification Application Data Inventory database. Placing KEEP keywords before DISCARD keywords is equivalent to the expression 'DISCARD except for.'

Keywords are evaluated in the order specified. The first match terminates the filtering for a specific SMF record.

Note: For some new data sets, SMF records do not have the volume serial number and thus volume Retention filters will have no volume to match. In this situation, use volume Evaluation filters instead.

Example 1

```
GLOBAL_SYSID GLOBAL_DSN GLOBAL_VOL GLOBAL_UNIT GLOBAL_JOB DISCARD_(**.*report*.**)
```

In this example, any Global KEEPs or DISCARDs will be honored first. If the data set being filtered matches a Global KEEP or DISCARD, the test ends with the disposition of the data set determined by the matching Global filter. If no Global filters match, then a data set will be kept unless it matches the local DISCARD filter (a name with that has the text "report" in any qualifier).

Example 2

```
EVAL_DSN(**.DATA) GLOBAL_SYSID GLOBAL_DSN GLOBAL_VOL GLOBAL_UNIT GLOBAL_JOB
```

In this example, data sets with a low level qualifier of "DATA" will be evaluated whether or not a Global filter might have otherwise excluded the data sets from evaluation.

Using KEEP | DISCARD with JOB

You might or might not KEEP or DISCARD the data that you intend to by using the KEEP | DISCARD syntax with the JOB statement. For example, assume you have two jobs: JOBA and JOBB. The first reference to a data set is in JOBA, and JOBA is the job you DISCARD. You will not see a data set that was referenced in JOBA later on in JOBB because JOBA was the first reference to the data set and therefore discarded by the DISCARD_JOB(JOBA) statement.

You might also have a data set appear in the Automated Critical Data Identification Application Data Inventory database by another job if that job was the first reference. For example, the first reference to a data set is in JOBA. The data set is also referenced in JOBB, but because Automated Critical Data Identification identifies the first reference of a data set, if you DISCARD JOB(JOBB) and think the data set was excluded — it is not. The first reference was in JOBA, and therefore, written to the Automated Critical Data Identification Application Data Inventory database for JOBA.

The DISCARD option limits the number of data set records added to the Automated Critical Data Identification Application Data Inventory database. This might be desirable if unnecessary data sets are a burden on the Automated Critical Data Identification Application Data Inventory database. However, data sets that are never added cannot be evaluated for inclusion as Selection Data Set entries. Nor are they included in various reports, thus losing visibility when researching data set names that do not appear in the Automated Critical Data Identification Application Data Inventory database.

An alternative is to allow data set entries to be created in the Automated Critical Data Identification Application Data Inventory database using the KEEP option, but exclude them from evaluation using the EVAL or EXCL filters.

For new applications, you might start by using Evaluation and Exclude statements to see which data sets are excluded using various EXCL masks. When assured that EXCL masks will not encompass more data sets than intended, change the EXCL filters to DISCARD filters.

Example

```
KEEP_DSN(PAYR.**) DISCARD_VOL(SYS*)
```

In this example, any data sets from volumes matching 'SYS*' are not included in the Automated Critical Data Identification Application Data Inventory database except for data sets matching 'PAYR.**'.

```
DISCARD_JOB(JOBA)
```

In this example, any data sets found in JOBA are not included in the Automated Critical Data Identification Application Data Inventory database.

KEEP only or EVAL only examples

These examples show how to use only KEEP statements, or EVAL statements, to specify or evaluate a list of the data sets to add to the Automated Critical Data Identification Application Data Inventory database.

Using KEEP statements

```
KEEP_DSN(INV*.**)
KEEP_DSN(PAYR.**)
DISCARD_DSN(**)
```

In the example, the KEEP statements identify specific criteria to identify which data sets are to be added to the Automated Critical Data Identification Application Data Inventory database. The sequence of the KEEP statements before the DISCARD statement ensures that the desired data sets are added. All other data sets, denoted by DSN(**) on the DISCARD statement, are not added to the Automated Critical Data Identification Application Data Inventory database.

Using EVAL statements

```
EVAL_DSN(INV*.**)
EVAL_DSN(PAYR.**)
EXCL_DSN(**)
```

In the example, the EVAL statements identify specific criteria for which data sets are to be evaluated for possible inclusion in a Selection Data Set. The sequence of the EVAL statements before the EXCL statement ensures that the desired data sets are evaluated. All other data sets, denoted by 'DSN(**)' on the EXCL statement, are not evaluated for inclusion in a Selection Data Set.

EXCL statement

```
EXCL_JOB(JOBA)
```

In the example, any data sets found in JOBA are written to the Automated Critical Data Identification Application Data Inventory database but excluded from the application.

Data set Evaluation filter syntax

Data set Evaluation filters determine whether or not data set names in the Automated Critical Data Identification Application Data Inventory database are evaluated for criticality — and whether they are included in the backup Selection Data Set.

Control statement syntax

{EVAL| EXCL}_{DSN| VOL| SYSID| UNIT| JOB| PGM(*program name or mask*)| DSORG}(*value*)
{EXCL_BUT_CRITICAL}{DSN| DSN mask| PGM(*program name or mask*)}

Keywords

EVAL
Evaluate a data set for inclusion in the Selection Data Set. After data set names reside in the Automated Critical Data Identification Application Data Inventory database, they are evaluated for criticality.

EXCL
Exclude a data set from the Selection Data Set.

For example, instead of using DISCARD to discard data sets from the Automated Critical Data Identification Application Data Inventory database, you can choose to allow data set names from RSP JCL Capture, RSP SMF Capture, or SMF Scan to be included using KEEP. You can then use EXCL to exclude them from the Selection Data Set so an audit trail is available.

Note: An EXCLUDE, or an INCLUDE, VOL filter will be matched against all volumes for a data set. If a data set is a multivolume data set and a volume matches the volume specified on the VOL statement, the filter will be applied.

DSN
Specifies the data set name.

VOL
Specifies the volume.

SYSID
Specifies the system ID.

UNIT
Specifies the unit.

JOB
Specifies the job name.

PGM(*program name or mask*)
Specifies the program name.

DSORG (*value*)
Specifies the data set organization where *value* is one of the following:

- DA (direct)
- IS (ISAM)
- PO (partition)
- PS (physical sequential)
- V* (where V* indicates any VSAM data set)
- VE (VSAM ESDS)
- VK (VSAM KSDS)
- VL (VSAM LDS)
- VR (VSAM RRDS)
- VS (VSAM undetermined)

Tip: Use EVAL to force exceptions to an EXCL filter by placing an EVAL statement before an EXCL statement.

EXCL_BUT_CRITICAL
This filter is available if both Automated Critical Data Identification and Critical Backup Tracking and Inventory are licensed and you want to use the optional features available in the Application Backup and Restore feature of Critical Backup Tracking and Inventory.

When you use Automated Critical Data Identification to create a Selection Data Set to perform a post-application backup, it is typical to use EXCLUDE filters to exclude data sets from the backup because the backups are performed elsewhere (for example, when another application is backing up the data set).

However, when BKUPEND is run, it discovers and displays critical data sets that are not backed up. For the comparison to encompass all critical data sets, including those you do not want in a post-application backup, use EXCL_BUT_CRITICAL to exclude data sets from the Selection Data Set and allow them to be compared to the Critical Backup Tracking and Inventory discovery of data set backups.

Note: For existing Automated Critical Data Identification applications, change any EXCLUDE filters to EXCL_BUT_CRITICAL filters if you want to create the BKUPEND exception report or if you want to use the Critical Backup Tracking and Inventory restore by application feature. (This step is not required if you already are excluding your backup jobs from the Automated Critical Data Identification job list.)

DSN
Data set name or mask.

PGM(*discretenameORmask*)
Program name or mask.

Data set Include filters (INCL, ACCOMP, and ALLOC keywords)

Use these filters when you want to force a data set to be considered critical, or if you want to add other data sets to the Selection Data Set.

Control statement syntax

```
{INCL |ACCOMP | ALLOC}_{DSN}(value)
{INCL |ACCOMP | ALLOC}_{EXTDSN}(value)
{INCL}_{VOL |SYSID | UNIT |JOB |DSORG}(value) |PGM}(program name or mask)}
```

Filter keywords

INCL
Include filter.

ACCOMP
Accompany filter.

ALLOC
Allocate filter.

value
Where *value* can be explicitly stated for the category of filter specified or an extended ACS filter mask.

DSN, VOL, SYSID, UNIT, DSORG, and JOB categories

Using the INCL (also ACCOMP or ALLOC for DSN) keyword within these categories makes all data sets that match these filters critical. The matching data sets become an INCLUDE, ACCOMPANY, or ALLOCATE in a Selection Data Set unless they are not cataloged.

Note: An INCLUDE filter or an EXCLUDE, VOL filter are matched against all volumes for a data set. If a data set is a multivolume data set, and a volume matches the VOL statement, the filter action occurs.

Note: Use of the ACCOMP_DSN option requires permission by a BKMINI token. When using this option, ensure the BKMINI token FILTERS_OVERRIDE_ACCOMPANY is set to YES.

EXTDSN category

The EXTDSN category includes data sets that are not being evaluated for this application.

- If the data set is not in the Automated Critical Data Identification Application Data Inventory database, it is added.
- If it exists in the Automated Critical Data Identification Application Data Inventory database the data set is ignored.

Any external data set added to the Automated Critical Data Identification Application Data Inventory database is processed like the regular data sets in the Automated Critical Data Identification Application Data Inventory database. It can still be excluded if it matches an exclude filter or if it is not cataloged.

Note: The **Application AutoBuild Options** menu **4 (VERIFICATION)** options apply to INCL_EXTDSN and ACCOMP_EXTDSN. For example: INCL_EXTDSN is converted to ACCOMPANY if the data set is indeed tape and the build option **USER TAPES** is set to **AS ACCOMPANY**.

Defining Local filters

You can define filters at the local level for job collection, data set retention, and data set evaluation.

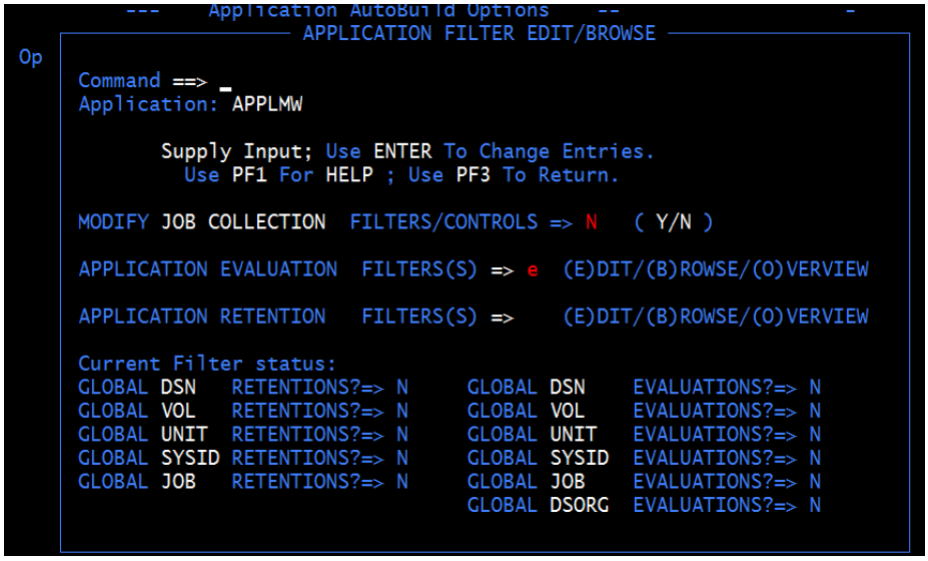
Defining Local filters

You can define Local (application level) Retention and Evaluation filters by following these steps.

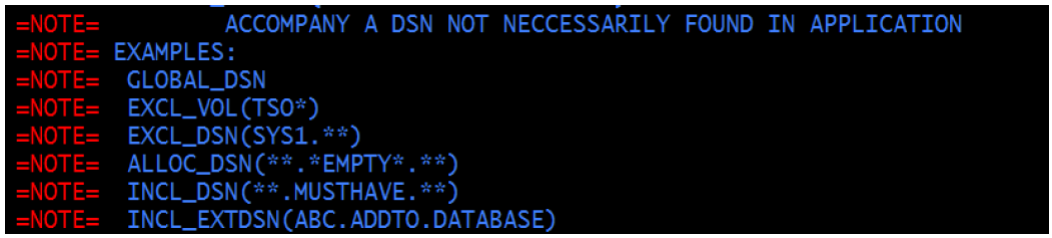
1. On the Automated Critical Data Identification menu panel, type 2 in the Command line and press Enter to select the **Select/Create Appl** option.
2. When the **Select Current/New Application Name** panel appears, type the name of your Automated Critical Data Identification application in the **Use Current/Old Application Name** field and press Enter.

   The following figure shows the entry for Automated Critical Data Identification application APPLMW:

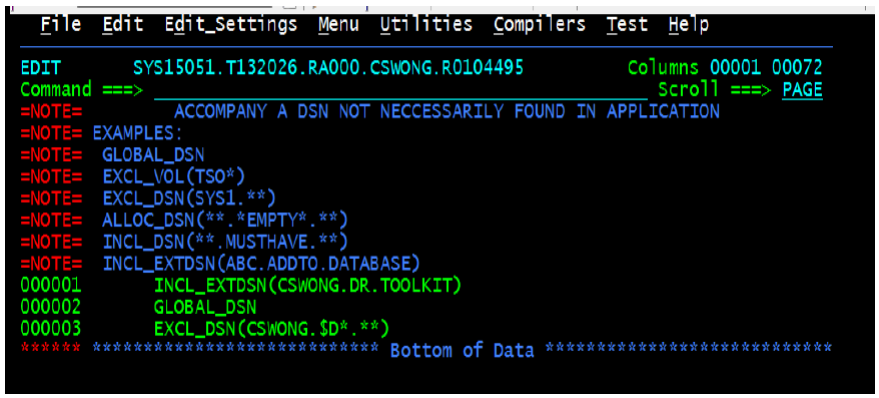
3. On the Application Autobuild Options panel type 3 in the **Option** line and press Enter to display the Application Filter Edit/Browse panel:



4. On the Application Filter Edit/Browse panel, type E in the **Application Evaluation Filter(s)** field and press Enter to select an application evaluation filter, as shown in the preceding figure. When you press Enter and EDIT session opens for a sample filter, as shown in the following figure:



5. For this example, enable the GLOBAL_DSN filter, create a local EXCL_DSN filter for the application, and use the INCL_EXTDSN keyword to manually include an external data set that is critical to the application. Use the MD line command to convert =NOTE= lines into data lines:



Defining Local filters for job collection

Job collection filters filter job names during the job collection process. Job collection filters apply only to job names collected from a job scheduling product.

To use job name filtering, you must define the filtering patterns for the application before submitting the job collection. The filtering pattern limits the scope of the jobs collected by including or excluding job entries.

You define filtering patterns using Include and Exclude entries:

- Include entries specify the job names or masks supplied to be included in the final job collection list. Include entries are processed before any Exclude entries.

- Exclude entries specify the job names or masks supplied to be *excluded* from the final job collection list. Exclude entries are processed after any Include entries.

For example, if an application uses a single anchor job name, which returns both a daily and a weekly set of job entries, you can separate the daily and weekly entries into two applications.

1. From the Automated Critical Data Identification Main Menu, select the **SELECT/CREATE APPL** option.
   The **SELECT Current/NEW Application Name** panel is displayed.
2. Select or type an application name.
   The **Application AutoBuild Options** menu is displayed.
3. Select the **FILTERS** option.
   The **APPLICATION FILTER EDIT/BROWSE** panel is displayed.
4. In the **MODIFY JOB COLLECTION FILTERS/CONTROLS** field, type Y.
   An ISPF Edit session is displayed.
5. Specify INCL_JOB, EXCL_JOB, or INCL_EXTJOB entries.
   The order of the keywords you specify is important. During filter processing, the first match terminates filter processing for a given job name. The INCL_EXTJOB filter is not affected by filter processing. INCL_EXTJOB filters can be placed anywhere in the filter list.

   Note:

   - Exclude statements take precedence and are processed after Include statements.
   - You can specify additional job name selection criteria depending on the job scheduling product interface. For example, Unicenter CA-7 Sched IDs might affect the job names that are selected from the job scheduling product data.

Job collection filter syntax

Job collection filters alter the job list that comprises an application. Job collection filters only apply to job names collected from a job scheduling product and are local to the application.

Control statement syntax

`{INCL | EXCL}_JOB(`*jobname or mask*`)`

`INCL_EXTJOB (`*jobname or mask*`)`

Keywords

{INCL | EXCL}_JOB
The INCL and EXCL statements determine which job names are added to the Automated Critical Data Identification Application Data Inventory database from the job collection. *jobname or mask* can be either explicitly stated job names or an extended ACS filter mask.

INCL_EXTJOB
Add a job name or mask directly to the Automated Critical Data Identification Application Data Inventory database, bypassing any job collection processing or filtering. *jobname or mask* can be either explicitly stated job names or an extended ACS filter mask.

Job name filtering mask guidelines

Follow these guidelines when using job name filtering masks.

- When specifying Include and Exclude filtering entries, if multiple job names or masks are supplied, they must be blank delimited. You must also use valid masking characters and patterns:

Masking character	Usage
*	Used in front or at the end of the name, as a wildcard character mask.

Masking character	Usage
<	Represents a single alpha-only character.
>	Represents a single numeric-only character.
%	Designates a single character (any type) placeholder value.
!	Designates a single national character placeholder value: @, #, $

For example:

```
JOBNAME INClude(s) => PET%%34* PEFFR* PROD76* *CSRD*
JOBNAME EXClude(s) => PEFFR%7* PROD%%45
```

In the example above, only those job names that meet any of the following criteria are included on the final job collection list:

- Job names that start with PET, have two characters preceding a '34' and can end with any character.
- Job names that start with PEFFR and end with any characters.
- Job names that start with PR0D76 and have any ending character.
- Job names that have the character string CSRD in any position.

The following jobs are excluded from the job collection list if they meet any of the following criteria:

- Job names that start with the string PEFFR, and a single character, plus a numeric 7, plus any ending character in that sequence.
- Jobs names that start with string PR0D, and have two characters following of any type and end with a numeric 45.

If you exclude a job that triggers subsequent jobs, those later jobs might also be excluded. For example, if JOB1 is the anchor and it triggers JOB2 and JOB3. JOB3 triggers JOB4, which in turn triggers JOB5. If you exclude JOB3, JOB4 and JOB5 might also be excluded. The BKMINI token JOBCOLL_EXCLUDE_DEP controls whether JOB3's exclusion also excludes its dependent jobs.

How filters and controls work for GDSs and Verification GDG rules

These rules describe how filters and controls work for generation data sets (GDS) in conjunction with Verification GDG rules.

Rules

1. Filters can be specified for absolute (G0000V00s) or relative generations (*nn*). However, a filter for an absolute GDS is not practical because it would need to be constantly modified; therefore, the following examples assume that filters for specific generations would be specified as relative names.

   For example: EXCL_DSN(SAMPLE.GDG(0)).
2. Relative generations must be specified as either 0 or -*n* with no leading zeros. Do not specify +0 or -03.

   For example:

   SAMPLE.**(0) is valid.
3. You can use a mask with a relative generation. Relative generation data set (GDS) filters can only be used as Evaluation filters because the match of absolute and relative GDS names cannot be accomplished until all the application jobs are complete. BKMAPLEN processing matches relative and absolute GDS names.
4. Relative filter data set names are matched to both relative and absolute names in the Automated Critical Data Identification Application Data Inventory database.

For example, the Automated Critical Data Identification Application Data Inventory database contains SAMPLE.GDG.G0010V00, which is generation 0; therefore, a filter of SAMPLE.GDG(0) will match SAMPLE.GDG.G0010V00.

5. A filter can override a GDG rule but cannot change the type of GDG name to be used in the Selection Data Set.

   For example, if the GDG rule states that new generations are to be excluded, a filter of INCL_DSN(SAMPLE.GDG(0)), assuming that generation zero was a new generation, forces the generation zero data set name to be included. The actual GDG name type (absolute or relative) is still determined by the GDG rules.

6. EXTernal adds of GDSs are subject to the GDG rules — they do not take precedence.

   For example, with an INCL_EXTDSN(SAMPLE.GDG(0)) control where generation 0 is a new generation and the **ABARS CATEGORY FOR NEW GENERATIONS** option is set to E (Exclude), the externally included data set is excluded because of the GDG rules in effect.

   However, non-EXT filters still apply to GDSs so the data set might still be forced to INCLUDE status with an INCL_DSN filter.

   For example: INCL_EXTDSN(SAMPLE.GDG(0)) INCL_DSN(SAMPLE.GDG(0))

   Whether an INCL_EXTDSN is needed depends on whether the desired generation is present in the Automated Critical Data Identification Application Data Inventory database (from either RSP JCL Capture or RSP SMF Capture).

   Again, the rule for EXT type filters is that they are used to add an entry to the Automated Critical Data Identification Application Data Inventory database. You might need another filter to force the desired category.

   Summary of precedence:

   - Non-EXT filters (the first filter match takes precedence)
   - GDG rules
   - EXT Include category

Note: The exception to filters overriding GDG rules is that the name type to be used in the Selection Data Set is fixed by the GDG option (absolute or relative). For example, if a filter of INCL_EXTDSN(TEST.GDG(0)) is specified but the GDG option for the type of GDS to be used in the Selection Data Set is Absolute, the Include External filter causes the absolute equivalent data set name to be placed in the Automated Critical Data Identification Application Data Inventory database.

Reviewing the results of application filtering

Review the results of filtering activities for an application using one of these methods.

- Select the **LIST APPLICATIONS** option from the Automated Critical Data Identification Main Menu.
- Select the **STATUS** option from the **Application AutoBuild Options** menu.
- Select the **VERIFICATION** option from the **Application AutoBuild Options** menu.

Filtering guidelines

Follow these guidelines when working with filters.

- Universal filters are processed first and Global filters are normally processed second (although it depends on placement within the Local filters when it was enabled). Local filters (for individual applications) are processed after any Universal filters and typically after Global filters. Filters are processed using "drop-through" logic; that is, the first encounter of a filter that matches a data set's name is the filter that is used.

- Global filters are optional and processed in a relative sequence to Local application filters. If the GLOBAL_DSN is enabled and is in the Local filter list before any Local filters, then Global filters take precedence. Exceptions to Global filters should be placed in front of GLOBAL_DSN.
- Job name filters do not apply if job names are specified manually through job collection using the ISPF Edit option. Job name filters have no impact on INCL_EXTJOB filters.
- KEEP and DISCARD filters affect which records are added to the Automated Critical Data Identification Application Data Inventory database.
- EVAL and EXCL filters affect whether a data set record that has been collected is to be evaluated for possible inclusion in the Selection Data Set.
- The INCL_DSN, ACCOMP_DSN and ALLOC_DSN keywords are used to force a data set into the Selection Data Set that would otherwise be in another category.
- EXTDSN causes an external data set to be *considered* for inclusion in the Automated Critical Data Identification Application Data Inventory database. It does not indicate automatic inclusion into the Selection Data Set. To force the data set into a Selection Data Set, you must either use an INCL_DSN, ACCOMP_DSN or ALLOC_DSN filter statement.
- Filters for generation data sets can be absolute (g0000v00) or relative(*nn*). However, a filter for an absolute GDS makes no sense because it would need to be constantly changed.
- **Defining filters when RSP is not running.** If you are using SMF Scan, or if RSP is not active, you will receive a notification that RSP is not running. This is normal. The filters are saved. However, when RSP is active, saving the filters propagates them to every image.
- **Defining filters to identify data outside mirrored volumes.** To identify data that has strayed outside mirrored volumes, place the mirrored volumes in an EXCL_VOL(volsers) filter. The data Automated Critical Data Identification identifies after filtering out the mirrored volumes are the exceptions that you need to address.
- **Including or excluding migrated data.** You can include or exclude data that is currently migrated. Use the control statement syntax: { INCL | EXCL }_VOL(MIGRA1 | MIGRA2)

Chapter 16. Getting started with Critical Backup Tracking and Inventory

This information describes how to prepare to use Critical Backup Tracking and Inventory to track backup data. Complete these steps before you begin using Critical Backup Tracking and Inventory.

1. Create batch intercept file filters.
2. Identify involved images; set up and start the intercept tasks.
3. Create test scenarios.
4. Evaluate restore scenarios.

Identifying involved images; set up and start the intercept tasks

The Critical Backup Tracking and Inventory tracking started task must be implemented on any image where backup programs might be running. Perform the installation steps necessary for the started tasks. When the tasks are started, tracking begins.

Creating test scenarios

Rather than waiting for executions of various backup utilities or types of data sets to present themselves from normal production runs, it is recommended that you determine the types of utilities you plan to track and then run some test jobs using these utilities. Consider testing backup tracking using DFDSS/FDR, IEBGENER, SORT, and so forth.

Note: In most cases, data set RESTORE operations are performed by passing the backup output file to the same program used to perform the backup. For example, the IEBGENER SYSUT2 data set created by the backup is passed to IEBGENER as input (SYSUT1) to perform the RESTORE.

Backup utilities that use a DD statement to identify the data set to be backed up require that the DD statement identify one and only one complete data set; PDS members and concatenated data sets are not supported.

It is also helpful to analyze RESTORE skeleton options to set up tests for various types of data sets.

Restore scenario considerations

Critical Backup Tracking and Inventory builds restore JCL and control statements to match the utility or command used to back up a data set or volume. Alternatively, you can use your own restore JCL.

As you consider the options for restoring, first familiarize yourself with the following programs, procedures, and parameters:

- The Build Restore (BKMBLDRJ) program, where you select the data sets or volumes to be restored. The BKMBLDRJ program drives skeleton processing, which controls the generation of the restore JCL and control statements.
- Build Restore program parameters.
- Skeleton processing procedure.

Default and sample skeletons are provided with Critical Backup Tracking and Inventory for the utilities and commands that Critical Backup Tracking and Inventory supports. Although it is possible to create a single skeleton to generate the appropriate JCL or control statements for any backup utility, to reduce the complexity of skeletons, Critical Backup Tracking and Inventory uses different skeletons based on the utility or command used for the backup.

Critical Backup Tracking and Inventory allows you to define the default skeleton for each corresponding backup utility or command. You can also create your own default or special usage skeletons.

Because time is critical for either a local restore or a disaster recovery, it is advisable to review the options to use for the Build Restore parameters and the skeletons so that restoration can be accomplished quickly.

BKMBLDRJ options and skeleton variations

This information outlines BKMBLDRJ options and skeleton variation considerations.

Dependent data sets or volumes

Use care when you are restoring data sets or volumes where the associated data sets or volumes must also be restored to place an application or group of data sets to the same point in time required for a restart. Although Critical Backup Tracking and Inventory allows selection of a single data set to be restored, it is your responsibility to know whether the associated data sets must also be restored to the same point in time.

Whether for local or disaster recovery, examine how the BKMBLDRJ restore options can be utilized to recover all data sets or volumes that must be treated as a group.

Restore, replace versus fail

An important strategic decision is how to handle collisions when restoring data sets. Some utilities automatically allocate target data sets and usually have a "replace" option to choose whether an existing like-named data set (or just a catalog entry) should be replaced.

For utilities that require allocation of the target data set, the decision whether to replace is based on how the restore JCL and control statements are coded (such as first deleting any existing data set or catalog entry to make sure a subsequent allocate works).

This is especially important in a disaster recovery plan that depends on the initial status of user catalogs. If populated catalogs are restored with no scrubbing, all data sets will have catalog entries. Conversely, if catalogs are either selectively recovered or scrubbed, the first attempt to restore a data set should, in theory, not experience a collision. Critical Backup Tracking and Inventory includes skeletons for each utility in two types: one type that assumes REPLACE or a DELETE prior to allocation and another type that assumes that existing data sets should result in an error. (The default skeleton JCL is configured not to replace data sets.)

Target data set volume location

The default skeleton restore jobs provide target volume serial and generic unit type allocations based on the location of the data sets when they were backed up. This permits both non-SMS allocation and SMS allocation controlled by the ACS routines. Critical Backup Tracking and Inventory cannot determine which, if any, esoteric unit names were used to allocate the original data sets. If these defaults are undesirable for your installation, you can code alternate skeleton jobs to use esoteric unit names, SMS class names, or volume serials compatible with your environment. To prevent possible disruption as a result of Critical Backup Tracking and Inventory maintenance installation, it is recommended to preserve the original members of the skeleton library. Therefore, you are advised to create new skeleton members in SKELLIB for those utility restore jobs to be customized. To default to the customized restore jobs, specify the new SKELLIB member names in the corresponding BKMINI member tokens in section :CBTI_RESTORE_OPTIONS. To use the customized restore jobs selectively (using the **Set Up Restore** option only), specify the new SKELLIB member names in the **SKEL** parameter of the BLDREST command.

Relative date masking

Where permitted, relative date masking enables you to specify date search criteria relative to the current date.

Relative date masks use these characters:

Character	Usage
*	Represents the current date
+	Inclusive
-	Exclusive
n	Represents the number of days (1-999)

Note: When a date field is left blank, all dates are displayed. If you specify a specific date, it must be in the format YYYY/DD/MM or MM/DD/YYYY.

You can specify date criteria using these relative date mask formats:

Mask	Meaning
*	Display events that occurred today.
*+*n*	(**Full Volume Dump Search Setup** panel and **Jobname List Search Setup** panel.) Display all generation 0 events from today through *n* days from today. All generations of each event are displayed regardless of the date. **Note:** On the **History** panel, this format displays all events equal to today through *n* relative days from today.
*-*n*	Display items *n* days from the current date.

Examples

Specify `*` to display events from today.

Specify `*-1` to display events from yesterday.

Specify `*+1` to display events from today and yesterday.

On the **Full Volume Dump Search Setup** panel, specify `*+4` to display all events, from today through four days from today, with each of their subsequent generations (regardless of the date).

Specify `*-4` to display events four days from today.

Chapter 17. Critical Backup Tracking and Inventory setup and configuration

Use the **Critical Backup Tracking Inventory** options provided on the **Setup and Configuration** panel to configure Critical Backup Tracking and Inventory.

Preparing to track backup data using Critical Backup Tracking and Inventory

This information describes how to prepare Critical Backup Tracking and Inventory to track backup data.

Use the Critical Backup Tracking and Inventory options provided on the **Setup and Configuration** panel to configure Critical Backup Tracking and Inventory.

Note: For Critical Backup Tracking and Inventory V2.3 and later, Advanced Backup and Recovery for z/OS supports the FREE=CLOSE JCL parameter for DFSMSdss (PGM=ADRDSSU, DUMP command) on the output DD statements.

This support requires that both the DAD started task (BKMSTSKD) and the Critical Backup Tracking and Inventory started task (BKMSTSKA) be active.

1. (Optional) If your site intends to use Critical Backup Tracking and Inventory to track backups performed by user-defined backup utilities, use the **User-Defined Backup Utility Management** option on the **Setup and Configuration** panel to identify these utilities to Critical Backup Tracking and Inventory.
2. Create the batch intercept file (BIF) filters.
3. Build the Batch Intercept File and refresh Critical Backup Tracking and Inventory.
4. (Optional) Set up DFSMShsm ARCINBAK backups. DFSMShsm ARCINBAK jobs are tracked by the Critical Backup Tracking and Inventory started task (if the job matches a job name mask in the batch intercept file). To track ARCINBAK backups, the `DD ARCPRINT` statement must be present.
5. Run the backups. The Critical Backup Tracking and Inventory started task will track backup information for backups that you defined.
6. After Critical Backup Tracking and Inventory has performed some initial tracking, review the results to see if your filter settings are appropriate.
7. Expire cycles from the Advanced Backup and Recovery for z/OS Inventory Data Set. You must set up and run job BKMEXPIR to remove expired backups and expired DFSMShsm AUTODUMP versions from the Advanced Backup and Recovery for z/OS Inventory Data Set.

Defining job exclusion filters for DAD processing

DAD uses job exclusion filters to identify and ignore any SMF data received from the specified job names and job masks. Use the **Setup and Configuration** panel's `DAD Jobname Management` option to add or delete job exclusion filters.

When a PDS or VSAM data set is migrated and recalled by HSM after a backup, the data set appears to have changed because the recall opens the data set for output, which updates the DAD change fields for the data set. The data set appears to have changed when in fact, it has only been recalled. You can avoid this problem by defining exclusion filters for DAD processing so that data from HSM jobs does not update the DAD.

1. Type 1 in the **Command** line on the **Main Menu** panel and press Enter to display the **Setup and Configuration** panel.
2. Type 7 in the **Command** line on the **Setup and Configuration** panel and press Enter to display the **DAD Jobname Management** panel.

3. When the **DAD Jobname Management** panel appears, use the online help to guide as you as you specify field information and use the primary and line commands.
4. When you have completed your entries on the **DAD Jobname Management** panel, press END to save your job exclusion filters and display the **Setup and Configuration** panel.

DAD Jobname Management panel field and command reference

The DAD Jobname Management panel displays a list of all job exclusion filters defined for DAD processing. The list is empty when you first install Critical Backup Tracking and Inventory.

Field descriptions

These fields appear on the DAD Jobname Management panel:

Command
Use the **Command** line to type primary commands.

C
Use this column to type line commands next to the appropriate job exclusion filter. Use the line commands to insert, delete, and repeat filters in the list.

Jobname
Use this field to specify a name for your jobname exclusion filter. The name you specify can have a maximum of 8 alphanumeric characters; your entry cannot contain blank spaces. You may also use the asterisk (*) or question mark (?) masking characters in any position except the first:

- An asterisk (*) means that any character at that position and any characters that follow it are considered a match for the filter.
- A question mark (?) means that any single character at that position is considered a match for the filter.

For example:

- Job* matches Job1, Jobname, Job2987654, and so on.
- Jobprd?10 matches Jobprd010, JobprdA10, and so on.

Update ID
(Display only) This column shows the user ID under which the job exclusion filter was last modified.

Date Updated
(Display only) This column shows the date on which the job exclusion filter was last updated.

Time Updated
(Display only) This column shows the time at which the job exclusion filter was last updated.

Date Updated
(Display only) This column shows the date on which the job exclusion filter was last updated.

Add ID
(Display only) This column shows the user ID under which the job exclusion filter was created.

Add Date
(Display only) This column shows the date on which the job exclusion filter was created.

Add Time
(Display only) This column shows the time at which the job exclusion filter was created.

Primary commands

REF or REFRESH
Type REF or REFRESH in the **Command** line and press Enter to refresh the list shown on the panel by reloading the data from Inventory Data Set (IDS).

CAN or CANCEL
Type CAN or CANCEL in the **Command** line and press Enter to quit the **User Defined Backup Utilities Management** without saving your entries or modifications.

SAVE
Type SAVE in the **Command** line and press Enter to save your entries and updates on the **User Defined Backup Utilities Management** panel to the Inventory Data Set (IDS). The new records overwrite any existing user-defined backup utility records already stored in the IDS.

END or Enter
Press the END function key or the Enter key to save your entries and updates, and then exit the **User Defined Backup Utilities Management** panel.

DATEY or DATED
Type the appropriate command in the **Command** line and press Enter to specify how you want the date information formatted.

- DATEY—Show dates in YYYY/MM/DD format.
- DATED—Show dates in DD/MM/YYYY format.

SORTA or SORTD
Type the appropriate command in the **Command** line and press Enter to sort the records in the list, based on Jobname.

- SORTA—Sort the records into ascending order.
- SORTD—Sort the records into descending order.

Line commands

Specify line commands in the **C** column next to the appropriate jobname and press Enter.

I
Insert a new blank record in the list, immediately below the selected record. You can use the SORTA or SORTD primary commands to reorder the list based on jobname.

D
Delete the record for the selected job exclusion filter. Be aware that if you delete a record from the list and then subsequently use a primary command to save the list, the saved list does not contain the deleted record.

R
Copy the selected record and insert the copy immediately below the selected record. This line command provides a convenient way to clone a new job exclusion filter based on a similar job exclusion filter.

Defining user-defined backup utilities to Critical Backup Tracking and Inventory

If your site plans to use Critical Backup Tracking and Inventory to track backups performed by user-defined utilities, you must identify these utilities to Critical Backup Tracking and Inventory. You can use the **User-Defined Backup Utilities Management** option on the **Setup and Configuration** panel for this purpose.

Use the Critical Backup Tracking and Inventory options provided on the **Setup and Configuration** panel to configure Critical Backup Tracking and Inventory.

1. From the **Main Menu** panel, type 1 in the **Command** line and press Enter to display the **Setup and Configuration** panel.
2. When the **Setup and Configuration** panel appears, type 8 in the **Command** line and press Enter to display the **User Defined Backup Utilities Management** panel.
3. When the **User Defined Backup Utilities Management** panel appears, use the online help to guide you as you specify field information and use the primary and line commands.

4. When you have completed your entries on the **User Defined Backup Utilities Management** panel, press END to save your list of utilities and display the **Setup and Configuration** panel.

User Defined Backup Utility Management panel field and command reference

The **User Defined Backup Utility Management** panel displays all user defined backup utilities known to Critical Backup Tracking and Inventory. This list is empty when you first install Advanced Backup and Recovery for z/OS.

After a user-defined backup utility has been defined to Critical Backup Tracking and Inventory from this panel, you can initiate backup tracking for it by defining it to the backup tracking INCLUDE/EXCLUDE filters. Use options 3 and 4 on the **Setup and Configuration** panel.

Field descriptions

The following fields appear on the **User Defined Backup Utility Management** panel:

Command
Use the **Command** line to type primary commands.

C
Use this column to type line commands next to the appropriate user-defined backup utility. Use the line commands to insert, delete, and repeat lines in the list.

Program Name
(Required) Use this field to specify the backup utility's program name. Your entry may contain as many as 8 alphanumeric characters (A-Z, 0–9, $, @, or #); the first character may not be numeric.

Input DD
(Required) Use this field to specify the DD name of the backup utility's input data.

Primary Output DD
(Required) Use this field to specify the DD name of the backup utility's output destination for the copied data.

Duplex Output DD
(Optional) Use this field to specify the DD name of the backup utility's second output destination for the copied data.

SYSOUT DD
(Optional) Use this field to specify the DD name of the backup utility's SYSOUT data.

User ID
(Display only) This field shows the user ID under which the utility definition was added or last updated.

Primary commands

REF or REFRESH
Type REF or REFRESH in the **Command** line and press Enter to refresh the list shown on the panel by reloading the data from Inventory Data Set (IDS).

CAN or CANCEL
Type CAN or CANCEL in the **Command** line and press Enter to quit the **User Defined Backup Utilities Management** without saving your entries or modifications.

SAVE
Type SAVE in the **Command** line and press Enter to save your entries and updates on the **User Defined Backup Utilities Management** panel to the Inventory Data Set (IDS). The new records overwrite any existing user-defined backup utility records already stored in the IDS.

END or Enter
Press the END function key or the Enter key to save your entries and updates, and then exit the **User Defined Backup Utilities Management** panel.

SORTA or SORTD
Type the appropriate command in the **Command** line and press Enter to sort the records in the list, based on Program Name.

- SORTA—Sort the records into ascending order.
- SORTD—Sort the records into descending order.

Line commands

Specify line commands in the **C** column next to the appropriate program name and press Enter.

I
Insert a new blank record in the list, immediately below the selected record. You can use the SORTA or SORTD primary commands to reorder the list based on program name.

D
Delete the record for the selected user-defined backup utility. Be aware that if you delete a record from the list and then subsequently use a primary command to save the list, the saved list does not contain the deleted record.

R
Copy the selected record and insert the copy immediately below the selected record. This line command provides a convenient way to clone a new utility definition based on a similar existing definition.

Batch intercept file (BIF) filters

Because Critical Backup Tracking and Inventory tracks only those jobs, steps, or program names identified by BIF filters, you must create these filters before using Critical Backup Tracking and Inventory.

You can create the filters later; however, you must identify at least some of the backup jobs before any results can be produced.

Before you create BIF filters, Advanced Backup and Recovery for z/OS must be installed and the Advanced Backup and Recovery for z/OS Inventory Data Set must be allocated.

Note: If you use a name other than the vendor-supplied name of supported programs, define the utility equivalents through the **Admin** menu option **Define Utility Equivalents**.

Although Critical Backup Tracking and Inventory tracks backups without JCL changes, using job name or step name masks can significantly reduce the task of defining backup jobs. Therefore, you might want to consider whether changing any of the job names might make the task easier. If changing job names is feasible, also consider whether it might facilitate the restore process. Critical Backup Tracking and Inventory offers restoration by:

- JOBNAME
- JOBGROUP
- DSN
- VOL

If data set naming conventions do not easily allow restoring a group of data sets by DSN using masks, it might be more convenient to restore by JOBNAME or JOBGROUP (even if this means changing some backup job names).

Note: Because Critical Backup Tracking and Inventory assumes that a match on job name, step name, and program name is a backup, avoid using masks that might cause utilities (for example, an IDCAMS REPRO might be used in a VSAM reorganization process) to be tracked.

Critical Backup Tracking and Inventory

Use the Critical Backup Tracking and Inventory features (options 3 through 6 on the **Backup and Recovery Management** menu panel) to track backups, manage job groups, restore data, and display detailed information about backup and recovery events.

Preparing to track backup data

Complete these steps before you begin using Critical Backup Tracking and Inventory.

Use the **Critical Backup Tracking Inventory** options provided on the **Setup and Configuration** panel to configure Critical Backup Tracking and Inventory.

1. Create the batch intercept file (BIF) filters.
2. (Optional) If you plan to use Critical Backup Tracking and Inventory to track backups performed by user-defined utilities, use the **Setup and Configuration** panel's `User-Defined Backup Utilities Management` option to define the utilities to Critical Backup Tracking and Inventory.
3. Identify involved images; setup and start the intercept tasks. The Critical Backup Tracking and Inventory tracking started task must be implemented on any image where backup programs might be running. Perform the installation steps necessary for the started tasks. When the tasks are started, tracking will begin.
4. Create test scenarios. Rather than wait for executions of various backup utilities or types of data sets to present themselves from normal production run, it is recommended that you identify the types of utilities specified in the BIF entries and then run some test jobs using these utilities. It is also helpful to analyze restore skeleton options to set up tests for various types of data sets.
5. Consider restore scenarios.

Tracking backup data

The Critical Backup Tracking and Inventory feature enables you to track critical backup data.

1. Create the batch intercept file (BIF) filters.
2. Build the Batch Intercept File and refresh Critical Backup Tracking and Inventory.
3. (Optional.) Set up DFSMShsm ARCINBAK backups. DFSMShsm ARCINBAK jobs are tracked by the Critical Backup Tracking and Inventory started task (if the job matches a job name mask in the batch intercept file). To track ARCINBAK backups, the `DD ARCPRINT` statement must be present.
4. Run the backups. The Critical Backup Tracking and Inventory started task will track backup information for backups that you defined.
5. After Critical Backup Tracking and Inventory has performed some initial tracking, review the results to see if your filter settings are appropriate.
6. Expire cycles from the Advanced Backup and Recovery for z/OS Inventory Data Set. You must set up and run job BKMEXPIRE to remove uncataloged backups from the IDS.

DFSMShsm ARCINBAK example

To track ARCINBAK backups, the `DD ARCPRINT` statement must be present as shown in this example.

```
//BKMBJ910 JOB ,'ARCINBAK',CLASS=A,MSGCLASS=X,NOTIFY=&SYSUID
//*************************************************************
//* DO HSM INLINE BACKUP OF SEVERAL DATASETS USING ARCINBAK.
//* FOR INFORMATION ABOUT ARCINBAK SEE THE
//* DFSMShsm STORAGE ADMIN. GUIDE CHAPTER 10
//*************************************************************
//ARCINBAK EXEC PGM=ARCINBAK
//ARCPRINT  DD  SYSOUT=*
//ARCSNAP   DD  SYSOUT=*
//BACK01    DD  DISP=SHR,DSN=BKMX.BKM.IVP.J910.BACK01
//BACK02    DD  DISP=SHR,DSN=BKMX.BKM.IVP.J910.BACK02
//BACK03    DD  DISP=SHR,DSN=BKMX.BKM.IVP.J910.BACK03
```

Tracking full volume backups

Select the Critical Backup Tracking and Inventory **Full Volume Dumps** option to display all full volume backups found in the Advanced Backup and Recovery for z/OS Inventory Data Set that match the search criteria you specify.

1. From the **Main Menu**, select the **Backup and Recovery Management** option.
2. From the **Backup and Recovery Management** menu, select the **Full Volume Dumps** option.
3. On the **Full Volume Dump Search Setup** panel, specify the criteria to display full volume dumps:
   a) In the **Volume or Mask** field, specify a volume name for the backup events you want to display. You can specify a maximum of six characters. Volume name masks are permitted. The default is * (all volumes).
   b) In the **Auto Expand List** field, specify whether to display previous versions of each backup:
      - Type Y to display all versions of each backup.
      - Type N to display only the current version of each backup.
   c) In the **Event Date** field, specify date criteria or a specific date. Dates must be specified in the format: YYYY/MM/DD or MM/DD/YYYY. If you do not specify a date, backups for all dates are displayed. Relative date masking is permitted.
   d) In the **RC (FVD/BKM)** field, specify whether to display full volume dump (FVD) or Advanced Backup and Recovery for z/OS (BKM) information based on return codes , or both:
      - Specify E to display all events ending with a return code of eight or greater (error messages).
      - Specify W to display all events ending with a return code of four (warning messages).
      - Specify *n* , <*n* , or >*n* (where *n* is an integer 0–999) to display events with specific return codes.

      For example:

      Type 0 to display all events with a return code of zero.

      Type <4 to display all events with a return code less than four.

      Type >4 to display all events with a return code greater than four.
   e) In the **Sort list by** field, specify how to sort the list: by volume (**V**), date (**D**), or by errors (**E**).
   f) In the **Store values to be used as default for this USERID** field, type a forward slash (/) to store the current values specified on this panel. If you select this option, the next time you access the **Full Volume Dump Search Setup** panel, the fields will contain your most recent search criteria.

      Tip: To view or change the options on the **Full Volume Dump Search Setup** panel after it has been stored, type S or type a question mark (?) in the **C** field in the list heading on the **Full Volume Dumps** panel.
   g) Type a forward slash (/) in the **Don't show this panel again** field to skip the **Full Volume Dumps** panel when you select the Full Volume Dumps option from the previous panel.
4. Press Enter.

   The **Full Volume Dumps** panel displays all events found in the Advanced Backup and Recovery for z/OS Inventory Data Set that match the search criteria you specified.

Sorting the Full Volume Dumps display

The contents of the **Full Volume Dumps** panel can be sorted by date, error date, or volume name. Choose one of these options to sort the display.

- Type one of these options in the **Sort** field:

Option	Description
D	Sort the display by date.
E	Sort the display by errors (by date).
V	Sort the display by volume name.

- Type one of these commands on the **Command** line:

Option	Description
SD	Change the sort value to **D** and sort the display by date.
SE	Change the sort value to **E** and sort the display by errors (by date).
SV	Change the sort value to **V** and sort the display by volume name.

Filtering the Full Volume Dumps display

This information describes how to filter the contents of the **Full Volume Dumps** panel. You can filter the contents of the **Full Volume Dumps** panel using any combination of available options.

- To filter the display by volume name, change the volume name, or volume name mask, of full volume dumps to displayed in the **Volume** field. You can specify a maximum of six characters.
- To display or hide backup versions that are not current, type one of these options in the **±** field:

Option	Description
+	Display all backup versions.
-	Display only the current version of each backup.

- To filter the display by the full volume dump return code, specify one of these options in the **RC** field.

Option	Description
E	Display all full volume dump events ending with return code of eight or greater (error messages).
W	Display all full volume dump events ending with return code of four (warning messages).
n **, <*n*** **, or >*n* (where *n* is an integer in the range 0-999)**	Display events with specific return codes.

  For example:

  Type `0` to display events with a return code of zero.

  Type <4 to display all events with a return code less than four.

  Type >4 to display all events with a return code greater than four.

  Note: If the return code is prefixed with an asterisk, then Critical Backup Tracking and Inventory estimated the return code based on the ending status because no return code could be located in the database.

- To filter the display by the Advanced Backup and Recovery for z/OS return code, specify one of these options in the **RCB** field.

Option	Description
E	Display all events ending with return code of eight or greater (error messages).
W	Display all events ending with return code of four (warning messages).
n **, <*n*** **, or >*n* (where *n* is an integer in the range 0-999)**	Display events with specific return codes.

  For example:

  Type `0` to display events with a return code of zero.

  Type <4 to display all events with a return code less than four.

Type >4 to display all events with a return code greater than four.

Note: If the return code is prefixed with an asterisk, then Critical Backup Tracking and Inventory estimated the return code based on the ending status because no return code could be located in the database.

- To filter the display by the date, specify date criteria or a specific date. Dates must be specified in the format: YYYY/MM/DD or MM/DD/YYYY. If you do not specify a date, all dates will be displayed. Relative date masking is permitted. Date searches are performed against the most current generation only.

  Examples of relative date masking are:

Mask	**Interpretation**
*	Today only
*+1	Today and yesterday
*-1	Yesterday only
*+*nnn*	Today and up to 999 days ago
*-*nnn*	*nnn* days ago only

Tracking backups by job name

The Critical Backup Tracking and Inventory **Jobs** option searches the Advanced Backup and Recovery for z/OS Inventory Data Set and then displays all backups (produced by tracked backup jobs) that match the search criteria you specify.

1. From the Main Menu, select the **Backup and Recovery Management** option.
2. From the **Backup and Recovery Management** menu, select the **Jobs** option.
3. On the **Jobname List Search Setup** panel, specify the criteria for the events you want to view:
   a) In the **Jobname or Mask** field, specify the job whose backup events you want to view. You can specify a maximum of eight characters. Job name masks are permitted. The default is * (all job names).
   b) In the **Auto Expand List** field, specify whether to display all versions of each backup job:
      - Y — Display all versions of each backup job.
      - N — Display only the current version of each backup job.
   c) In the **Job Date** field, specify date criteria or a specific date. Dates must be specified in the format: YYYY/MM/DD or MM/DD/YYYY. If you do not specify a date, backups for all dates will be displayed. Relative date masking is permitted.
   d) In the **RC (BKP/CBTI)** field, specify the Critical Backup Tracking and Inventory return codes or the backup job (or both) to display:
      - Specify E to display all events ending with a return code of eight or greater (error messages).
      - Specify W to display all events ending with a return code of four (warning messages).
      - Specify *n* , <*n* , or >*n* (where *n* is an integer 0–999) to display events with specific return codes.

      Examples:

      Type 0 to display only those events where the return code is 0.

      Type 4 to display only those events where the return code is 4.

      To specify a range of return codes:

      Type <4 to display all events with a return code less than four.

      Type >4 to display all events with a return code greater than four.
   e) In the **Sort list by** field, specify how to sort the list: by job (**J**), date (**D**), or by errors (**E**).

f) In the **Store values to be used as default for this USERID** field, type a forward slash (/) to store the current values specified on this panel. If selected, the next time you access the **Jobname List Search Setup** panel, the fields will contain your most recent search criteria. (To view or change this panel after it has been stored, type S or type a question mark (?) in the **C** field in the list heading on the **Jobname List Search Setup** panel.)

Tip: To view or change the options on the **Jobname List Search Setup** panel after it has been stored, type S or type a question mark (?) in the **C** field in the list heading on the **Jobname List** panel.

g) Type a forward slash (/) in the **Don't show this panel again** field to display the **Jobname List** panel when you select the **Jobs** option from the **Backup and Recovery Management** panel.

4. Press Enter.

The **Jobname List** panel displays all events found in the Advanced Backup and Recovery for z/OS Inventory Data Set that match the search criteria you specified.

Sorting the Jobname List display

You can sort he contents of the **Jobname List** panel by date, error date, or job name.

- Type one of these options in the **Sort** field:

Option	Description
D	Sort the display by date.
E	Sort the display by errors (by date).
J	Sort the display by job name.

- Type one of these commands on the **Command** line:

Option	Description
SD	Change the sort value to **D** and sort the display by date.
SE	Change the sort value to **E** and sort the display by errors (by date).
SV	Change the sort value to **J** and sort the display by job name.

Filtering the Jobname List display

This information describes how to filter the contents of the **Jobname List** panel. You can filter the contents of the **Jobname List** panel using any combination of available options.

- To filter the display by job name, specify the job name or job name mask, of the jobs to display in the **Jobname** field. You can specify a maximum of eight characters.
- To display or hide backup job versions that are not current, type one of these options in the **±** field:

Option	Description
+	Display all versions of the backup jobs.
-	Display only the current version of the backup jobs.

- To filter the display by the job return code (0-999), specify one of these options in the **RC** field.

Option	Description
E	Display all events ending with a return code of eight or greater (error messages).
W	Display all events ending with a return code of four (warning messages).
n **, <*n*** **, or >*n* (where *n* is an integer 0–999)**	Display events with specific return codes.

For example:

Type 0 to display events with a return code of zero.

Type <4 to display all events with a return code less than four.

Type >4 to display all events with a return code greater than four.

Note: If the return code is prefixed with an asterisk, then Critical Backup Tracking and Inventory estimated the return code based on the ending status because no return code could be located in the database.

- To filter the display by the Advanced Backup and Recovery for z/OS return code (0-999), specify one of these options in the **RCB** field.

Option	Description
E	Display all events ending with a return code of eight or greater (error messages).
W	Display all events ending with a return code of four (warning messages).
n **, <*n*** **, or >*n* (where *n* is an integer 0–999)**	Display events with specific return codes.

For example:

Type 0 to display events with a return code of zero.

Type <4 to display all events with a return code less than four.

Type >4 to display all events with a return code greater than four.

Note: If the return code is prefixed with an asterisk, then Critical Backup Tracking and Inventory estimated the return code based on the ending status because no return code could be located in the database.

- To filter the display by the date, specify date criteria or a specific date. Dates must be specified in the format: YYYY/MM/DD or MM/DD/YYYY. If you do not specify a date, all dates will be displayed. Relative date masking is permitted in the **Date** field. Date searches are performed against the most current generation only.

  Examples of relative date masking are:

Mask	Interpretation
*	Today only
*+1	Today and yesterday
*-1	Yesterday only
*+*nnn*	Today and up to *nnn* days ago (where *nnn* is in the range 0-999)
*-*nnn*	*nnn* days ago only (where *nnn* is in the range 0-999)

Managing job groups

Use the Critical Backup Tracking and Inventory **Job Groups** option to display a list of job groups, and to add, edit, or delete job groups.

Displaying job groups

The Critical Backup Tracking and Inventory **Job Groups** option displays all job groups found in the Advanced Backup and Recovery for z/OS Inventory Data Set that match the search criteria you specify.

1. From the **Main Menu**, select the **Backup and Recovery Management** option.

The **Backup and Recovery Management** panel is displayed.

2. Select the **Job Groups** option.

   The **Job Group List Search Setup** panel is displayed.

3. In the **Add NEW Job Group** field, specify one of these options:

Option	Description
Y	Add a new job group.
N	Do not add a new job group.

4. Specify the search criteria for the job groups you want to view:

   a) In the **Search for Existing Job Group Name or Mask** field, specify the job group name to display. You can specify a job group mask. The default is * (display all job groups).

   b) In the **Store values to be used as default for this USERID** field, type a forward slash (/) to store the values specified on this panel. If selected, the next time you access the **Jobname List Search Setup** panel, the fields will contain your most recent specifications.

   Tip: To view or change this panel after it has been stored, type S or a question mark (?) in the **C** field in the list heading on the **Jobname List Search Setup** panel.

   c) In the **Don't show this panel again** field, type a forward slash (/) to display the **Jobname List Search Setup** panel when you select option 5 from the **Backup and Recovery Management** panel.

5. Press Enter.

   The **Job Group List** panel displays all job groups found in the Advanced Backup and Recovery for z/OS Inventory Data Set that match the search criteria you specified. (If you are adding a new job group, the **Create Job Group** panel is displayed.)

Adding a new job group

You can add a new job group when you specify search criteria on the Job Group List Search Setup panel or you can add a new job group from the **Job Group List** panel.

1. From the **Job Group List** panel, type ADD on the **Command** line and press Enter.

   The **Create Job Group** panel is displayed.

2. Specify a job group name (maximum of eight characters) and press Enter.

   The **Add Jobname or Mask to Job Group** panel is displayed.

   Note: No wildcard characters or spaces are allowed in the job group name. The job group you specify must not currently exist. If you specify a name of an existing job group, the message: `JOB GROUP ALREADY EXISTS` is displayed.

3. Select from the following options:

Option	Description
D	Delete a job name or mask entry.
I	Insert a blank line.
R	Repeat a job name or entry.

4. To list all jobs not tracked by *this* job group, type `NOTBYTHIS`.
5. To list all jobs not tracked by *any* job group, type `NOJOBGRP`.
6. Press F3, or type SAVE and press Enter, to save the job names specified in the **Jobname/Mask** column for the displayed job group. Otherwise, type CAN to cancel your changes and display the **Create Job Group** panel.

Deleting a job group

This information describes how to delete a job group.

- From the **Job Group List** panel, type D next to the job group you want to delete.

Editing a job group

This information describes how to edit a selected job group to modify job names or masks.

1. From the **Job Group List** panel, type E next to the job group you want to edit.

   The **Add Jobname or Mask to Job Group** panel is displayed.
2. Select from the following options:

Option	Description
D	Delete a job name or mask entry.
I	Insert a blank line.
R	Repeat a job name or entry.

3. To list all jobs not tracked by *this* job group, type NOTBYTHIS.
4. To list all jobs not tracked by *any* job group, type NOJOBGRP.
5. Press F3, or type SAVE and press Enter, to save the job names specified in the **Jobname/Mask** column for the displayed job group. Otherwise, type CAN to cancel your changes.

Filtering the Job Group List display

This information describes how to filter the contents of the **Job Group List** panel.

- To filter the display by job group, specify the job group or job group mask, of the job groups to display in the **Job Group** field. You can specify a maximum of eight characters. The default is * (display all jobs).

Displaying jobs that are not tracked by the current job group (NOTBYTHIS)

This information describes how to use the **NOTBYTHIS** command to display jobs that are not being tracked by the current job group. The **NOTBYTHIS** command is available from the **CBTI - Add Jobname or Mask to Job Group** panel when you add or edit a job group.

1. From the **Add Jobname or Mask to Job Group** panel, type NOTBYTHIS on the **Command** line.

   Because the job list could be quite large, the **Display Jobname or Mask** panel is displayed to enable you to filter your results.
2. In the **Job Name or Mask** field, specify a job name or mask (maximum of eight characters) to filter the display of job names. The default is * (display all jobs).
3. Press Enter.

   The **Jobname List Not Tracked by This Job Group** panel displays the jobname, return code (RC) and the return code returned from the BKM process (RCB), and the date and time. From this panel, you can select jobs to be added to the job group you are editing or adding.
4. To select a job name or mask to add to the current job group:
   - Type S mask on the **Command** line to select a job name or mask to add to the current job group.
   - Type S next to the job names you want to select to add to the current job group.
5. Press F3 to add the selected job names to the current job group.

   Note: You must save the job group to update the job group list.

Displaying jobs that are not tracked by a job group (NOJOBGRP)

This information describes how to use the **NOJOBGRP** command to display jobs that are not being tracked by any job group.

- From the **Job Group List** panel, type NOJOBGRP on the **Command** line.

  The **Jobname List Not Tracked by a Job Group** panel displays the backup job name, return code (RC) of the backup utility, and the return code returned from the BKM process (RCB), and the date and time of the backup job.

Displaying the most recent execution of jobs tracked by a job group

From the **Job Group List** panel, you can list the most recent execution of the jobs being tracked by a selected job group.

- Type S in the **C** field next to the job group you want to view.

  The **Jobnames by Job Group** panel displays the most recent execution of the jobs being tracked by the selected job group.

  Tip: Use the **REFRESH** line command to refresh the display.

Sorting the Jobnames by Job Group display

The contents of the Critical Backup Tracking and Inventory **Jobnames by Job Group** panel can be sorted by date, error date, or job name.

- Type one of these options in the **Sort** field:

Option	Description
D	Sort the display by date.
E	Sort the display by errors (by date).
J	Sort the display by job name.

- Type one of these commands on the **Command** line:

Option	Description
SD	Change the sort value to **D** and sort the display by date.
SE	Change the sort value to **E** and sort the display by errors (by date).
SJ	Change the sort value to **J** and sort the display by job name.

Filtering the Jobnames by Job Group display

This information describes how to filter the contents of the **Jobnames by Job Group** panel. You can filter the contents of the display using any combination of available options.

- To filter the display by job name, specify the job name or job name mask, of the jobs to display in the **Jobname** field. You can specify a maximum of eight characters. The default is * (display all jobs).
- To display or hide previous backup versions, type one of these options in the **+-** field:

Option	Description
+	Display previous versions of the backup for the events.
-	Display only the latest version of the backup for the events.

- To filter the display by the job return code (0-999), specify one of these options in the **RC** field.

Option	Description
E	Display all events ending with a return code of eight or greater (error messages).
W	Display all events ending with a return code of four (warning messages).
n **, <*n*** **, or >*n* (where *n* is an integer 0–999)**	Display events with specific return codes.

  For example:

  Type `0` to display events with a return code of zero.

  Type <4 to display all events with a return code less than four.

  Type >4 to display all events with a return code greater than four.

Note: If the return code is prefixed with an asterisk, then Critical Backup Tracking and Inventory estimated the return code based on the ending status because no return code could be located in the database.

- To filter the display by the Advanced Backup and Recovery for z/OS return code (0-999), specify one of these options in the **RCB** field.

Option	Description
E	Display all events ending with a return code of eight or greater (error messages).
W	Display all events ending with a return code of four (warning messages).
n **, <*n*** **, or >*n* (where *n* is an integer 0–999)**	Display events with specific return codes.

  For example:

  Type `0` to display events with a return code of zero.

  Type <4 to display all events with a return code less than four.

  Type >4 to display all events with a return code greater than four.

  Note: If the return code is prefixed with an asterisk, then Critical Backup Tracking and Inventory estimated the return code based on the ending status because no return code could be located in the database.

- To filter the display by the date, specify date criteria or a specific date. Dates must be specified in the format: YYYY/MM/DD or MM/DD/YYYY. If you do not specify a date, all dates will be displayed. Relative date masking is permitted in the **Date** field. Date searches are performed against the most current generation only.

  Examples of relative date masking are:

Mask	Interpretation
*	Today only
*+1	Today and yesterday
*-1	Yesterday only
*+*nnn*	Today and up to 999 days ago
*-*nnn*	*nnn* days ago only

Displaying job groups currently tracking a job

From the **Full Volume Dumps** or **Jobname List** panel, you can list any job groups that are currently tracking a specific job.

- Type `GL` in the **C** field next to the job you want to view.

  The **Job Group List** panel displays the job groups currently tracking the selected job. If no job groups are found the message `NO JOB GROUPS FOUND` is displayed.

Displaying application data stored during BKUPEND processing

The Critical Backup Tracking and Inventory **Application** option displays all BKUPEND processing results found in the Advanced Backup and Recovery for z/OS Inventory Data Set that match the search criteria you specify.

1. From the **Main Menu**, select the **Backup and Recovery Management** option.

   The **Backup and Recovery Management** panel is displayed.
2. Select the **Applications** option.

The **Application List Search Setup** panel is displayed.

3. In the **Application Name or Mask** field, specify the application name (maximum of eight characters) to display. Application name masks are permitted. The default is * (all application names).
4. In the **Auto Expand List** field, specify whether to display previous versions of each BKUPEND:
   - Type Y to display previous versions of each BKUPEND.
   - Type N to display only the latest version of each BKUPEND.
5. In the **Store values to be used as default for this USERID** field, type a forward slash (/) to store the values specified on this panel. If selected, the next time you access the **Application List Search Setup** panel, the fields will contain the values you last specified.
6. In the **Don't show this panel again** field, type a forward slash (/) to display the **Application List** panel when you select option 6 from the **Backup and Recovery Management** panel.
7. Press Enter.

   The **Application List** panel displays all applications found in the Advanced Backup and Recovery for z/OS Inventory Data Set that match the search criteria you specified.

Filtering the Application List display

This information describes how to filter the contents of the **Application List** panel. You can filter the contents of the **Application List** panel using any combination of options.

- To filter the display by application name, specify the application name or application name mask, of the applications to display in the **APPL** field. You can specify a maximum of eight characters.
- To display or hide previous BKUPEND versions, type one of these options in the **±** field:

Option	Description
+	Display all versions of the BKUPEND events.
-	Display only the current version of the BKUPEND event.

Deleting an application BKUPEND entry

This information describes how to delete a BKUPEND entry for an application.

- From the **Application List** panel, type D next to the entry you want to delete.

Displaying a list of data sets that have been backed up

You can view a list of data sets that have been backed up from the **Full Volume Dumps** panel or the **Jobname List** panel.

1. Type N in the **C** field next to entry you want to view.

   The **View Data Set Name List** panel is displayed.
2. When you are finished viewing the list, press F3.

Viewing backup detail

You can view backup detail associated with a particular job from the **Full Volume Dumps** panel, the **Jobname List** panel, and the **Search for Dataset** panel using the **S** line command.

1. Type S in the **C** field next to the entry you want to view.

   The **View Backup Detail** panel is displayed.
2. When you are finished viewing the list, press F3.

Generating restore JCL

You can generate restore JCL from the **Full Volume Dumps**, **Jobname List**, **Job Group List**, **Application List**, or the **View Dataset Name List** panel.

You can generate restore JCL as follows:

- From the **Full Volume Dumps** panel, you can generate restore JCL at the full volume level.
- From the **Jobname List** panel, you can generate restore JCL at the job level.
- From the **Job Group List** panel, you can generate restore JCL at the job group level.
- From the **Application List** panel, you can generate restore JCL at the application cycle level.
- From the **View Dataset Name List** panel, you can generate restore JCL at the data set/file level.

Note: Critical Backup Tracking and Inventory supports the tracking of tape-to-tape backups, but cannot build the restore JCL automatically.

1. Type R in the **C** field next to the item you want to restore, and then press Enter.

   Note: If you accessed the **View Dataset Name List** panel from the **Application List** panel, you can select data sets/Unix files for restore using the primary commands R mask (select for restore) and X mask (exclude from restore).
2. If prompted, press F3 to process the selected item for restore.
3. Type S to select an option for the restore:
   - If you are generating restore JCL for a volume, job, job group, or data set, select one of these options:
     - **Quick Restore** — This option automatically generates the resolved skeleton JCL using predetermined defaults (you do not need to specify any options other than to replace an existing data set or Unix file, if one exists, or to bypass the replace).

       Quick Restore Panel for Datasets

```
                            Select Restore Option
    Menu  Diagnostics  Preferences

  BKM5R02                     CBTI - Quick RESTORE

  (S) SELECT AN OPTION:
     Build/View JCL

  Copy                     ==> 1       1-99
  REPLACE Existing DSNs    ==> N       Y or N
  SMS GDG Restore Status   ==> A       A -ACTIVE, D -DEFERRED, N -NONE,
  (DFSMSdss & FDR Only)                S -SOURCE (DFSMSdss Only), R -ROLLEDOFF
                                       I -INPUT (FDR Only)
  RENAME OPTIONS:
  New HLQ                  ==>
    Or
  FULLRENAMEMASKS          ==> N         Y or N   LIST===> NONE SELECTED



  Command ===>
   F1=HELP     F2=SPLIT    F3=END      F4=RETURN   F5=RFIND    F6=RCHANGE
   F7=UP       F8=DOWN     F9=SWAP    F10=LEFT    F11=RIGHT
```

       Quick Restore Panel for Unix files

```
                            Select Restore Option
    Menu  Diagnostics  Preferences

  BKM5R0Z             CBTI - Quick RESTORE - Unix files

  (S) SELECT AN OPTION:
     Build/View JCL

  Copy                     ==> 1       1-99
  REPLACE Existing Files   ==> N       Y or N



  Command ===>
   F1=HELP     F2=SPLIT    F3=END      F4=RETURN   F5=RFIND    F6=RCHANGE
   F7=UP       F8=DOWN     F9=SWAP    F10=LEFT
F11=RIGHT
```

  - **Set Up Restore** — This option creates BLDREST statements using options you specify on the **Set Up RESTORE** panel and then stores them in a file.
- If you are generating restore JCL for an application, select one of these options:
  - **Point-in-time** — (Default.) Use the backup found during BKUPEND processing at backup-end to restore data sets in the application.
  - **USEGEN0ALL** — Use the latest backup available for the restore regardless of the backups found during BKUPEND processing.
  - **USEGEN0NOBKUP** — Use the latest backup available for the restore of the data sets in the application where no backup existed during BKUPEND processing.

4. Type S to view the JCL generated based on the options you specified.

Skeleton processing options

This information describes the skeleton processing options on the **Set Up RESTORE** panel.

Purpose

If you selected the **Set Up Restore** option, specify the **SKELETON PROCESSING** options on the **Set Up RESTORE** panel.

Parameters

Build/View JCL
View the JCL created based on the parameters specified on the **Set Up RESTORE** panel.

Skeleton Library
The default value is the value assigned to the Critical Backup Tracking and Inventory BKMINI token SKELLIB. Use the default value or change it to a different SKELLIB data set.

JCLOUT Dataset
Specify a PDS or PDSE data set name to store the resolved skeleton JCL.

Alloc JCLOUT Old or New
Valid options are:

- O — Identify the specified JCLOUT data set with a disposition of OLD.
- N — Identify the specified JCLOUT data set with a disposition of NEW using the allocation parameters specified using the BKMINI tokens JCLOUT_ALLOC1, JCLOUT_ALLOC2 and JCLOUT_ALLOC3.

Copy
Specify the copy number (1-99) of a multiple copy backup to be used for the restore.

Include Location
Specify one or more fully-qualified tape management location names, or one or more location masks that can be used for restore. Masked names must conform to the extended masking rules.

Exclude Location
Specify one or more fully-qualified tape management location names that *cannot* be selected.

Note:

- When used without the **Include Location** option, any location except those that match the Exclude Location can be selected.
- When used with **Include Location** option, any location that matches the Include Location and does not match the Exclude Location can be selected.

Bad Backup Return Code
Specify the return code (*nnnn*) of the backup jobs Critical Backup Tracking and Inventory should consider a unsatisfactory backup. Any backup job with a return code equal to or higher than the specified return code will be considered unsatisfactory. Valid values are 0-9999.

New HLQ
The new high level qualifier of the data sets being restored. The high level qualifier will either be replaced, or appended to, the current data set name depending upon the skeleton definition.

MEMBEROUT name/prefix
Specify the names of the JCL members that will be generated. Specify a discrete PDS or PDSE member name, or a mask to be used in generating output member names.

A member name mask requires at least one question mark (?) as a placeholder. If the **Increment Member name** field is set to Y, a member name will be generated by substituting an incrementing number in place of the question mark (?) placeholders. Question mark placeholders must occupy the rightmost positions of a member name. For example, ABCD????? is valid and ABC?D? is not valid.

Note: A mask specification is required if the **Increment Member name** field is set to Y.

JOBNAMEOUT name/prefix
Specify a discrete job name, or a mask to be used in generating output job names. A job name mask requires at least one question mark (?) as a placeholder. If multiple job names are required, job names will be generated substituting an incrementing number in place of the specified question mark placeholders. The placeholders must occupy the rightmost positions of a job name. For example, ABCD????? is valid and ABC?D? is not valid.

Steplimit per Job
Specify the maximum number of steps per job (1–255) to be generated in the output job streams.

Increment Member name
Valid values are:

- Y — Force the generation of a new output PDS or PDSE member for the next BLDREST encountered. The member name is generated using the mask specified for **MEMBEROUT** field and by incrementing the last member name used by one.
- N — The name specified in the **MEMBEROUT** field will not be incremented.

Increment Jobname
Valid values are:

- Y — Force the generation of a JOB statement for the next BLDREST encountered. The job name is generated using the mask specified for the **JOBNAMEOUT** field and incrementing the last job name used by one.
- N — The name specified in the **JOBNAMEOUT** field will not be incremented.

Include Physical Dumps
Specifies physical dumps are to be considered when looking for the most recent backups to be included in the restore. Valid values are:

- Y — Include physical dumps in the restore.
- N — Do not include physical dumps in the restore.

REPLACE Existing DSNs
Valid values are:

- Y — Delete the existing (cataloged) data set and replace it with the data set to recover.
- N — Data sets to recover that already exist (cataloged) will not be replaced.

No Backup Found RC
Specify the desired return code (0, 4, 8) if a matching data set has no backup.

No Backup Found Status
Valid values are:

- B — Build the JCL if a matching data set that was included has no backup.
- N — Do not build the JCL if a matching data set that was included has no backup.

Build Restore

The Build Restore program, BLDREST, builds restore JCL and restore utility control statements from one or more specified backups, previously tracked by the Critical Backup Tracking and Inventory started task and recorded in the Advanced Backup and Recovery for z/OS Inventory Data Set.

Restore JCL or control statements can be created by selecting from Critical Backup Tracking and Inventory the backups you want to restore. Backups are selected by one of the five categories allowed by the Build Restore (, BKMBLDRJ,) command:

- JOBNAME — Backups are tracked within Critical Backup Tracking and Inventory by the job, step, or utility that created the backup, with alternate keys created to locate any backup by data set name.
- JOBGROUP — You can optionally create job groups to associate a group of backup jobs.
- DSN — For every data set backup tracked by the Critical Backup Tracking and Inventory started task, data set attributes are collected at the time of the backup and saved in the Advanced Backup and Recovery for z/OS Inventory Data Set. Attributes are retained to allow queries and to generate restore JCL or control statements. Each attribute is assigned a variable name for subsequent reference in a restore skeleton.
- VOL — Full volume backups are tracked by volume serial number. It is recommended that full volume dumps be performed in restore operations separate from any other kind of backups.
- APPL — The Automated Critical Data Identification application cycle is added to the Advanced Backup and Recovery for z/OS Inventory Data Set by running BKUPEND.

BLDREST creates the restore JCL and control statements using generic *skeletons*. Skeletons contain model JCL and control statements, with variables that specify values specific to a given backup to be restored. The Build Restore process creates tailored output files that contain the restore JCL. The restore JCL is created by substituting the data set attributes (such as blksize) stored in the Inventory Data Set for the backups in place of skeleton variables.

BLDREST checks the catalog before building the restore JCL to ensure that the backup is on the same volume as it was at the time of backup. If it was moved or stacked, the new information is used when building the restore JCL.

You submit the BLDREST program using the restore procedure, BKMBPROC, located in the Advanced Backup and Recovery for z/OS JCL library.

The JCL or control statement requirements for restoring a backup vary, depending on the backup utility used. A common need is to swap the input and output data set names. Some utilities track the original allocation attributes, and allocate target data sets on a restoration. Other utilities require the restore to include the allocation of the target data set. Non-SMS managed data sets might require that the restore be made to different volumes than the original volume locations.

Rules

- Multiple commands in a backup step — Each backup command results in one output step. For example, a backup performed using IDCAMS containing three REPRO commands results in three steps in the output JCL member, each containing one REPRO command to accomplish the restore.
- Partial multivolume data sets from full volume dumps — If you are restoring from full volume dumps, partial restores of multivolume data sets could result. It is your responsibility to determine if all parts of a multivolume data set are present, to determine the sequence, re-catalog them and build PATHs if necessary.
- Non existent generations— If during restore, a data set generation is specifically requested and the generation does not exist, the job will issue the message `Requested Generation Not Found.`
- Duplicate data sets — During restore, Critical Backup Tracking and Inventory checks for duplicate data sets within a command and restores only the latest version. This check prevents restoring the same data set multiple times.
- Multiple restores from the same input tape — If your restore job has multiple steps using the same volume as input, VOL=RETAIN keeps it mounted until the job ends.

- Disaster recovery — If you use Critical Backup Tracking and Inventory, the Critical Backup Tracking and Inventory started task is required to be running at the disaster recovery site.

  The Critical Backup Tracking and Inventory started task provides common services for Critical Backup Tracking and Inventory backups as well restores. If you want to stop the tracking of backups, but allow the build of the restores, issue the following command on each image where the started task BKMSTSKA is active:F *xxx*,TURNOFF

  Note: Where *xxx* is the name of the CBTI started task.

BLDREST

BLDREST builds the restore JCL. Use the BLDREST parameters to select the backups you want to restore.

Required parameters

- When running BLDREST, at least one JOBNAMEOUT and at least one MEMBEROUT parameter must be included before the first BLDREST keyword.
- Use JOBNAMEOUT to specify the names of the restore jobs to run.
- Use MEMBEROUT to specify the names of the job JCL members that will be generated. STEPLIMITOUT, INCRJOBNAMEOUT, INCRMEMBEROUT are optional.

Attention: Backup job names should *not* match restore job names. Non-DFSMSdss and FDR type utilities such as IDCAMS REPRO could be tracked and considered a backup when in fact it is a restore. The backup generation results stored in the Advanced Backup and Recovery for z/OS Inventory Data Set will be incorrect.

One category must be specified: DSN, JOBNAME, JOBGROUP, VOL or APPL. Any valid wildcard characters are permitted for those parameters that support masks.

APPL(***applname*** **)**
Specify within parentheses one or more fully-qualified application names, or one or more application name masks identifying data sets to be restored.

DSN(***dsn*** **|** ***mask*** **)**
Specify within parentheses one or more fully-qualified data set names, or one or more data set name masks identifying data sets to be restored.

JOBNAME(***jobnames*** **|** ***mask*** **)**
Specify within parentheses one or more discrete backup job names, or one or more job name masks identifying job names to be restored. If the job name includes physical dumps and the data sets are multivolume, Critical Backup Tracking and Inventory restores the data sets on the selected volumes, but it does not combine them.

JOBGROUP(***jobgroup*** **)**
Specify within parentheses the name of a job group to be restored. Restore JCL will be built for all backups created by all steps and commands within all jobs defined in the job group.

VOL(***volume*** **|** ***mask*** **)**
Specify within parentheses one or more discrete volumes, or one or more masks matching volumes to be restored.

Note: For physical full volume and physical data set dumps, it is recommended that full volume restores be performed separately from any other kind of restores.

Optional parameters

BADBKUPRC(**8|** ***nn*** **)**
Specify the return code of the backup jobs Critical Backup Tracking and Inventory is to consider a bad backup. Any backup job with a return code equal to or higher than that specified RC is considered to be bad. *nn* is an integer 0 or greater. The default is eight.

CMDNO(*|*nnn*)
Valid with JOBNAME only. Specify the command within a backup utility to be restored. If multiple backup jobs are selected, CMDNO applies to all jobs. Example: If an IDCAMS backup contains three REPRO commands, CMDNO(2) generates restore JCL for only the second REPRO. The default is *.

COPY(*nn*)
Specify the copy number of a multiple copy backup to be used for the restore. *nn* is an integer of 1 or greater.

CYCLE(0|-*nn*)
Valid with APPL only. Specify the relative cycle within parentheses of the application or applications to be restored. *nn* must be 0 or -*nn*. For example, -1 specifies the next to the last backup: BLDREST APPL(PAYROLL) CYCLE(-1)

EXCLAPPL(*appl* |*mask*)
Valid with APPL only. Specify exclusions from the restore list generated by means of the APPL parameter. Specify within parentheses one or more fully-qualified application names, or one or more application name masks to identify applications to be excluded from the generated restore list. For example:

```
INCLAPPL(PAY**)

EXCLAPPL(*OFF)
```

This example causes all PAY* applications to be included in the restore except any ending with OFF.

EXCLDSN(*dsn* |*mask*)
EXCLDSN can only be specified when either DSN or INCLDSN parameters are specified. Specify exclusions from the restore list generated by means of either the DSN or INCLDSN parameters. Specify within parentheses one or more fully-qualified data set names or one or more data set name masks to identify data sets to be excluded from the generated restore list. For example:

```
INCLDSN(PAYROLL.**)

EXCLDSN(PAYROLL.*.LOGFILE.**)
```

This example causes all PAYROLL.** data sets to be included in the restore except any with LOGFILE appearing anywhere in nodes 2 through *nn*.

EXCLJOB(*jobname* |*mask*)
Valid with JOBNAME, JOBGROUP, and APPL. Not valid with DSN, VOL. Specify job names to be excluded from those specified in the JOBNAME or JOBGROUP parameter. Specify within parentheses one or more discrete job names, or one or more job name masks to identify job names to be excluded from the generated restore list.

EXCLVOL(*volser* |*mask*)
Valid with VOL only. Specify one or more volsers to be excluded from the restore list generated from VOL. Specify within parentheses one or more discrete volumes, or one or more volume masks to identify volumes to be excluded from the generated restore list.

FULLRENAMEMASKS(*old_mask1* ,*new_mask1*[*old_maskn*,*new_maskn*])]
Required for FULLRENAME. Specify pairs of old and new name masks to rename data sets as you set up a restore operation in batch or ISPF BLDREST command processing. Use blanks or commas to separate the masks.

The rules for *old_mask* are identical to those for using the INCLUDE parameter in the DFSMSdss DUMP and RESTORE control statements. Filter rules for *old_mask* are:

- A single asterisk (*) indicates any single qualifier or any group of characters within a qualifier.
- A double asterisk (**) indicates zero or more qualifiers. Your mask specification may contain no more than one double asterisk.
- A percent sign (%) indicates any single character.

Allowable filter characters for *old_mask* are:

- A single asterisk represents exactly one DSN qualifier of any value. For example: (*.) or (.*.) or (.*).

- When an asterisk is combined with valid DSN characters or the percent sign (%), the asterisk indicates zero to nn characters of any value.
- A double asterisk (**) represents zero to nn DSN qualifiers of any value. For example: (**.) or (.**.) or (.**) .
- You can use the double asterisk only once in the mask. For example, **.abcd.** is not a valid mask.
- A percent sign (%) represents a single non-blank character.
- An exclamation point (!) represents a single national character, such as @, #, $.
- A less-than sign (<) represents a single nonnumeric character (national symbols included).
- A greater-than sign (>) represents a single numeric character.

For example, the mask **.PAYROLL*.%%%%%23*.*.DATA would match a data set named TLQ050.PAYROLL.CYCLE23.YEAREND.DATA.

Filter rules for *new_mask* are:

- A single asterisk (*) indicates any single qualifier or any group of characters within a qualifier.
- You cannot use a single asterisk to represent a partial qualifier in *new_mask*. For example, aa.*bb.** is not a valid specification for *new_mask*.
- A double asterisk (**) represents zero to *nn* DSN qualifiers of any value.
- You can use the double asterisk only once in the mask. For example, **.abcd.** is not a valid mask.
- A percent sign (%) represents a single non-blank character.
- A plus sign (+) followed by one to eight characters means "insert this new qualifier."
- A minus sign (-) followed by one to eight characters means "remove this qualifier from the new name."

Note: When you use FULLRENAMEMASKS with an ISPF RESTORE to tailor a restore or recover operation, specify Y in the **FULLRENAMEMASKS** field and press Enter. On the panel that is displayed, you can specify as many as 15 *old_mask*, *new_mask* pairs. When you complete your specifications, press PF3 to re-display the **Set Up RESTORE** panel. The **LIST** field shows the number of mask pairs you specified on the panel and the **FULLRENAMEMASKS** field displays N, which indicates that the masks have been properly specified.

GEN(0|*-nn*)
Not valid with APPL. Specify the relative generation of the backup to be restored, if more than one backup generation is available. *nn* must be 0 or *-nn* For example, -1 specifies the next to the last backup. The default is 0 (the most recent backup).

When used with DSN, GEN applies to all data sets to be restored within the specified BLDREST DSN command; therefore, use multiple BLDREST DSN commands to restore an application where selected data sets must be restored from a different generation.

For example:

```
BLDREST DSN(PAYROLL.**)

EXCLDSN(PAYROLL.MASTER)

BLDREST DSN(PAYROLL.MASTER) GEN(-1)
```

INCLAPPL(*appl* |*mask*)
Valid with APPL only. Specify inclusions for the restore list generated by means of the APPL parameter. Specify within parentheses one or more fully-qualified application names, or one or more application name masks to identify applications to be included in the generated restore list. For example:

```
INCLAPPL(PAY**)

EXCLAPPL(*OFF)
```

This example causes all PAY* applications to be included in the restore except any ending with OFF.

INCLDSN(*dsn |mask*)
Not valid with DSN. Specify within parentheses one or more fully-qualified data set names, or one or more data set name masks matching data sets to be restored. INCLDSN limits the data sets to be restored to ONLY those within the specified jobs (for JOBNAME and JOBGROUP category) or volumes (for VOL category) that are specified in the command with the INCLDSN parameter.

INCLLOC(*location-code|mask*)
Specify within parentheses either of the following:

- One or more discrete Tape Management System location names
- One or more location name masks

The tape locations specified is used to determine the candidate backup copies to be used for restore.

You can use EXCLLOC(*location-code | mask*) to specify Tape Management System location names that you want excluded. The tape locations specified are used to exclude backup copies to be used for restore.

INCLPHYDUMP(YES|NO)
Required for the DSN, JOBNAME, and JOBGROUP categories. Not valid with VOL. INCLPHYDUMP specifies whether matches in the specified category (DSN, JOBNAME, or JOBGROUP) to physical dumps are to be included in the restore. When used with JOBNAME, JOBGROUP, or APPL and N is specified, if the backup chosen is a physical dump, an error message is issued and no restore JCL is produced. When N is used with DSN, physical dumps are ignored when looking for the desired backup generation.

You can also set a default by modifying the BKMINI PARMLIB member token INCLUDE_PHYSICAL_DUMPS.

INCLPHYDUMP is used only when looking for the GEN(0) backup when USEGEN0NOBKUP or USEGEN0ALL are used. Specifies physical dumps are to be considered when looking for the most recent backups to be included in the restore. The default is NO.

NEWHLQ(*hlq*)
1 to 8 byte node that will be used to build a new name for restored data sets. This value will be used as a high level qualifier preceding the rest of the original data set name.

NOBKUPFOUND (BUILD |NOBUILD,RC(*nn*|8))
Specifies whether the BLDREST command should be built and the desired return code produced if a matching data set that was included has no backup.

NOWAIT
Use this keyword to perform the restore job without ensuring all prior backups have been recorded in the IDS. Use this option only if you are certain the related backups have been recorded in the IDS.

REPLACE(Y|N)

- N — Data sets to recover that already exist (cataloged) are not replaced.
- Y — Delete the existing (cataloged) data set and REPLACE it with the data set being recovered.

SKEL(*skelname*)
Specify a skeleton library member name to be used in place of the default skeletons. If specified and the backups to be included involve multiple utility types, the skeleton must be capable of generating restore JCL for any anticipated utility types.

STEPNAME(*|*stepname*)
STEPNAME and STEPNO are mutually exclusive and are valid with JOBNAME only. Specify the step name of the backup to be restored. The default is * (restore all steps). If multiple backup jobs are selected, step name applies to all jobs. If multiple occurrences of step name are found in a job, all matching step names are restored.

STEPNO(*|*nnn*)
STEPNAME and STEPNO are mutually exclusive and are valid with JOBNAME only. Specify the JES step number of the backup to be restored. The default is * (restore all steps). If multiple backup jobs are selected, STEPNO applies to all jobs.

TGTGDS(*option*)
This option determines how the TGTGDS parameter is specified during JCL generation and specifies in what status, during a data set operation, to place restored SMS-managed GDG data sets. This parameter is for DFSMSdss and FDR only, and it is ignored for any other backup utility. The following are valid option specifications for this parameter:

- ACTIVE—Specifies that the target data set is to be assigned the ACTIVE status (for example, rolled into the GDG base).
- DEFERRED—Specifies that the target data set is to be assigned the DEFERRED status).
- INPUT—This option is valid for FDR backups only, specifying the target data set is set to the original SMS status of the GDG generation as recorded on the backup.
- NONE—Specifies that the TGTGDS parameter is not added to the RESTORE control cards.
- ROLLEDOFF—Specifies that the target data set is to be assigned the rolled-off status.
- SOURCE—This option is valid for DFSMSdss backups only, specifying that the target data set is assigned the same status as that of the source data set.

USEGEN0ALL
USEGEN0ALL and USEGEN0NOBKUP are mutually exclusive and are valid with APPL only. Always use the latest backup, even if a backup was found at BKUPEND.

USEGEN0NOBKUP
USEGEN0ALL and USEGEN0NOBKUP are mutually exclusive and are valid with APPL only. Use the latest backup for selected data sets if no backup was found at BKUPEND.

Valid parameters per category

The following table shows the parameters that are valid for each BLDREST category.

DSN	JOBNAME	JOBGROUP	VOL	APPL
BADBKUPRC	BADBKUPRC	BADBKUPRC	BADBKUPRC	BADBKUPRC
COPY	CMDNO	COPY	COPY	COPY
EXCLDSN	COPY	EXCLDSN	EXCLDSN	EXCLAPPL
FULLRENAMEMASKS	EXCLDSN	EXCLJOB	EXCLVOL	EXCLDSN
GEN	EXCLJOB	FULLRENAMEMASKS	FULLRENAMEMASKS	EXCLJOB
INCLPHYDUMP	FULLRENAMEMASKS	GEN	GEN	CYCLE
NEWHLQ	GEN	INCLPHYDUMP	INCLDSN	INCLDSN
NOBKUPFOUND	INCLDSN	NEWHLQ	NEWHLQ	INCLPHYDUMP
REPLACE	INCLPHYDUMP	NOBKUPFOUND	NOBKUPFOUND	NEWHLQ
SKEL	NEWHLQ	REPLACE	REPLACE	NOBKUPFOUND
TGTGDS	NOBKUPFOUND	SKEL	SKEL	REPLACE
	REPLACE	TGTGDS	TGTGDS	SKEL
	SKEL			USEGEN0ALL
	STEPNAME			USEGEN0NOBACKUP
	STEPNO			TGTGDS
	TGTGDS			

Building the RESTORE JCL

The Build Restore program and commands build the restore JCL for the utilities that backed up the data. The Build Restore commands control the maximum number of steps per job you want generated, the restore job names to generate, whether you want to restore by DSN, VOLUME, JOB, JOBGROUP, or APPL, and so on.

You run the Build Restore program, BKMBLDRJ, using the BKMBPROC procedure, found in the product JCL library.

1. Modify the BKMBPROC job statement to conform to your installation requirements.

2. Change HLQ to specify the high level qualifier.
3. Define a JCLOUT statement. JCLOUT specifies the library where Build Restore stores the generated restore JCL.

   When you submit the BLDREST command, the Build Restore program uses skeletons that contain model JCL and control statements to build the restore JCL for the restore utilities.

   Note:

   - The default skeleton JCL for each utility can be found in the product SKELLIB library. The skeleton JCL should not be modified because it might be written over when maintenance is applied or during the next install.
   - Skeleton JCL can also be specified in the BLDREST command by specifying the SKEL command.
   - You can create your own skeleton defaults using different names.

Example 1. A Build Restore by JOBNAME was selected using job BKMBJA10. Instead of restoring all of the data sets backed up within JOBNAME(BKMBJA10), the INCLDSN statement is used to select a specific data set for restore. The restore JCL will be built for the data set using the skeleton JCL found in the Advanced Backup and Recovery for z/OS SKELLIB for the utility that backed it up. The restore JCL will be created in the PDS data set specified by JCLOUT='BKM.JCLOUT' and the member the restore JCL will be stored in is specified by MEMBEROUT BKMREST. The job name specified for the restore job is RESTDSN.

For disaster recovery preparation purposes, you can set up this step on site and then submit it to perform the restores when you arrive at the disaster recovery site.

```
 //BKMRPROC PROC SOUT='*',
 //       LOAD1='?hlq.SBKMLOAD',
 //       PARMLIB='?hlq.SBKMPARM',
 //       SKELLIB='?hlq.SBKMSENU',
 //       JCLOUT='?hlq.JCLOUT',
 //       IDS='?hlq.IDS'
 //BUILDRJ  EXEC
PGM=BKMBLDRJ,REGION=0M,TIME=1440
   JOBNAMEOUT     RESTDSN
   MEMBEROUT      BKMREST
   STEPLIMITOUT   255
   BLDREST        JOBNAME(BKMBJA10)
                  INCLDSN(
                       BKMI.BKM.IVPAST1.NOBKP.DATASET
                       )
                  GEN(0)
                  INCLPHYDUMP(NO)
                  REPLACE(NO)
                  COPY(1 )
                  SKEL(DSSLOG)
```

Example 2. Running the Build Restore program and overriding the JCLOUT data set, and the default skeleton library member for the DFSMSdss utility using the SKEL command of BLDREST.

```
 //BKMRPROC PROC SOUT='SYSOUT-CLASS',
 //       LOAD1='?hlq.SBKMLOAD',
 //       PARMLIB='?hlq.SBKMPARM',
 //       SKELLIB='?hlq.SBKMSENU',
 //       JCLOUT='?hlq.JCLOUT',
 //       IDS='?hlq.IDS'
 //BUILDRJ  EXEC
PGM=BKMBLDRJ,REGION=0M,TIME=1440
   JOBNAMEOUT     RESTDSN
   MEMBEROUT      BKMREST
   STEPLIMITOUT   255
   BLDREST        JOBNAME(BKMBJA10)
                  INCLDSN(
                       BKMI.BKM.IVPAST1.NOBKP.DATASET
                       )
                  GEN(0)
                  INCLPHYDUMP(NO)
                  REPLACE(NO)
                  FULLRENAMEMASKS(                        +
                      EDP9.LAB**,EDP9.LAB+NOBACKUP
                      EDP9.TEST*.AIX*, EDP9.+LAB.+NOBACKUP.**)
```

```
                COPY(1 )                    -
```

Submitting the RESTORE JCL

This information describes how to submit the restore JCL.

- To submit the restore JCL, edit the PDS library you specified in JCLOUT, find the member you specified in MEMBEROUT, and submit the job to restore the data.

The data set being restored was backed up by ABARS. Therefore, the ABARS skeleton JCL was used to create the restore syntax. If the data set had been backed up by DFSMSdss, the DFSMSdss skeleton JCL would have been used and DFSMSdss restore JCL would have been created instead. The same occurs for all utilities.

```
//BKMB1  EXEC PGM=BKMBMAIN,PARM='BKMFUNC',REGION=0M,TIME=1440
//STEPLIB  DD   DISP=SHR,DSN=?hlq.SBKMLOAD
//INI      DD   DISP=SHR,DSN=?hlq.SBKMPARM(BKMINI)
//BKMIDS   DD   DISP=SHR,DSN=BKM.INVDB
//SYSTSPRT DD   SYSOUT=*
//SYSPRINT DD   SYSOUT=*
//SYSOUT   DD   SYSOUT=*
//SYSABEND DD   SYSOUT=*
//SNAP     DD   SYSOUT=*
//SYSIN DD *
 ARECOVER AG(IVPAST1)              -
          VEREXE                      -
          GEN(0)                    -
          COPYSET(1)                      -
          DATASETCONFLICT(BYPASS)    -
          TERMMSGS(YES)             -
          NOSTACK                 -
          AGGENQ(Y)             -
          XMIT                -
  INCLUDE(                              -
     BKMI.BKM.IVPAST1.NOBKP.DATASET     -
       )
```

JOBNAMEOUT command

Specify a discrete job name or a mask to be used in generating output job names.

Purpose

(Required) Specifies a discrete job name or a mask to be used in generating output job names.

JOBNAMEOUT{*jobname* | *jobnamemask* }

Parameters

jobname

A job name. The maximum length is eight characters.

jobnamemask

A job name mask. A job name mask requires at least one question mark as a placeholder. If multiple job names are required automatically, job names will be generated substituting an incrementing a number in place of the specified question mark placeholders. These placeholders must occupy only the rightmost positions of a job name.

For example:

- Valid: ABCD????
- Invalid: ABC?D?

A job name mask is not required; however, if multiple jobs are required and the job name is discrete, they will all have the same job name.

At least one **JOBNAMEOUT** command must be specified before the first BLDREST command. Multiple **JOBNAMEOUT** commands can be specified to identify a subsequent discrete job name or to identify a new mask to be used. There is no default value for this command.

Generating unique as opposed to duplicate job names is one method to control how many restore jobs can be launched concurrently, and conversely, controlling jobs that are to be run sequentially (MVS initiator rule: no two jobs of the same name will be initiated concurrently).

INCRJOBNAMEOUT command

INCRJOBNAMEOUT forces the generation of a JOB statement for the next BLDREST command encountered.

Purpose

(Optional.) **INCRJOBNAMEOUT** forces the generation of a JOB statement for the next BLDREST command encountered. The job name is generated using the mask specified for JOBNAMEOUT and incrementing the last job name used by one. **INCRJOBNAMEOUT** requires the previous **JOBNAMEOUT** was specified with a mask. There is no default value for this command.

```
INCRJOBNAMEOUT
```

Parameters

This command does not have any parameters.

MEMBEROUT command

MEMBEROUT specifies a discrete PDS member name, or a mask to be used in generating output member names.

Purpose

(Required) Specify a discrete PDS member name, or a mask to be used in generating output member names.

```
MEMBEROUT{membername |membernamemask}
```

Parameters

membername
A member name.

membernamemask
A member name mask. A member name mask requires at least one question mark as a placeholder. If a subsequent **INCRMEMBEROUT** command is encountered, a member name will be generated by substituting an incrementing number in place of the specified question mark placeholders. These placeholders must occupy only the rightmost positions of a member name.

For example:

- Valid: ABCD????
- Invalid: ABC?D?

A mask specification is required to subsequently use an **INCRMEMBEROUT** command to switch output to a new member and automatically generate a member name.

At least one **MEMBEROUT** command must be specified before the first BLDREST command. Multiple **MEMBEROUT** commands may be specified to force subsequent output to a different member. There is no default value for this command.

INCRMEMBEROUT command

INCRMEMBEROUT forces the generation of a new output PDS member for the next BLDREST command encountered.

Purpose

(Optional) **INCRMEMBEROUT** forces the generation of a new output PDS member for the next BLDREST command encountered. The member name is generated using the mask specified for **MEMBEROUT** and incrementing the last member name used by one. **INCRMEMBEROUT** requires the previous **MEMBEROUT** was specified with a mask.

```
INCRMEMBEROUT
```

Parameters

This command has no parameters. There is no default value for this command.

STEPLIMITOUT command

STEPLIMITOUT specifies the maximum number of steps per job to be generated in the output job streams.

Purpose

(Optional) **STEPLIMITOUT** specifies the maximum number of steps per job to be generated in the output job streams.

```
STEPLIMITOUTnnn
```

Parameters

nnn
: The maximum number of steps per job to be generated in the output job streams (maximum of 999). The default is 255.

Minimally required keywords

This information describes the minimally required keywords for BLDREST.

Specifying M1 for **MEMBEROUT** causes all restore JCL to be generated into one output JCL member named M1.

Because **STEPLIMITOUT** is not coded, jobs are populated with up to the maximum of 255 steps (the default).

If multiple jobs are required, all jobs will be named RESTORE1, the discretely specified job name for **JOBNAMEOUT**.

```
JOBNAMEOUT RESTORE1
MEMBEROUT M1
    BLDREST ...
    BLDREST ...
```

Note: If multiple jobs are generated, they will initiate sequentially, due to the job names being the same.

Generating different job names to allow concurrent initiation

This information provides examples of how to generate different job names to allow concurrent initiation.

Two methods of generating different job names are available.

- Discretely specified job names: In this example, RESTORE1 and RESTORE2 are the discrete job names specified in the **JOBNAMEOUT** commands.

  ```
  MEMBEROUT M1
  JOBNAMEOUT RESTORE1
      BLDREST ...
  ```

```
    BLDREST ...
JOBNAMEOUT RESTORE2
    BLDREST ...
```

- Automatic incrementing of job name: Specifying RSTOR??? for **JOBNAMEOUT** generates job names of RSTOR001, RSTOR002, and so on.

```
MEMBEROUT M1
JOBNAMEOUT RSTOR???
    BLDREST ...
    BLDREST ...
INCRJOBNAMEOUT
    BLDREST ...
```

Generating multiple output JCL members

This information provides examples of how to generate multiple output JCL members.

- Discretely specified member names: In this example, M1 and M2 are the discrete member names specified for the **MEMBEROUT** commands.

```
MEMBEROUT M1
JOBNAMEOUT RESTORE1
    BLDREST ...
    BLDREST ...
MEMBEROUT M2
    BLDREST ...
```

- Automatic incrementing of member name: Specifying M??? for **MEMBEROUT** and then including **INCRMEMBEROUT** generates member names of M001, M002,, and so on.

```
MEMBEROUT M???
JOBNAMEOUT RSTOR???
    BLDREST ...
    BLDREST ...
INCRMEMBEROUT
INCRJOBNAMEOUT
    BLDREST ...
```

Build Restore scenario 1

STEPLIMITOUT is set to 3 to illustrate how the restore JCL might require more jobs and steps than the original backups being restored.

Database job, step, and commands

```
Job  A1
  Step - IDCAMS
     4 REPROs
  Step - DSS
     3 Logical DUMPs
Job  A2
  Step - IEBGENER
  Step - IEBCOPY
     2 COPYs
```

Restore control statements

```
MEMBEROUT MEM?????
STEPLIMITOUT  3
JOBNAMEOUT REST1???
   BLDREST  JOBNAME(A*)
```

Generated output JCL

Member MEM00001:

```
//REST1001  JOB  ...
  (from 1st IDCAMS Skel)
//S1   EXEC  PGM=IDCAMS
```

```
 (1st REPRO)
//S2   EXEC  PGM=IDCAMS
  (2nd REPRO)
//S3   EXEC  PGM=IDCAMS
  (3rd REPRO)

//REST1001  JOB ...
  (from 4th IDCAMS Skel)
//S1  EXEC  PGM=IDCAMS
  (4TH REPRO)
//S2  EXEC  PGM=ADRDSSU
  ( 1st RESTORE)
//S3  EXEC  PGM=ADRDSSU
  ( 2nd RESTORE)

//REST1001  JOB ...
  (from 3rd DSS Skel)
//S1  EXEC  PGM=ADRDSSU
  (3rd RESTORE)
//S2  EXEC  PGM=IEBGENER
  ( 1st Gener)
//S3  EXEC  PGM=IEBCOPY
  ( 1st COPY)

//REST1001  JOB ...
  (from 2nd IEBCOPY Skel)
//S1  EXEC  PGM=IEBCOPY
  ( 2nd COPY)
```

Build Restore scenario 2

This information provides an example of a Build Restore scenario.

Database job, step, and commands

```
Job  A1
  Step - IDCAMS
     4 REPROs
  Step - DSS
     3 Logical DUMPs
Job  A2
  Step - IEBGENER
  Step - IEBCOPY
     2 COPYs
```

Restore control statements

```
STEPLIMITOUT  3
MEMBEROUT  MEMJA1
JOBNAMEOUT REST1???
   BLDREST  JOBNAME(A1)
MEMBEROUT  MEMJA2
INCRJOBNAMEOUT
   BLDREST  JOBNAME(A2)
```

Generated output JCL

Member MEMJA1:

```
//REST1001  JOB  ...
  (from 1st IDCAMS Skel)
//S1   EXEC  PGM=IDCAMS
 (1st REPRO)
//S2   EXEC  PGM=IDCAMS
  (2nd REPRO)
//S3   EXEC  PGM=IDCAMS
  (3rd REPRO)

//REST1001  JOB ...
  (from 4th IDCAMS Skel)
//S1  EXEC  PGM=IDCAMS
  (4TH REPRO)
//S2  EXEC  PGM=ADRDSSU
  ( 1st RESTORE)
//S3  EXEC  PGM=ADRDSSU
```

```
  ( 2nd RESTORE)

//REST1001  JOB ...
  (from 3rd DSS Skel)
//S1  EXEC  PGM=ADRDSSU
  (3rd RESTORE)
```

Member MEMJA2:

```
//REST1002   JOB
//S1  EXEC  PGM=IEBGENER
  ( 1st Gener)
//S2  EXEC  PGM=IEBCOPY
  ( 1st COPY)
//S3  EXEC  PGM=IEBCOPY
  ( 2nd COPY)
```

Build Restore scenario 3

This information provides an example of a Build Restore scenario.

Database job, step, and commands

```
Job  A1
  Step - IDCAMS
     4 REPROs
  Step - DSS
     3 Logical DUMPs
Job  A2
  Step - IEBGENER
  Step - IEBCOPY
     2 COPYs
```

Restore control statements

```
STEPLIMITOUT  3
MEMBEROUT  MEMJA1
JOBNAMEOUT REST1???
   BLDREST  JOBNAME(A1)
MEMBEROUT  MEMJA2
INCRJOBNAMEOUT
   BLDREST  JOBNAME(A2)
```

Generated output JCL

Member MEMJA1:

```
//REST1001  JOB  ...
  (from 1st IDCAMS Skel)
//S1   EXEC  PGM=IDCAMS
 (1st REPRO)
//S2   EXEC  PGM=IDCAMS
  (2nd REPRO)
//S3   EXEC  PGM=IDCAMS
  (3rd REPRO)

//REST1001  JOB ...
  (from 4th IDCAMS Skel)
//S1  EXEC  PGM=IDCAMS
  (4TH REPRO)
//S2  EXEC  PGM=ADRDSSU
  ( 1st RESTORE)
//S3  EXEC  PGM=ADRDSSU
  ( 2nd  RESTORE)

//REST1001  JOB ...
  (from 3rd DSS Skel)
//S1  EXEC  PGM=ADRDSSU
  (3rd RESTORE)
```

Member MEMJA2:

```
//REST1002   JOB
//S1  EXEC  PGM=IEBGENER
  ( 1st Gener)
//S2  EXEC  PGM=IEBCOPY
  ( 1st COPY)
//S3  EXEC  PGM=IEBCOPY
  ( 2nd COPY)
```

Application Backup and Restore

Advanced Backup and Recovery for z/OS enables you to compare Automated Critical Data Identification critical data sets to Critical Backup Tracking and Inventory tracked backups and restore data sets for Automated Critical Data Identification applications (and specific cycles within an application).

Comparison of Automated Critical Data Identification critical data sets to Critical Backup Tracking and Inventory recorded backups

Automated Critical Data Identification discovers data sets used by applications as they run and records them in the Automated Critical Data Identification Application Data Inventory database. Automated Critical Data Identification includes various controls to set rules for data set criticality. The resulting list of critical data sets can be used as input to a post-application backup.

In many applications, it might be appropriate to embed backups in the execution of the application in addition to, or instead of, performing backups after the application batch processing completes. Manually placed backups can create a maintenance problem because you might add a new data set to an application but forget to add the corresponding backup.

A backup might also be placed originally such that it occurs after all modifications, only to become invalid if the application process is altered but the position of the backup is not changed. By using the comparison feature, these data sets can be identified and compared to what was backed up. Critical Backup Tracking and Inventory also ensures that the recovery JCL or control statements match the utility used for the backup.

When Critical Backup Tracking and Inventory is used in conjunction with Automated Critical Data Identification, at the point it is known that application backups should be complete for a given cycle, Critical Backup Tracking and Inventory can search for backups corresponding to Automated Critical Data Identification identified critical data sets.

This process performs the following:

- Identify critical data sets for which there is no backup.
- Identify critical data sets for which there is a backup and display the output file name and tape location.
- Identify backups that might be invalid because the data set has been modified since the last backup.
- Prove that critical data is backed up.
- Match the backups to off-site data.

After application backups are properly in place, the process can then be used to alert appropriate personnel if an application change has occurred that invalidates the intended backup strategy.

Restore by application

Restoring by application is often the preferred way to restore data sets in the event of a disaster. Even for the loss of a single data set, it is often necessary to restore all data sets used by the application to cover the situation where multiple data sets must be in sync with each other. It might also be necessary to revert the application to something other than the last cycle that was executed.

Critical Backup Tracking and Inventory, when used in conjunction with Automated Critical Data Identification, saves the critical data set list in Critical Backup Tracking and Inventory for as many application cycles as you specify, with the backup information appropriate for the cycle. This process permits the restore of the data sets involved with a particular application cycle.

When Automated Critical Data Identification is used in conjunction with Critical Backup Tracking and Inventory to track and allow restoration by application, the sequence of events is as follows:

1. The application runs with the Automated Critical Data Identification started task, RSP, tracking the referenced data sets.
2. The application completes and the BKMAPLEN program signals the end of the application cycle.

   Unless overridden by options that you specified, BKMAPLEN performs the following tasks:

   a. Increments the current cycle number.

   b. Builds database records for the just-completed cycle (while applying all specified filters).

   c. Analyzes data set criticality. This analysis is used to create a Selection Data Set that can be used either to drive a post-application cycle backup or by BKUPEND to verify that all critical data sets were backed up in-stream.
3. BKUPEND copies the list of Automated Critical Data Identification critical data sets to the Advanced Backup and Recovery for z/OS Inventory Data Set while adding the backup information. This process provides point-in-time recovery by application.

   An exception report identifies all data sets for which no backup could be found, and if you are using the optional Changed Data Set Tracking feature, the report indicates that although a backup is found, it might be out of date because the data set has changed since the most recent backup was taken. To use this feature, set the **CHANGED_DATASET_TRACKING** token in the product parameter library's BKMINI member to Y and the BKMSTSKD started task must be running on all LPARs where the data sets might be changed, and the application SETUP option **CHECK-CHANGED-SINCE-LAST-BKUP** must be set to Y.

   For example, if BKUPEND is run at a time when the application data should be backed up but it is discovered that a data set change has been made, the backup is no longer current. The application might be running 24 hours a day, 7 days a week, or it could signal that the BKUPEND is in the wrong place. You can choose return codes to trigger an alert if a backup integrity problem is detected.

Automated Critical Data Identification options

These Automated Critical Data Identification options are available to use with the Critical Backup Tracking and Inventory ability to track and restore by application.

Backup Cycles to Retain

The **Backup Cycles to Retain** option (found on the **Application SETUP Options** panel) defines how many application cycles will be tracked by Critical Backup Tracking and Inventory in the Advanced Backup and Recovery for z/OS Inventory Data Set.

Whereas the Automated Critical Data Identification option (as defined by: **Number of RSP cycles to retain** and **Number of JCL cycles to retain**, found in **0 (SETUP)**) allows *nn* cycles to be tracked, Critical Backup Tracking and Inventory tracks cycles including the backup information found for each critical data set discovered by Automated Critical Data Identification, hence allowing restoration of the data sets specific to a given application cycle. The cycle numbers do not need to be the same.

Frequently, the Automated Critical Data Identification cycle is only required until the next application cycle runs, whereas the Critical Backup Tracking and Inventory **Backup Cycles to Retain** can be used for as many application backup cycles as desired.

Note: The Automated Critical Data Identification option to track more than one cycle only affects how many cycles' worth of raw input (SMF and JCL information) are saved in the Automated Critical Data Identification Application Data Inventory database. These records are called Data Set Detail records (DSD).

If in the Automated Critical Data Identification **Application AutoBuild Options** menu, **0 (SETUP)**, the cycles (**Number of RSP cycles to retain** and **Number of JCL cycles to retain**), are set higher than one (in general, 1 is used and is not the number of application backup cycles tracked by Critical Backup Tracking and Inventory), the DSD records for the various cycles are combined, and analyzed, and the result is the

creation of Data Set Records (DSR). These records are built for the last cycle completed, and reflect the results of **0 (SETUP)**, and filters. They are used to generate a Selection Data Set (list of data sets). At any time, only one version of the DSR records exist.

EXCL_BUT_CRITICAL Filter

When Automated Critical Data Identification is used to create a Selection Data Set in order to drive a post-application backup, it is typical to use EXCLUDE filters to exclude data sets from the backup because the backups are accomplished elsewhere, such as when they are embedded in the application or another application is backing up the data set.

However, when the Critical Backup Tracking and Inventory program BKUPEND is run, one of the program's valuable features is to discover and display any critical data sets that are not backed up. For this comparison to encompass all critical data sets, including those you do not want in a post-application backup, it will be necessary to use EXCL_BUT_CRITICAL to exclude data sets from the Selection Data Set, yet allow them to be compared to the Critical Backup Tracking and Inventory discovery of data set backups, or if you want to use the Critical Backup Tracking and Inventory Application Backup and Restore feature.

Note: If you have existing Automated Critical Data Identification applications, change any EXCLUDE filters to EXCL_BUT_CRITICAL filters if the data sets are backed up somewhere else and the BKUPEND exception report is desired, or if you want to use the Critical Backup Tracking and Inventory restore by application feature. This step is not required if you already solved the problem by excluding your backup jobs from the Automated Critical Data Identification job list.

Overview of the BKUPEND program

The BKMBKEND job (found in the product JCL library) copies the list of Automated Critical Data Identification critical data set names to the Inventory Data Set (IDS) while adding the backup information.

BKUPEND performs the following tasks:

- Locates and matches backups to application critical data sets discovered by Automated Critical Data Identification.
- Saves the list of critical data sets, along with the pertinent backup information, by Automated Critical Data Identification application name (APPL) in the IDS.
- Produces a report indicating whether all backups were found, a more recent backup was found, and optionally indicates whether any data set backups might be out of date because the data set changed since the last backup.
- Provides return code options you can use if you want to be notified when not all backups are found or they appear to be out of date.

The summary report identifies any data set for which no backup is found, and if the DAD option **CHANGED_DATASET_TRACKING** is set to Y, the report indicates that although a backup is found, it could be out of date because the data set has changed since the last backup.

Backups might not be found for various reasons:

- A data set backup was missed so that no backup exists for a data set
- BKUPEND was run before the backups completed
- A new step or job was added that interfered with BKUPEND
- The backup failed (in this case, fix the backup and run BKUPEND again)

BKMBKEND placement

You must run the BKMBKEND job after the Automated Critical Data Identification BKMAPLEN job or step has been run for the application and at the point in time when you know all backups associated with the just-completed application cycle should have completed.

If all backups are embedded in the application, BKMBKEND can be placed immediately after BKMAPLEN.

If some or all data sets are backed up after the application completes (that is, after BKMAPLEN), place BKMBKEND after the last of all backups is finished.

Take care not to run more than one BKMBKEND for each application cycle, but be sure to run BKMBKEND for each application cycle because this is the only opportunity for Critical Backup Tracking and Inventory to save the list of critical data sets and their associated backups.

Note: If a BKMBKEND is run more than once for the same Automated Critical Data Identification application cycle, BKMBKEND fails and issues a RC=8.

In most cases, you would run the BKMAPLEN job, which establishes the next application cycle before each BKMBKEND job.

For rerun situations where you do not want a new BKMAPLEN job executed, use the **RERUN** option for the BKMBKEND job.

BKUPEND program

The BKUPEND program provides a summary report in the SYSOUT from the BKMBKEND job to identify critical data sets for which no backup found, or for which a more recent backup was found, or those that might have changed since they were last backed up.

Run the BKUPEND program in batch using JCL member BKMBKEND, found in the Advanced Backup and Recovery for z/OS JCL library.

Required parameters

These keyword parameters control the program return code and whether the associated messages contain informational, warning, or error condition content.

BKUPEND
The program locates backups for a list of critical data sets produced by the BKMAPLEN job.

Optional parameters

APPL(*name*)
Specifies the Automated Critical Data Identification application name that you want to include in the BKUPEND report. Wildcard characters are not permitted. The application name can be specified using the APPL keyword as input to the BKMBMAIN program or the BPARM keyword in the BKMBPROC procedure (refer to the examples in this topic).

APPL-ONLY-BACKUPS(Y|N)
Specify whether application created backups only are to be recorded.

- Y—Record only those backups tracked by Critical Backup Tracking and Inventory and created by backup jobs listed in the Automated Critical Data Identification application options BACKUP JOB LIST.
- N—Record all backups tracked by Critical Backup Tracking and Inventory .

BAD-BKUP-RC(*nnnn*)
Specify a return code (0-9999). If the return code from any current backup found is greater than or equal to the return code specified, a warning message is issued.

BKOUT
Specify to remove all BKUPEND recorded information for the current cycle for a specific application. This parameter can be used to reset the environment such that the BKMBKEND job can be rerun for the current application cycle.

CHANGED-SINCE-LAST-BKUP-RC(0|4|8)
(Parameter abbreviation is CSLBRC) Specify whether a return code of 0, 4, or 8 should be generated if, during BKUPEND, any data set identified as critical by Automated Critical Data Identification has an associated backup in Critical Backup Tracking and Inventory but the data set has changed since that backup (the last backup).

For example, BKUPEND is run at a time when the application data should all be backed up, however if it is discovered that a data set change has been made, the backup is no longer current. The application could be running 24 hours a day every day of the week, or it could indicate that the BKUPEND is in the wrong place.

A message will be generated if at least one backup is encountered where the data has since changed.

If you are using this option, monitor the size of the Data Set Activity Database to ensure it is large enough. You will also need to be sure to delete old or invalid records from the database on a regular basis.

When PDS or VSAM data sets are migrated and recalled by HSM after they were backed up, these data sets *appear* to have been changed. The process of recalling opens the data sets for output, which updates the DAD change fields for these data sets, creating the appearance of change when in fact, the data sets have only been recalled. To prevent this problem, you can add job exclusion filters to the DAD processing. These job exclusion filters so that data from HSM jobs is prevented from updating the DAD.

Note the following:

- This option requires the change tracking feature, which is implemented as follows:
  1. Define a Data Set Activity Database.
  2. Set the PARMLIB BKMINI token **CHANGED_DATASET_TRACKING** to Y.
  3. Start the BKMSTSKD started task.
- Data sets from full volume physical dumps are excluded from this process and are not added to the Data Set Activity Database when the BKMINI token `CHANGED_DATASET_TRACKING = Y`.
- The return code specified using the **CHANGED_DATASET_TRACKING_RC** parameter is the final BKUPEND return code if one or more data sets have been changed since the last backup.

After the steps above have been set up, whenever the actual backups run, the data set names found in the backups are added to the Data Set Activity Database for changes to be tracked.

CHECK-CHANGED-SINCE-LAST-BKUP(Y|N)
(Parameter abbreviation is CSLB) Specify whether to report the changed since last backup status of each critical data set:

- Y—Identify data sets changed since the current backup.
- N—Do not identify data sets changed since the current backup.

EXCLUDE-ACCOMPANY(Y|N)
This option controls whether BKMBKEND looks for a backup in the Advanced Backup and Recovery for z/OS Inventory Data Set for any critical data set classified in the Automated Critical Data Identification application as an ABARS ACCOMPANY data set.

- Y—Do not search and report for backups for ACCOMPANY data sets.
- N—Search and report for backups for ACCOMPANY data sets.

If Automated Critical Data Identification builds backup control statements (a Selection Data Set) for any utility other than ABARS, data sets classified as ACCOMPANY are switched to an INCLUDE. In this case, there is no point in classifying tapes as ACCOMPANY because they end up being copied by other means.

To avoid misleading 'no backup found' errors in BKMBKEND, specify EXCLUDE-ACCOMPANY(Y) if you have classified any tape data set as an ABARS ACCOMPANY using Automated Critical Data Identification, but do not back them up or ACCOMPANY them using an ABARS backup tracked by ABARS Manager.

ACCOMPANY is an ABARS term for tape data sets that you do not want copied because your intent is to transport the original to a disaster recovery site.

For tapes classified as an ABARS ACCOMPANY in Automated Critical Data Identification, if Automated Critical Data Identification builds an ABARS Selection Data Set, the tape data set names are created

within the ABARS ACCOMPANY keyword. ABARS does not copy ACCOMPANY data sets but does include the data set names in the control information created during the ABARS backup.

If an ABARS backup is tracked by ABARS Manager, the Advanced Backup and Recovery for z/OS Inventory Data Set will contain an entry indicating that the tape data set was indeed 'accompanied' and BKUPEND will find the match in the Advanced Backup and Recovery for z/OS Inventory Data Set.

EXCLUDE-ALLOCATE(Y|N)
This option controls whether BKMBKEND looks for a backup in the Advanced Backup and Recovery for z/OS Inventory Data Set for any critical data set classified in the Automated Critical Data Identification application as an ABARS ALLOCATE.

- Y—Do not search and report for backups for ALLOCATE data sets.
- N—Search and report for backups for ALLOCATE data sets.

If Automated Critical Data Identification builds backup control statements (a Selection Data Set) for any utility other than ABARS, data sets classified as ALLOCATE are switched to an INCLUDE. In this case, the data set is backed up and BKUPEND should find the backups in Critical Backup Tracking and Inventory.

To avoid misleading 'no backup found' errors in BKMBKEND, specify EXCLUDE-ALLOCATE(Y) if you classify any data set through Automated Critical Data Identification as an ALLOCATE, but do no back them up or ALLOCATE them through an ABARS backup tracked by ABARS Manager.

Data sets identified by Automated Critical Data Identification as critical can be categorized as ALLOCATE, as opposed to INCLUDE. A typical scenario where Automated Critical Data Identification categorizes a data set as ALLOCATE is when the first usage of an existing data set (for example, DISP OLD) is for output. Because the first usage of the data set is to write over existing data, to save backup time, the data set merely needs to be allocated for a rerun to satisfy the initiator expecting to find it (DISP OLD).

If Automated Critical Data Identification generates a Selection Data Set in the ABARS format, ALLOCATE data sets are placed within the ABARS ALLOCATE keyword because ABARS has the functionality to collect and save the attributes of a data set so it can allocate the data set during a recovery. If Automated Critical Data Identification generates a Selection Data Set for a utility that does not support ALLOCATEs, they are changed to INCLUDEs.

If you have Automated Critical Data Identification critical data sets marked as ALLOCATE, and ABARS is used and tracked by ABARS Manager, BKUPEND matches them to the Critical Backup Tracking and Inventory record.

INCLUDE-HSM(Y|N)
This option controls whether BKMBKEND looks for a backup in the DFSMShsm Incremental Backup inventory for any critical data set identified by Automated Critical Data Identification.

- Y—Search and report for backups made by DFSMShsm Incremental Backup.
- N—Do not search and report for backups made by DFSMShsm Incremental Backup.

INCLUDE-PHYSICAL(Y|N)
This option controls whether BKMBKEND looks for a backup in the Critical Backup Tracking and Inventory tracked volume dump inventory.

- Y—Search and report for backups made by tracked volume dumps.
- N—Do not search and report for backups made by tracked volume dumps.

NO-BKUP-FOUND-RC(0|4|8|)
(Parameter abbreviation is NBFRC) Specifies whether a return code of 0, 4, or 8 should be generated if any data set identified as critical in Automated Critical Data Identification does not have a corresponding backup in Critical Backup Tracking and Inventory. A message will be generated if at least one not-found backup is encountered. The message is suffixed with an I, W, or E, depending on whether a return code of 0, 4, or 8 is selected.

NOFLUSH
(Parameter alias is **NODADFLUSH**) This option controls whether the DAD buffers are flushed and written to the DAD database for BKUPEND changed data set analysis. If you are tracking changed data but not reporting on changed data for this application, using this option reduces the amount of time required to execute the BKMBKEND job.

NOWAIT
Use this keyword to perform critical data set backup analysis without ensuring all prior backups have been recorded in the IDS. Use this option only if you are certain the related backups have been recorded in the IDS.

RERUN
This option permits multiple executions of the BKMBKEND job without intervening BKMAPLEN jobs. The previously recorded BKUPEND information is replaced by the final BKUPEND information.

Examples

The following provides an example of the application name being specified using the **APPL** keyword as input to program BKMBMAIN:

```
//BKMBKEND EXEC BKMBPROC
//BKMBATCH.SYSIN DD *
  BKUPEND APPL(APPLRR)
          NO-BKUP-FOUND-RC(4)
          BAD-BKUP-RC(9999)
          CHANGED-SINCE-LAST-BKUP-RC(4)
//
```

The following provides an example of the application name being specified using the **BPARM** keyword in the BKMBPROC procedure:

```
//BKMBKEND EXEC BKMBPROC,BPARM='APPL(APPLRR)'
//BKMBATCH.SYSIN DD *
  BKUPEND
          NO-BKUP-FOUND-RC(4)
          BAD-BKUP-RC(9999)
          CHANGED-SINCE-LAST-BKUP-RC(4)
//
```

Using the Application Backup and Restore feature

After Critical Backup Tracking and Inventory and Automated Critical Data Identification are configured, you can use the information in this topic to use the Application Backup and Restore feature.

1. Create the Critical Backup Tracking and Inventory BIF filters to enable the started task to track the application backups.
2. Add the Automated Critical Data Identification Application Data Inventory database name to the BKMINI product parameter library member so Critical Backup Tracking and Inventory can interface with Automated Critical Data Identification:
   a) Locate the :PRODUCT_DATASETS section.
   b) Add the Automated Critical Data Identification Application Data Inventory database name to token: ACDIDB.
   For example: `ACDIDB=` *`prefix.value`*`.ACDIDB`
   c) Execute the BKMJCLBL job (located in the product JCL library) to update the symbolic parameters in the BKMBPROC procedure in the product JCL library.
3. Customize and run the Critical Backup Tracking and Inventory BKMEXPIR job:
   a) Update the BKMEXPIR JCL with the product JCL library name in the JCLLIB statement.
   b) Submit the BKMEXPIR job.
4. Define an application to Automated Critical Data Identification:

   Important: The VERIFICATION GDG option, **GDS Names in Selection Data Set,** must be set to either `Absolute` or `Both` because relative generations cannot be resolved.

a) Configure the tokens in section :CBTI_BACKUP_END_OPTIONS of the product parameter library's BKMINI member to specify the BKUPEND default values for all new applications.

   Use the Automated Critical Data Identification Application SETUP Options panel to override the BKMINI BKUPEND specified options for existing applications.

b) (Optional) Create EXCL_BUT_CRITICAL filters. Data sets excluded from an Automated Critical Data Identification application but considered critical during restore should be added to the Automated Critical Data Identification application using the EXCL_BUT_CRITICAL filter. These could be data sets backed up by other applications that are required when restoring this application, or these could be data sets excluded from backup because the backup was embedded in the application.

5. Verify that the Automated Critical Data Identification BKMSTSKR started task is running and the Automated Critical Data Identification job collection has been completed.
6. Run the application.
7. When the application cycle completes, run the Automated Critical Data Identification BKMAPLEN program found in the Automated Critical Data Identification JCL library to signal the end of the application cycle.

   Unless overridden by options, BKMAPLEN increments the current Automated Critical Data Identification cycle number (refer to **Next RSP cycle number** and **Next JCL cycle number**), builds database records for the just-completed application cycle while applying any filters, and creates a Selection Data Set which can be used to drive a post-application cycle backup. BKUPEND can then be used to verify that all data sets in the list were backed up.
8. If the BKUPEND parameters you want to use are different than the ones specified for the application by the Automated Critical Data Identification Application SETUP Options panel, you can specify them as input parameters to the job in the BKMBKEND member found in the Advanced Backup and Recovery for z/OS JCL library.

   Run the BKMBKEND job.

   You must run the BKMBKEND job after BKMAPLEN for each application that has completed, and at the point in time when you know all backups associated with the just-completed application cycle should have completed.

   If all backups are embedded in the application, BKMBKEND can be placed immediately after BKMAPLEN. If some or all data sets are backed up after the application completes (that is, after BKMAPLEN) place BKMBKEND after the last of all backups are finished.

Reporting on changed data status

Complete these steps to use the CHANGED_SINCE_LAST_BKUP_RC and CHECK_CHANGED_SINCE_LAST_BKUP parameters for BKUPEND processing.

1. Set up the BKMSTSKD (DAD) started task.
2. Set the token CHANGED_DATASET_TRACKING to Y in BKMINI.
3. Set tokens CHANGED_SINCE_LAST_BKUP_RC and CHECK_CHANGED_SINCE_LAST_BKUP in BKMINI
4. Set application Automated Critical Data Identification SETUP options CHANGED_SINCE_LAST_BKUP_RC and CHECK_CHANGED_SINCE_LAST_BKUP.
5. Start the BKMSTSKD (DAD) started task.

The data set names found in the backups are added to the Data Set Activity Database (DAD) for changes to be tracked.

Note: Data sets from full volume physical dumps tracked by Critical Backup Tracking and Inventory are excluded from this process and not added to the DAD when BKMINI token CHANGED_DATASET_TRACKING = Y.

The return code specified by the CHANGED_DATASET_TRACKING_RC parameter is the final return code produced by the BKMBKEND job if one or more data sets have been changed since the last backup.

Chapter 18. Setup and configuration for ABARS Manager

This topic provides procedures for setting up aggregate and group filter management.

Setting up aggregate management

This topic provides procedures for adding aggregates to the IDS, changing aggregates already in the IDS, and deleting aggregates from the IDS.

Advanced Backup and Recovery for z/OS aggregates are not automatically added by Advanced Backup and Recovery for z/OS to the Inventory Data Set (IDS) because many organizations prefer to manage aggregate information themselves. You can use the Advanced Backup and Recovery for z/OS menu and the appropriate line command to add to the IDS the aggregates you want to manage.

Note: After you have added aggregates to the IDS, you can use line commands to change the Advanced Backup and Recovery for z/OS attributes of the aggregates stored in the IDS or to delete them from the IDS.

1. Type 1 at the **Command** prompt on the **Setup and Configuration** panel, and then press Enter.

   Advanced Backup and Recovery for z/OS displays the **Aggregate Management with SMS Interface Setup** panel.
2. Type the appropriate aggregate name or mask in the **Aggregate name or mask** field on the **Aggregate Management with SMS Interface Setup** panel.
3. If you want to save the aggregate name or mask you specified so that Advanced Backup and Recovery for z/OS uses that specification by default every time the active user ID accesses this panel, type / to the left of the **Store values to be used as default for this USERID** field.
4. If you want to skip this panel whenever you select the Aggregate Management with SMS Interface option from the main menu, type / to the left of the **Don't show this panel again** field.
5. Press Enter to display the **Aggregate Management with SMS Interface** panel, which shows a combination of data input to Advanced Backup and Recovery for z/OS and data that DFSMS captured from ISMF.
6. Add to the Advanced Backup and Recovery for z/OS Inventory Data Set (IDS) the aggregates you want tracked and managed by ABARS Manager by specifying a mask in the **Aggregate** field or typing A in the **C** column to the left the appropriate aggregate's name, and then press Enter.
7. Press Enter to add to the IDS all aggregates that match the mask you specified or to add the individual aggregate you selected from the list with the **Add** line command.
8. Repeat steps 6 and 7 until you have added all the aggregates you want to manage using ABARS Manager.

Using line commands to manage aggregates

This topic tells you how to use line commands on the **Aggregate Management with SMS Interface** panel to perform various management tasks on aggregates and view aggregate information.

The **Aggregate Management with SMS Interface** panel shows a list of aggregates to which you can apply line commands.

Note: Be aware that until an aggregate has been added to the Advanced Backup and Recovery for z/OS IDS, not all of the line commands listed on the panel are valid for that aggregate. For example, you cannot use the **C** line command on an aggregate that has not been added to the IDS.

1. Type the appropriate line command in the **C** column to the left of the aggregate's name, and then press enter.

Option	Description
A	Add aggregate to the IDS.
B	Submit a backup for the selected aggregate.
C	Change the information stored in the IDS for this aggregate.
D	Delete the selected aggregate from the IDS.
I	Display information from the IDS and from SMS for the selected aggregate.
BI	Browse the instruction data set (if one exists) for the selected aggregate.
EI	Edit the instruction data set (if one exists) for the selected aggregate.
BL	Browse the SDSL parameters.
CL	Convert an ABARS-defined or user-defined Selection Data Set to SDSL format.
EL	Edit SDSL parameters.
BS	Browse Selection Data Sets.
ES	Edit Selection Data Sets.
BKM	For the selected aggregate, invoke ABARS Manager.

2. Use the appropriate primary command to manipulate the list of aggregates shown on the **Aggregate Management with SMS Interface** panel:

Option	Description
ADD	Adds any aggregate matching the name or mask specified to ABARS Manager, using the values identified on the ISMF aggregate definition. For example: ADD ALL, ADD *, ADD PROD*
M	Invoke the Monitoring feature.
REF	Refresh the display.
RES	Reset the screen values to their default settings.
SA	Sort by aggregate name.
SB	Sort to show aggregates in the IDS at the top of the display, sorted by aggregate. All other aggregates are sorted to the bottom of the list.
SM	Sort by management class.

Adding an aggregate to the IDS

Before an aggregate can be tracked and managed by ABARS Manager, you must add it to the Advanced Backup and Recovery for z/OS Inventory Data Set (IDS) using the **Aggregate Management with SMS Interface** panel.

You can use ABARS Manager to view information about an aggregate, regardless of whether its definition has been added to the IDS.

1. Type 1 at the **Command** prompt on the **Main Menu** and then press Enter.
2. Type 1 at the **Command** prompt on the **Setup and Configuration** panel, and then press Enter.

   Advanced Backup and Recovery for z/OS displays the **Aggregate Management with SMS Interface Setup** panel.
3. Type the aggregate name or mask for the aggregate(s) you want to display and press Enter.

   If the aggregate or aggregates are already in the IDS, YES appears in the **In IDS** column; if YES does not appear in the **In IDS** column, the specified aggregates can be added to the IDS.

4. Type A in the **C** column to the left of the **Aggregate** column or type `ADD mask` in the **Command** line and press Enter to display the **Aggregate Control** panel.
5. Make the appropriate specifications on the **Aggregate Control** panel.

The following figure shows the A line command being used to add the IVPSTND aggregate to the IDS.

```
  Menu  Diagnostics  Preferences
    Advanced Backup and Recovery for z/OS        Row 2 to 10 of 285
             Aggregate Management w/SMS Interface
Command ===> _____________________________________________ Scroll ===> PAGE

Primary Commands: ADD mask,M,REFresh,RESet,SA,SB,SM
   Line Commands: A,B,C,D,I,BI,EI,BL,CL,EL,BS,ES,BKM

               In             Retain         |---- SMS Summary Information ---|
 C  Aggregate  BKM Use       Bkup Rcv          # of           Copy Copy Retain
 _  *_______   IDS SDSL Incr Gens Gens        Copies MGMTCLAS Ser  Tech  Vers
-------------------------------------------------------------------------------
___ IVPSDSNU   YES NO   NO      5    1           1   ABARS5   CONT STND     5
___ IVPSTD     YES NO   NO      5    1           1   ABARS5   CONT STND     5
___ IVP2SDS    YES NO   NO      5    1           1   ABARS5   CONT STND     5
a__ IVPSTND    --- ---  ---     -    -           2   ABARS2   CONT STND     2
```

About using the Backup Gens/Groups to Retain field for Incremental ABARS Manager backups

This topic provides important information about how using the **Backup Gens/Groups to Retain** option affects backups run by Incremental ABARS Manager.

Using the **Backup Gens/Groups to Retain** field on the **Aggregate Control** panel can have two different meanings depending on whether you also specify Y for the **Identify as Incremental** field.

- If you specify N for the **Identify as Incremental** field, then use the **Backup Gens/Groups to Retain** field to specify how many base backup generations you want kept. A typical use for this situation is to allow recovery of a backup generation other than the most current.
- If you specify Y for the **Identify as Incremental** field, then use the **Backup Gens/Groups to Retain** field to specify how many groups of generations you want kept, where a group is base backup plus all associated incremental backups. When you specify Y in the **Identify as Incremental** field, incremental backup generations for this aggregate are associated with the previous base backup generation as a group. When the next base backup is run and whenever a subsequent incremental backup is run, another group is retained in the IDS. The number of generations in a group varies, depending on how many incremental backups are associated with a base backup. For recovery and expiration purposes, ABARS Manager views a group of generations as a single generation. For this reason, ABARS Manager, unlike native ABARS, must control the expiration of ABARS backups. This is also why an aggregate should use a management class with retain versions 9999 or NOLIMIT

Example

Consider an example in which one base backup and two incremental backups have been run for an aggregate. These three generations must be retained as a group to facilitate a complete recover. If you specify 1 for the **Backup Gens/Groups to Retain** field and another base backup is run, the previous three generations are expired as a group.

Submitting an ABARS Manager backup of an aggregate

Use the information in this topic to submit an ABARS Manager backup by using a line command on the **Aggregate Management with SMS Interface** panel.

You can submit an ABARS Manager backup on an aggregate directly from the **Aggregate Management with SMS Interface** panel. Be aware that you cannot run more than one ABARS Manager backup against a selected aggregate at any one point in time.

1. Type B in the **C** column for the aggregate you want backed up.

2. Press Enter to display the **Submit Backup** panel.
3. Use the online help to make the appropriate specifications on the **Submit Backup** panel.
4. Submit the backup for execution.

The following figure shows an ABARS Manager backup being specified for aggregate IVPSTD.

```
   Menu  Diagnostics  Preferences

     IBM Tivoli Advanced Backup and Recovery for z/OS  Row 2 to 10 of 285
           Aggregate Management w/SMS Interface
 Command ===> _____________________________________________ Scroll ===> PAGE

 Primary Commands: ADD mask,M,REFresh,RESet,SA,SB,SM
    Line Commands: A,B,C,D,I,BI,EI,BL,CL,EL,BS,ES,BKM

                 In             Retain         |---- SMS Summary Information ---|
  C  Aggregate  BKM Use       Bkup Rcv          # of           Copy Copy Retain
  _  I*______   IDS SDSL Incr Gens Gens        Copies MGMTCLAS Ser  Tech  Vers
-------------------------------------------------------------------------------
 ___ IVP        --- ---  ---     -    -           1   ABARS1   CONT STND     1
 ___ IVPINCR    YES NO   YES  9999    2           1   INCR     CONT STND  NLIM
 ___ IVPSDSNU   YES NO   NO       5    1           1   ABARS5   CONT STND     5
 b__ IVPSTD     YES NO   NO       5    1           1   ABARS5   CONT STND     5
 ___ IVP2SDS    YES NO   NO       5    1           1   ABARS5   CONT STND     5
 ****************************** Bottom of data ********************************
```

About concurrent copy

DFSMShsm issues messages to the console regarding the status of the DFSMSdss Concurrent Copy operation. Message ARC6402I is issued when the Concurrent Copy session initialization is complete. Message ARC6402I indicates the number of data sets for which Concurrent Copy initialization was successful or unsuccessful.

If all data sets to be backed up by DFSMSdss are eligible for Concurrent Copy, message ARC6402I is issued as soon as Concurrent Copy initialization is complete. At this point, for sites that do not have AUTO operations, a REXX EXEC can be used to submit a job to act as a trigger for the scheduler.

When the **CCTRIGGER** parameter in the BKMINI member of the product parameter library is set to Y, and the management class causes Concurrent Copy, ABARS Manager intercepts the messages during DFSMSdss processing and automatically issues the REXX EXEC at 'commit'.

A sample REXX EXEC is provided in member BKMCCTRG of the product parameter library. You can modify this any way you like. It is just a sample. If you want to create separate job names for separate aggregates, you must add REXX logic. Both the aggregate name and a status parameter are passed to the REXX EXEC. The BKMABARS procedure, which HSM uses to launch an ABARS **ABACKUP**, has a SYSEXEC DD from which the REXX can be launched.

If any data set is not eligible for Concurrent Copy, message ARC6402I is issued only when the entire DFSMSdss backup is physically complete, which could be several hours later. ABARS Manager provides several **ABACKUP** parameters to enhance this process. If a failure occurs during Concurrent Copy processing and these **ABACKUP** commands have been provided, the following occurs:

- Continue – Concurrent Copy failure is to be ignored and continue processing.
- WTOR – A message will inform the console operator of the problem and the appropriate course of action.
- ABEND – The task is to stop the backup for this aggregate and the job is to ABEND as soon as possible.

Because the use of Concurrent Copy for an aggregate backup is controlled by the SMS management class, if you need to change the Concurrent Copy options for an aggregate, you must change the management class definition for the aggregate and reactivate the updated SMS configuration, unless you are using ABARS Manager. ABARS Manager provides additional flexibility in the use of Concurrent Copy. You can temporarily and dynamically disable the use of Concurrent Copy for an entire aggregate or for only the backup of the activity log and instruction data set.

Changing an aggregate in the IDS

Use the information in this topic to change an aggregate's option settings in the IDS.

After an aggregate has been added to the Inventory Data Set (IDS), you can use the **Aggregate Management with SMS Interface** and **Aggregate Control** panels to modify the option settings stored for it in the IDS.

1. Type 1 at the **Command** prompt on the **Main Menu**, and then press Enter.
2. Type 1 at the **Command** prompt on the **Setup and Configuration** panel, and then press Enter.
3. In the **Aggregate** field, type the aggregate name or mask for the aggregate whose information you want to modify, and then press Enter to display on the lower half of the panel a list that includes the appropriate aggregate.
4. In the **C** column next to the appropriate aggregate's name, type C and then press Enter to display the **Aggregate Control** panel.
5. Make the appropriate specifications on the **Aggregate Control** panel and save your changes.

Deleting an aggregate from the IDS

You can use the **Aggregate Management with SMS Interface** panel to remove an aggregate from the Inventory Data Set (IDS) without removing the aggregate from ISMF.

1. Type 1 at the **Command** prompt on the **Main Menu**, and then press Enter.
2. Type 1 at the **Command** prompt on the **Setup and Configuration** panel, and then press Enter to display the **Aggregate Management with SMS Interface** panel.
3. In the **Aggregate** field, type the aggregate name or mask for aggregate whose information you want to modify, and then press Enter to display on the lower half of the panel a list that includes the appropriate aggregate.
4. In the **C** column next to the appropriate aggregate's name, type D and then press Enter to display the **Aggregate Control** panel.
5. When the delete confirmation panel appears, type Y in the **REPLY** field, and then press Enter.

The following figure shows the D line command being used to delete the TEST1 aggregate from the IDS.

```
     IBM Tivoli Advanced Backup and Recovery for z/OS  Row 2 to 10 of 285
              Aggregate Management w/SMS Interface
Command ===> _____________________________________________ Scroll ===> HALF

Primary Commands: ADD mask,M,REFresh,RESet,SA,SB,SM
   Line Commands: A,B,C,D,I,BI,EI,BL,CL,EL,BS,ES,BKM

               In             Retain         |---- SMS Summary Information ---|
 C  Aggregate  BKM Use       Bkup Rcv          # of          Copy Copy Retain
 _  *_______   IDS SDSL Incr Gens Gens        Copies MGMTCLAS Ser  Tech  Vers
-------------------------------------------------------------------------------
D__ TEST1      YES NO   NO   9999    1           1   ABARS    FAIL STND  NLIM
___ TEST2      --- ---  ---     -    -           1   ABARS    FAIL STND  NLIM
___ TEST3      YES NO   NO      2    1          **   ******** **** ****  ****
___ TEST4      YES NO   NO      5    1          **   ******** **** ****  ****
```

Viewing aggregate information

After an aggregate has been added to the Inventory Data Set (IDS), you can use the **Aggregate Management with SMS Interface and Aggregate Inquiry** panels to modify the information stored for that aggregate in the IDS.

1. Type 1 at the **Command** prompt on the **Main Menu** panel, and then press Enter to display the Setup and Configuration panel.
2. Type 1 at the **Command** prompt on the **Setup and Configuration** panel, and then press Enter to display the **Aggregate Management with SMS Interface** panel.

3. In the **Aggregate** field, type the aggregate name or mask for the aggregate whose information you want to view, and then press Enter to display on the lower half of the panel a list that includes the appropriate aggregate.
4. In the **C** column next to the appropriate aggregate's name, type `I` and then press Enter to display the **Aggregate Inquiry** panel, which shows all the information stored for this aggregate in the IDS.

The following figure shows an example of the **Aggregate Inquiry** panel and the information it displays for the selected aggregate, which in this case, is named IVPSTD:

```
                    ABM - Aggregate Inquiry
Command ===> ____________________________________________________________
                                                              More:     +
ABM SUMMARY INFORMATION
Aggregate Name            => IVPSTD
ABM Managed               => YES
Retain Backup Gens/Groups => 0005  0001-9999
Recovery Gens             => 0001  0001-9999
Identified as Incremental => N     Y or N

SDSL PROCESSING
Use SDSL Processing       => N     Y or N         _ (S) View SDSL Parameters
                                                  NO PARAMETERS EXIST
AUDIT TRAIL
Created by Userid         => DSA          Last Updated by Userid => DSA
Creation Date             => 2001/11/20   Last Update Date       => 2001/12/16
Creation Time             => 14:17:28     Last Update Time       => 11:15:11

                                                                    Return
BACKUP OPTIONS                                                 Y/N    Code
Change DSN found in multiple categories to suitable category => Y     => 4_
Change ALLOCATE of VBS data set with DSORG=NONE to INCLUDE   => Y     => 4_
Change ACCOMPANY of DASD data set to INCLUDE                 => Y     => 4_
Change ALLOCATE of Tape data set to INCLUDE                  => Y     => 4_
Change ALLOCATE of data set with blksize > 32760 to INCLUDE  => Y     => 4_
Remove uncataloged data sets                                 => Y     => 4_
Remove data sets resulting in catalog lookup error           => Y     => 4_
Change aliases references to true names                      => Y     => 4_
Remove orphaned aliases (true name not found)                => Y     => 4_
Change ICF Catalogs to ALLOCATE                              => Y     => 4_
Remove GDG base name references other than ALLOCATE          => Y     => 4_
Fuzzy Catalog Status detected during ABM Pre-Processing              => 0_
                                                                    Return
SDSL OPTIONS (Only referenced if "Use SDSL processing" is Y)   Y/N    Code
Remove DCOLLECTed DSN when catalog points to a different VOL => Y     => 4

SMS SUMMARY INFORMATION
Aggregate Description
IVP AGGREGATE TO TOLERATE ENQ FAILURE

Number of Copies          => 01
Management Class          => ABARS5
Retain Generations        => 0005
Copy Serialization        => CONTINUE
Copy Technique            => STANDARD
Last Update Userid        => DSA
Last Update Date/Time     => 2001/11/20 14:11
instruction data set      => N/A
Output Data Set Prefix    => BKMI.TODISK.TOL.IVPSTD
Selection Data Set Name(s)
=> BKMI.IVPSTD.TOL.SELECT
=>
=>
```

Browsing an aggregate's instruction data set

You can use the **Aggregate Management with SMS Interface** panel's **BI** line command to browse the instruction data set for an aggregate.

Be aware that you cannot browse an instruction data set until it has been defined to ISMF and cataloged.

1. Type 1 at the **Command** prompt on the **Main Menu**, and then press Enter.
2. Type 1 at the **Command** prompt on the **Setup and Configuration** panel, and then press Enter.

3. In the **Aggregate** field, type the aggregate name or mask for the aggregate whose instruction data set you want to browse, and then press Enter to display on the lower half of the panel a list that includes the appropriate aggregate.
4. In the **C** column next to the appropriate aggregate's name, type `BI` and then press Enter to open and ISPF Browse panel that shows the contents of the instruction data set.

Editing an aggregate's instruction data set

You can use the **Aggregate Management with SMS Interface** panel's **EI** line command to edit the instruction data set for an aggregate.

Be aware that you cannot edit an instruction data set until it has been defined to ISMF and cataloged.

1. Type 1 at the **Command** prompt on the **Main Menu**, and then press Enter.
2. Type 1 at the **Command** prompt on the **Setup and Configuration** panel, and then press Enter.
3. In the **Aggregate** field, type the aggregate name or mask for the aggregate whose instruction data set you want to edit, and then press Enter to display on the lower half of the panel a list that includes the appropriate aggregate.
4. In the **C** column next to the appropriate aggregate's name, type `EI` and then press Enter to open and ISPF Edit panel that shows the contents of the instruction data set.
5. Make the appropriate changes and save them.

Browsing an aggregate's SDSL parameters

You can use the **Aggregate Management with SMS Interface** panel's **BL** line command to browse the SDSL parameters (if they exist) for an aggregate.

1. Type 1 at the **Command** prompt on the **Main Menu**, and then press Enter.
2. Type 1 at the **Command** prompt on the **Setup and Configuration** panel, and then press Enter.
3. In the **Aggregate** field, type the aggregate name or mask for the aggregate whose SDSL parameters you want to browse, and then press Enter to display on the lower half of the panel a list that includes the appropriate aggregate.
4. In the **C** column next to the appropriate aggregate's name, type `BL` and then press Enter.

   If no SDSL parameters exist for the aggregate, a message to that effect appears; if SDSL parameters do exist for the selected aggregate, an ISPF Browse window displays them.

Converting an aggregate's Selection Data Set to SDSL parameters

You can use the **Aggregate Management with SMS Interface** panel's **CL** line command to convert an aggregate's ISMF assigned Selection Data Set or a user-supplied Selection Data Set to SDSL parameters format, and then store them in the IDS.

1. Type 1 at the **Command** prompt on the **Main Menu**, and then press Enter.
2. Type 1 at the **Command** prompt on the **Setup and Configuration** panel, and then press Enter.
3. In the **Aggregate** field, type the aggregate name or mask for the aggregate whose Selection Data Set you want to convert, and then press Enter to display on the lower half of the panel a list that includes the appropriate aggregate.
4. In the **C** column next to the appropriate aggregate's name, type `CL` and then press Enter.

   If the aggregate you selected already has SDSL parameters in the IDS, use the pop-up window to specify whether you want to cancel the conversion or to delete the existing parameters; continue with step 5. Otherwise, if the selected aggregate has no SDSL parameters already in the IDS, the **SDS(s) to SDSL Conversion** panel appears; go directly to step 6.
5. In the pop-up window's **REPLY** field, type the appropriate letter, and then press Enter:
   - Specify `N` to cancel your conversion request and return to the previous panel.
   - Specify `Y` to delete the existing SDSL parameters and display the **SDS(s) to SDSL Conversion** panel.

If you specify N, the **Aggregate Management with SMS Interface** panel appears. If you specify Y, continue with step 6.

6. On the **SDS(s) to SDSL Conversion** panel, make the appropriate specifications, and then press Enter.
   - If you specified that you wanted to convert the data sets, the SDSL parameters appear in an ISPF Edit panel, where you can make the appropriate changes.
   - If you specified that you wanted to cancel the conversion, the **Aggregate Management with SMS Interface** panel appears.

Editing an aggregate's SDSL parameters

You can use the **Aggregate Management with SMS Interface** panel's **EL** line command to edit an aggregate's SDSL parameters.

1. Type 1 at the **Command** prompt on the **Main Menu**, and then press Enter.
2. Type 1 at the **Command** prompt on the **Setup and Configuration** panel, and then press Enter.
3. In the **Aggregate** field, type the aggregate name or mask for the aggregate whose SDSL parameters you want to edit, and then press Enter to display on the lower half of the panel a list that includes the appropriate aggregate.
4. In the **C** column next to the appropriate aggregate's name, type EL and then press Enter.

   If the selected aggregate selected has SDSL parameters in the IDS, an ISPF Edit panel appears; continue with step 5. Otherwise, a message appears, indicating that there are no SDSL parameters for the selected data set.
5. On the ISPF Edit panel, make the appropriate changes to the SDSL parameters and save them.

Browsing an aggregate's Selection Data Sets

You can use the **Aggregate Management with SMS Interface** panel's **BS** line command to browse the Selection Data Sets for an aggregate.

Be aware that you cannot browse the Selection Data Sets for an aggregate until the aggregate has been defined to ISMF and cataloged.

1. Type 1 at the **Command** prompt on the **Main Menu**, and then press Enter.
2. Type 1 at the **Command** prompt on the **Setup and Configuration** panel, and then press Enter.
3. In the **Aggregate** field, type the aggregate name or mask for the aggregate whose Selection Data Set you want to browse, and then press Enter to display on the lower half of the panel a list that includes the appropriate aggregate.
4. In the **C** column next to the appropriate aggregate's name, type BS and then press Enter.
   - If there are multiple Selection Data Sets for the selected aggregate, use the pop-up window to select the one you want to browse.
   - If there is a single Selection Data Set for the selected aggregate, it appears in an ISPF Browse panel.

About PDSs and PDSEs

When do you use a PDS for an aggregate's Selection Data Set? When do you use a PDSE? This topic discusses the relative merits of using PDSs and PDSEs for the Selection Data Sets.

ABARS Manager is proactive in fixing errors if **SDS_MODIFICATION_OK=Y** in the product parameter library's BKMINI member.

To fix errors, ABARS Manager must write to the Selection Data Set. If you set **SDS_MODIFICATION_OK=N**, ABARS Manager does not fix any errors and therefore does not write to the Selection Data Set. Selection data sets set up in a PDS do not work well with ABARS Manager when **SDS_MODIFICATION_OK=Y** because only one program can write to a given PDS member at a time.

If PDS files are used, eventually failures occur because it is almost certain that two aggregates will execute simultaneously at some point in time. For this very reason, it is always recommend that sequential files be used. If the PDS structure is required, then use PDSE.

PDSEs have some minor disadvantages compared to sequential data sets:

- Adding a new Selection Data Set (member) to a PDSE could fail because the PDSE is out of space, whereas failure is less likely if Selection Data Sets are individual sequential data sets.
- ISPF, Automated Critical Data Identification, and ABARS Manager can work well together when accessing different members or even the same member of a PDSE. However, Automated Critical Data Identification or ABARS Manager could fail if other programs lock the entire library either because they must (a reorg) or because they were designed to lock the entire library when they really only needed exclusive use of a member.

  If the data set is locked out, ABARS Manager tries five times at 30-second intervals and then fail.

Editing an aggregate's Selection Data Set

You can use the **Aggregate Management with SMS Interface** panel's **ES** line command to edit the Selection Data Set for an aggregate.

1. Type 1 at the **Command** prompt on the **Main Menu**, and then press Enter.
2. Type 1 at the **Command** prompt on the **Setup and Configuration** panel, and then press Enter.
3. In the **Aggregate** field, type the aggregate name or mask for the aggregate whose SDSL parameters you want to edit, and then press Enter to display on the lower half of the panel a list that includes the appropriate aggregate.
4. In the **C** column next to the appropriate aggregate's name, type ES and then press Enter.

   If there are multiple Selection Data Sets for the selected aggregate, use the pop-up window to select the one you want to edit and press Enter to open an ISPF Edit panel; if there is a single Selection Data Set for the selected aggregate, it appears in an ISPF Edit panel.
5. On the ISPF Edit panel, make the appropriate changes to the Selection Data Set, and then save them.

About PDSs and PDSEs

When do you use a PDS for an aggregate's Selection Data Set? When do you use a PDSE? This topic discusses the relative merits of using PDSs and PDSEs for the Selection Data Sets.

ABARS Manager is proactive in fixing errors if **SDS_MODIFICATION_OK=Y** in the product parameter library's BKMINI member.

To fix errors, ABARS Manager must write to the Selection Data Set. If you set **SDS_MODIFICATION_OK=N**, ABARS Manager does not fix any errors and therefore does not write to the Selection Data Set. Selection data sets set up in a PDS do not work well with ABARS Manager when **SDS_MODIFICATION_OK=Y** because only one program can write to a given PDS member at a time.

If PDS files are used, eventually failures occur because it is almost certain that two aggregates will execute simultaneously at some point in time. For this very reason, it is always recommend that sequential files be used. If the PDS structure is required, then use PDSE.

PDSEs have some minor disadvantages compared to sequential data sets:

- Adding a new Selection Data Set (member) to a PDSE could fail because the PDSE is out of space, whereas failure is less likely if Selection Data Sets are individual sequential data sets.
- ISPF, Automated Critical Data Identification, and ABARS Manager can work well together when accessing different members or even the same member of a PDSE. However, Automated Critical Data Identification or ABARS Manager could fail if other programs lock the entire library either because they must (a reorg) or because they were designed to lock the entire library when they really only needed exclusive use of a member.

  If the data set is locked out, ABARS Manager tries five times at 30-second intervals and then fail.

Setting up group filter management

Group Filter Management allows you to create, change, and delete Group Filters within ABARS Manager to be used in conjunction with SDSL processing. You can set up group filter management from the **Group Filter Management** panel.

1. Type 1 at the **Command** prompt on the Advanced Backup and Recovery for z/OS **Main Menu**, and then press Enter.

   Advanced Backup and Recovery for z/OS displays the **Setup and Configuration** panel.
2. Type 2 at the **Command** prompt on the **Setup and Configuration** panel.

   The **Group Filter Management** panel appears.
3. At the **Command** prompt or in the **CMD** column, type the appropriate primary or line command for your task.

Adding a group filter

You can use the **Group Filter Management** panel to add group filter definitions to ABARS Manager.

1. Type 1 on the Advanced Backup and Recovery for z/OS **Main Menu**, and then press Enter to display the **Setup and Configuration** panel.
2. Type 2 at the **Command** prompt on the **Setup and Configuration** panel, and then press Enter to display the **Group Filter Management** panel.
3. At the **Command** prompt, type ADD and press Enter to display the **Create Group Filter** panel.
4. In the **Specify Group Filter Name** field, specify a name for the filter you are adding to ABARS Manager. Your entry cannot exceed 30 characters in length and cannot contain any blank spaces.
5. Press Enter to open an ISPF Edit session for the new filter.

Deleting a group filter

You can use the **Group Filter Management** panel to delete an unused or obsolete group filter definition from ABARS Manager.

1. Type 1 on the Advanced Backup and Recovery for z/OS **Main Menu**, and then press Enter to display the **Setup and Configuration** panel.
2. Type 2 at the **Command** prompt on the **Setup and Configuration** panel, and then press Enter to display the Group Filter Management panel.
3. In the **Group Filter** field, type * and press Enter to display on the lower half of the panel a list of group filters known to ABARS Manager.
4. In the **Cmd** column next to the name of the group filter you want to delete, type D and press Enter to pop up a delete confirmation window.
5. Type Y in the **REPLY** field and press Enter.

Editing a group filter

You can use the **Group Filter Management** panel to edit an ABARS Manager group filter definition.

1. Type 1 on the Advanced Backup and Recovery for z/OS **Main Menu**, and then press Enter to display the **Setup and Configuration** panel.
2. Type 2 at the **Command** prompt on the **Setup and Configuration** panel, and then press Enter to display the **Group Filter Management** panel.
3. In the **Group Filter** field, type * and press Enter to display on the lower half of the panel a list of group filters known to ABARS Manager.
4. In the **Cmd** column next to the name of the filter you want to edit, type E and press Enter to open an ISPF Edit session on the filter.

Chapter 19. Managing ABARS events

Use the Advanced Backup and Recovery for z/OS ABARS Events feature to display a filtered list of backup and recovery events.

From this filtered list of backup and recovery events, you can then do any of the following:

- View a list of data sets involved in an event.
- View detailed information about an event, including error, event record, summary, and activity log information.
- Delete an ABARS event record.
- Display space usage and allocation information for aggregates.
- View conflict, instruction, restart, and Selection Data Set information for ABACKUP and ARECOVER data sets.
- Browse or edit an aggregate's instruction data set, SDSL parameters, or Selection Data Set.
- Browse or edit cataloged restart and conflict ARECOVER data set information.
- Submit an ABACKUP or ARECOVER.
- Monitor ABARS backups or recoveries as they are executing and associate the ABARS started task with the actual aggregate and job execute it.

Displaying a list of ABARS events

You can use the **ABARS Events Search Setup** panel to create a filtered list of ABARS events.

When the list of events meeting your criteria appears, you can use line commands on any individual event in the list.

Note: The first time you select the ABARS Events option on the **Backup and Recovery Management** panel, the **ABARS Events Search Setup** panel appears automatically. If you want to skip the search setup panel on subsequent uses of this feature, type / in the Search setup panel's **Don't show this panel again** field.

1. Type 1 in the **Command** line on the **Backup and Recovery Management** panel, and then press Enter to display the **ABARS Events Search Setup** panel.
2. Use the **ABARS Events Search Setup** panel's online help to complete the fields that constitute the search criteria, and then press Enter to display the **ABARS Events** panel, which shows a list of all ABARS events that meet the criteria you specified.
3. Use the online help to explain the line and primary commands available from the **Command** line on the **ABARS Events** panel, or to sort the information shown.

Viewing ABACKUP or ARECOVER data set name lists

Use the **ABARS Events** panel's **N** line command to view a list of data sets included in an aggregate backup or restore event. From the generated list, you run further line commands on selected data sets.

1. Type 1 in the **Command** line on the **Backup and Recovery Management** panel, and then press Enter to display the **ABARS Events Search Setup** panel.
2. Use the **ABARS Events Search Setup** panel's online help to complete the fields that constitute the search criteria, and then press Enter to display the **ABARS Events** panel, which shows a list of all ABARS events that meet the criteria you specified.
3. Type N in the **C** column next to the appropriate ABACKUP (AB appears in **PGM** column) or ARECOVER (AR appears in the **PGM** column) event and press Enter to display the **View Data Set Name List** panel.
4. If you want to determine a data set's current cataloged status, type Y in the **Catalog Lookup** field and press Enter.

5. Use the **View Data Set Name List** panel's online help to guide your use of line commands for individual data sets and to issue primary commands from the **Command** prompt.
6. To view detailed information about a selected data set in the list, type I in the **C** column to the left of that data set's name, and then press Enter to display the information on the **View Data Set Detail** panel.

Viewing ABACKUP or ARECOVER event detail

Use the **ABARS Events** panel's **S** line command to view detailed information about an aggregate backup or restore event.

1. Type 1 in the **Command** line on the **Backup and Recovery Management** panel, and then press Enter to display the **ABARS Events Search Setup** panel.
2. Use the **ABARS Events Search Setup** panel's online help to complete the fields that constitute the search criteria, and then press Enter to display the **ABARS Events** panel, which shows a list of all ABARS events that meet the criteria you specified.
3. Type S in the **C** column next to the appropriate **ABACKUP** (AB appears in **PGM** column) or **ARECOVER** (AR appears in the **PGM** column) event and press Enter to display the **View Backup Detail** or **View Recovery Detail** panel.

Displaying space usage information for an aggregate

Use the ABARS Events panel's **SP** line command to view space usage or allocation information for a selected aggregate.

1. Type 1 in the **Command** line on the **Backup and Recovery Management** panel, and then press Enter to display the **ABARS Events Search Setup** panel.
2. Use the **ABARS Events Search Setup** panel's online help to complete the fields that constitute the search criteria, and then press Enter to display the **ABARS Events** panel, which shows a list of all ABARS events that meet the criteria you specified.
3. Type SP in the **C** column next to the appropriate aggregate in the list on the **ABARS Events** panel and press Enter to display the **Backup Space Display** panel.
4. Use the **Backup Space Display** panel's online help to get further information about the fields shown on the panel.

Viewing summary information for an ABACKUP or ARECOVER event

Use the **ABARS Events** panel's **SM** line command to view summary information for a backup event.

The **SM** line command provides output that shows the following types of information:

- For an **ABACKUP** event, you get generation information, backup event detail, a summary of the data sets created during the backup, and a summary of the data sets backed up.
- For an **ARECOVER** event, you get a recovery event summary, a list of data sets used by the recovery, data set name information for instruction, restart, conflict, and activity log data sets, and decompressed activity log information.

1. Type 1 in the **Command** line on the **Backup and Recovery Management** panel, and then press Enter to display the **ABARS Events Search Setup** panel.
2. Use the **ABARS Events Search Setup** panel's online help to complete the fields that constitute the search criteria, and then press Enter to display the **ABARS Events** panel, which shows a list of all ABARS events that meet the criteria you specified.
3. Type SM in the **C** column next to the appropriate **ABACKUP** (AB appears in **PGM** column) or **ARECOVER** (AR appears in the **PGM** column) event and press Enter to display the **Backup Event Summary** or **Recovery Event Summary** in an ISPF Browse panel. For Backup Event Summary information, be aware that a YES in the **Catlg** field indicates the current catalog status of that data set; NO indicates the data set is not currently cataloged (although it was required to be cataloged at the time of backup).

The following screen shows an example of the Backup Event Summary information:

```
Command ===>                                                  Scroll ===> PAGE
******************************** Top of Data ********************************

************************************
* ABACKUP GENERATION(S):           *
************************************
      Gen I Gen Type Date       Time     Backup Status  RCA RCB CFile Name
    ----- ----- ---- ---------- -------- -------------- --- --- ---------------
==>     0        B   2001/12/16 12:41:29 COMPLETE         0   0 BKMI.TODISK.TOL.
       -1        B   2001/12/16 12:35:57 COMPLETE         0   0 BKMI.TODISK.TOL.
      N/A        V   2001/11/28 12:00:28 COMPLETE         0   4 ** NONE FOUND **
      N/A        V   2001/11/28 11:58:41 COMPLETE         0   4 ** NONE FOUND **

************************************
* BACKUP EVENT DETAIL:             *
************************************
  Aggregate Name: IVP2SDS
  Backup Status : COMPLETE
  Return Code   : 0 / 0
  Backup Type   : BASE
  Jobname       : BKMBKUPQ
  Procname      : BKMBATCH
  Stepname      : BKM
  Run Date/Time : 2001/12/16 12:41:29
  End Date/Time : 2001/12/16 12:42:23
  Elapsed       :           00:00:50
  Log Date/Time : 2001/12/16 12:41:45

************************************
* DATASETS CREATED BY ABARS BACKUP: *
************************************
  Copy Set: 01
    Control File: BKMI.TODISK.TOL.IVP2SDS.C.C01V0002           SMS002
       Data File: BKMI.TODISK.TOL.IVP2SDS.D.C01V0002           SMS001
    Offline File: BKMI.TODISK.TOL.IVP2SDS.O.C01V0002            SMS001


************************************
* DATASETS BACKED UP BY ABARS:     *
************************************

Data Set Names                                Catlg Type      Level   GB  MB  KB
--------------------------------------------- ----- --------- ----- -------------
BKMI.IVP.CNTL                                 YES  INCLUDE   PRI            940.
BKMI.IVP.FB.DATASET                           YES  INCLUDE   PRI             55.
BKMI.IVP.FB.DATASET.EMPTY                     YES  INCLUDE   PRI             55.
BKMI.IVP.FB.DATASET1                          YES  INCLUDE   PRI             55.
BKMI.IVP.FB.DATASET2                          YES  INCLUDE   PRI             55.
```

The following figure shows an example of the Recovery Event Summary information:

```
ISRBROBF  SUMMARY: IVPSDSNU 2001/12/18 18:18         Line 00000000 Col 001 080
 Command ===>                                                 Scroll ===> PAGE
******************************** Top of Data ********************************

**************************
* Recovery Event Summary: *
**************************
  Aggregate Name : IVPSDSNU
  Recovery Status: COMPLETE
  Return Code    : 0 / 0
  Recovery Type  : BASE
  Jobname        : BKMRECOI
  Procname       : BKMBATCH
  Stepname       : BKM
  Run Date/Time  : 2001/12/18 18:18:15
  End Date/Time  : 2001/12/18 18:19:05
  Elapsed        :           00:00:47
  Log Date/Time  : 2001/12/18 18:18:19
  Generation     : 0

***********************************
* Data Sets used by ABARS Recovery: *
***********************************
  Copy Set: 01
    Control File: BKMI.TODISK.TOL.IVPSDSNU.C.C01V0006          SMS002
       Data File: BKMI.TODISK.TOL.IVPSDSNU.D.C01V0006          SMS001
     Output File: BKMI.TODISK.TOL.IVPSDSNU.O.C01V0006          SMS001
```

```
*********************************
* Data Sets recovered by ABARS: *
*********************************

Data Set Names                                Status Type     Level   GB  MB  KB
-------------------------------------------------------------------------------
BKMI.IVP.CNTL                                  *EXCL                            .0
BKMI.IVP.FB.DATASET                            *EXCL                            .0
BKMI.IVP.FB.DATASET.EMPTY                            INCLUDE   PRI            55.3
BKMI.IVP.FB.DATASET1                           *EXCL                            .0
BKMI.IVP.FB.DATASET2                           *EXCL                            .0
BKMI.IVP.GDG1.G0001V00                         *EXCL                            .0
BKMI.IVP.GDG1.G0002V00                         *EXCL                            .0
BKMI.IVP.GDG1.G0003V00                         *EXCL                            .0

*****************************************************************
* INSTRUCTION DSN: N/A                                          *
*****************************************************************

*****************************************************************
* RESTART DSN    : N/A                                          *
*****************************************************************

*****************************************************************
* CONFLICT DSN   : N/A                                          *
*****************************************************************
*****************************************************************
* ACTIVITY LOG   : HSMACT.H1.ARECOVER.IVPSDSNU.D00353.T181819   *
*****************************************************************

*****************************
* ACTIVITY LOG DECOMPRESSED *
*****************************

1PAGE 0001  DFSMSHSM 1.5.0  DATA  FACILITY  HIERARCHICAL  STORAGE  MANAGER   00.
0ARC6102I AGGREGATE RECOVERY STARTING USING CONTROL FILE DATASET BKMI.TODISK.TO
  STARTED TASK = BKMABARS.ABAR0117
0ARC6030I ACTIVITY LOG FOR CONTROL FILE DATASET BKMI.TODISK.TOL.IVPSDSNU.C.C01V
 HSMACT.H1.ARECOVER.IVPSDSNU.D00353.T181819
0ARC6178I INSTALLATION-WIDE EXIT MODULE ARCCREXT WAS SUCCESSFULLY LOADED DURING
 SET BKMI.TODISK.TOL.IVPSDSNU.C.C01V0006
0ARC6178I INSTALLATION-WIDE EXIT MODULE ARCSKEXT WAS SUCCESSFULLY LOADED DURING
 SET BKMI.TODISK.TOL.IVPSDSNU.C.C01V0006
0ARC6115I AGGREGATE RECOVERY USING CONTROL FILE DATASET BKMI.TODISK.TOL.IVPSDSN
   ......
   SMS001
0ARC6119I DATA CLASS NAMES EXISTED DURING AGGREGATE BACKUP OF AGGREGATE GROUP IV
 BKMI.TODISK.TOL.IVPSDSNU.C.C01V0006

...
```

Viewing ABACKUP or ARECOVER event activity logs

Use the **ABARS Events** panel's **VA** line command to view the activity log for an **ABACKUP** or **ARECOVER** event.

If the activity log was created by ABARS during the **ABACKUP** or **ARECOVER** process, the activity log was stored in the IDS. The version of the activity log displayed in the ISPF Browse panel is a copy of the original activity log stored in the IDS during the **ABACKUP** or **ARECOVER** processing.

1. Type 1 in the **Command** line on the **Backup and Recovery Management** panel, and then press Enter to display the **ABARS Events Search Setup** panel.
2. Use the **ABARS Events Search Setup** panel's online help to complete the fields that constitute the search criteria, and then press Enter to display the **ABARS Events** panel, which shows a list of all ABARS events that meet the criteria you specified.
3. Type VA in the **C** column next to the appropriate **ABACKUP** (AB appears in **PGM** column) or **ARECOVER** (AR appears in the **PGM** column) event and press Enter to display the activity log (if one exists) in an ISPF Browse panel.

The following figure shows an example activity log for an **ABACKUP** event.

```
ISREDDE2   HSMACT.H1.ABACKUP.IVP2SDS.D00351.T124145       Columns 00001 00072
Command ===>                                                Scroll ===> PAGE
****** ***************************** Top of Data ******************************
==MSG> -CAUTION- Data contains invalid (non-display) characters. Use command
==MSG>           ===> FIND P'.' to position cursor to these
==MSG> -Warning- The UNDO command is not available until you change
==MSG>           your edit profile using the command RECOVERY ON.
000001 1PAGE 0001  DFSMSHSM 1.5.0  DATA  FACILITY  HIERARCHICAL  STORAGE  MANAG
000002 0ARC6054I AGGREGATE BACKUP STARTING FOR AGGREGATE GROUP IVP2SDS, AT 12:4
000003 0ARC6030I ACTIVITY LOG FOR AGGREGATE GROUP IVP2SDS WILL BE ROUTED TO HSM
000004 0ARC6178I INSTALLATION-WIDE EXIT MODULE ARCBEEXT WAS SUCCESSFULLY LOADED
000005  IVP2SDS
000006 0ARC6178I INSTALLATION-WIDE EXIT MODULE ARCTVEXT WAS SUCCESSFULLY LOADED
000007  IVP2SDS
000008 0ARC6152I THE FOLLOWING DATASETS WERE SUCCESSFULLY PROCESSED FROM THE A
000009  AGGREGATE GROUP IVP2SDS
```

Viewing ARECOVER input conflict data sets

Use the **ABARS Events** panel's **VC** line command to view conflict data sets that were used by ABARS as input to an aggregate's **ARECOVER**.

If a conflict data set is used by ABARS during an **ARECOVER** process, the conflict data set is stored in the IDS. When you view an input conflict data set, a copy of the conflict data set that was stored in the IDS is displayed on the ISPF Browse panel.

1. Type 1 in the **Command** line on the **Backup and Recovery Management** panel, and then press Enter to display the **ABARS Events Search Setup** panel.
2. Use the **ABARS Events Search Setup** panel's online help to complete the fields that constitute the search criteria, and then press Enter to display the **ABARS Events** panel, which shows a list of all ABARS events that meet the criteria you specified.
3. Type S in the **C** column next to the appropriate **ARECOVER** (AR appears in the **PGM** column) event and press Enter to display the **View Recovery Detail** panel.

   If a conflict data set was used as **ARECOVER** input, its file name appears in the **Conflict DSN In** field.
4. If the **View Recovery Detail** panel indicates that a conflict data set was used as input to the selected **ARECOVER** event, type S in the **C** column next to the **Conflict DSN In** field and press Enter to display the conflict data set in an ISPF Browse panel.

Viewing ABACKUP or ARECOVER event errors

Use the **ABARS Events** panel's **VE** line command to view information about errors that occurred during an ABARS event.

During the **ABACKUP** or **ARECOVER** process, the activity log is interrogated for warning or error messages. If warning or error messages exist, they are stored in the IDS, identified by the executing **ABACKUP** or **ARECOVER** event name.

1. Type 1 in the **Command** line on the **Backup and Recovery Management** panel, and then press Enter to display the **ABARS Events Search Setup** panel.
2. Use the **ABARS Events Search Setup** panel's online help to complete the fields that constitute the search criteria, and then press Enter to display the **ABARS Events** panel, which shows a list of all ABARS events that meet the criteria you specified.
3. Type VE in the **C** column next to the appropriate aggregate in the list on the **ABARS Events** panel and press Enter to display the error information in an ISPF Browse panel.

The following figure shows an example of the error information stored in the IDS for an ABACKUP event:

```
ISREDDE2   ERROR MESSAGES FOR AGGREGATE: XYZTEST            Columns 00001 00072
 Command ===>                                                 Scroll ===> PAGE
 ****** ***************************** Top of Data ******************************
*********************************************
 * ERRORS/WARNINGS FROM THE ABACKUP ACTIVITY LOG *
 *********************************************
ARC6158E CATALOG ERROR OCCURRED DURING SUPERLOCATE FUNCTION FOR DATA
         SET EDP.NOT.REALLY.ONE FOR AGGREGATE GROUP XYZTEST CATALOG
         RETURN CODE IS 8 REASON CODE IS 0
ADR380E (001)-DTDSC(04) DATA SET EDP.LAB.CATONLY NOT PROCESSED 13-0008

*********************************************
 * SUMMARY OF BKM MODIFICATIONS TO SDS           *
 *********************************************
EDP.LAB.CATONLY                            RETAINED   UNKNOWN CATALOG
EDP.NOT.REALLY.ONE                         RETAINED   NOT CATALOGUED

 *********************************************
 * SUMMARY OF ERROR / WARNING MESSAGES FROM BKM  *
 *********************************************
BKMS1022W ERROR OBTAINING VTOC DATA FROM TSO001 :EDP.LAB.CATONLY
BKMS1022W ERROR OBTAINING VTOC DATA FROM TSO001 :EDP.LAB.CATONLY

 *********************************************
 * BACKUP BYPASSED BY ARCBEEXT FOR:              *
 *********************************************
EDP.NOT.REALLY.ONE
EDP.NOT.REALLY.THREE
EDP.NOT.REALLY.TWO
********************************** Bottom of Data ****************************
```

Viewing ABACKUP instruction data sets

If an instruction data set exists during an **ABACKUP** event, that data set is stored in the IDS and you access that data set from the **ABARS Events** panel by using the **VI** line command.

1. Type 1 in the **Command** line on the **Backup and Recovery Management** panel, and then press Enter to display the **ABARS Events Search Setup** panel.
2. Use the **ABARS Events Search Setup** panel's online help to complete the fields that constitute the search criteria, and then press Enter to display the **ABARS Events** panel, which shows a list of all ABARS events that meet the criteria you specified.
3. Type VI in the **C** column next to the appropriate ABACKUP (AB appears in the **PGM** column) event and press Enter to display the instruction data set (if one exists for the selected aggregate) in an ISPF Browse panel.

Viewing restart data sets used as input to an ARECOVER event

If a restart data set is used as input to an **ARECOVER** event, that data set is stored in the IDS and you can use the **ABARS Events** panel to access and view that data set.

1. Type 1 in the **Command** line on the **Backup and Recovery Management** panel, and then press Enter to display the **ABARS Events Search Setup** panel.
2. Use the **ABARS Events Search Setup** panel's online help to complete the fields that constitute the search criteria, and then press Enter to display the **ABARS Events** panel, which shows a list of all ABARS events that meet the criteria you specified.
3. Type S in the **C** column next to the appropriate **ARECOVER** (AR appears in the **PGM** column) event and press Enter to display the **View Recovery Detail** panel.

   If a restart data set was used as **ARECOVER** input, its file name appears in the **Restart DSN In** field.
4. If the **View Recovery Detail** panel indicates that a restart data set was used as input to the selected **ARECOVER** event, type S in the **C** column next to the **Restart DSN In** field and press Enter to display the restart data set in an ISPF Browse panel.

Viewing ABACKUP Selection Data Sets

Selection data sets used during an **ABACKUP** process are stored in the IDS. Use the ABARS Events panel's **VS** line command to access and view these Selection Data Sets.

1. Type 1 in the **Command** line on the **Backup and Recovery Management** panel, and then press Enter to display the **ABARS Events Search Setup** panel.
2. Use the **ABARS Events Search Setup** panel's online help to complete the fields that constitute the search criteria, and then press Enter to display the **ABARS Events** panel, which shows a list of all ABARS events that meet the criteria you specified.
3. Type VS in the **C** column next to the appropriate **ABACKUP** (AB appears in the **PGM** column) event and press Enter to display the **SDS(s) Used Before/During Backup** panel.
4. Type v to the left of the name of the Selection Data Set you want to view, and then press Enter.

   The data set appears in an ISPF View panel.

Deleting an ABARS event record

Use the **ABARS Events** panel's **DE** line command to delete the IDS record for an incremental or non-incremental backup. You can delete only those events that did not create an ABARS C, D, or O file.

If an incremental backup is being run and there is no valid base backup, ABARS Manager issues a message and quits. If the last event was a bad base backup, you can take any of the following actions:

- Use the **DE** line command to delete the bad base backup from the IDS, and then run another base backup.
- Leave the bad base backup in the IDS and run another base backup.
- Delete the bad base event, and then run an incremental backup that will associate to the previous cycle.

1. Type 1 in the **Command** line on the **Backup and Recovery Management** panel, and then press Enter to display the **ABARS Events Search Setup** panel.
2. Use the **ABARS Events Search Setup** panel's online help to complete the fields that constitute the search criteria, and then press Enter to display the **ABARS Events** panel, which shows a list of all ABARS events that meet the criteria you specified.
3. Type DE in the **C** column next to the appropriate ABACKUP event (AB appears in the **PGM** column) and press Enter to pop up a deletion confirmation window.
4. Type Y in the **REPLY** field on the confirmation window, and then press Enter.

Submitting an ABACKUP for an aggregate

Use the **ABARS Events** panel's **B** line command to submit an **ABACKUP** request on an aggregate.

Be aware that only one ABARS Manager function (either ABACKUP or ARECOVER) can be run on an aggregate at any one point in time.

1. Type 1 in the **Command** line on the **Backup and Recovery Management** panel, and then press Enter to display the **ABARS Events Search Setup** panel.
2. Use the **ABARS Events Search Setup** panel's online help to complete the fields that constitute the search criteria, and then press Enter to display the **ABARS Events** panel, which shows a list of all ABARS events that meet the criteria you specified.
3. Type B in the **C** column next to the appropriate aggregate's name and press Enter to display the **Submit Backup** panel.
4. Use the **Submit Backup** panel's online help to guide you as you make specifications in the fields on the **Submit Backup** panel. When your specifications are complete, press Enter.

About concurrent copy

DFSMShsm issues messages to the console regarding the status of the DFSMSdss Concurrent Copy operation. Message ARC6402I is issued when the Concurrent Copy session initialization is complete.

Message ARC6402I indicates the number of data sets for which Concurrent Copy initialization was successful or unsuccessful.

If all data sets to be backed up by DFSMSdss are eligible for Concurrent Copy, message ARC6402I is issued as soon as Concurrent Copy initialization is complete. At this point, for sites that do not have AUTO operations, a REXX EXEC can be used to submit a job to act as a trigger for the scheduler.

When the **CCTRIGGER** parameter in the BKMINI member of the product parameter library is set to Y, and the management class causes Concurrent Copy, ABARS Manager intercepts the messages during DFSMSdss processing and automatically issues the REXX EXEC at 'commit'.

A sample REXX EXEC is provided in member BKMCCTRG of the product parameter library. You can modify this any way you like. It is just a sample. If you want to create separate job names for separate aggregates, you must add REXX logic. Both the aggregate name and a status parameter are passed to the REXX EXEC. The BKMABARS procedure, which HSM uses to launch an ABARS **ABACKUP**, has a SYSEXEC DD from which the REXX can be launched.

If any data set is not eligible for Concurrent Copy, message ARC6402I is issued only when the entire DFSMSdss backup is physically complete, which could be several hours later. ABARS Manager provides several **ABACKUP** parameters to enhance this process. If a failure occurs during Concurrent Copy processing and these **ABACKUP** commands have been provided, the following occurs:

- Continue – Concurrent Copy failure is to be ignored and continue processing.
- WTOR – A message will inform the console operator of the problem and the appropriate course of action.
- ABEND – The task is to stop the backup for this aggregate and the job is to ABEND as soon as possible.

Because the use of Concurrent Copy for an aggregate backup is controlled by the SMS management class, if you need to change the Concurrent Copy options for an aggregate, you must change the management class definition for the aggregate and reactivate the updated SMS configuration, unless you are using ABARS Manager. ABARS Manager provides additional flexibility in the use of Concurrent Copy. You can temporarily and dynamically disable the use of Concurrent Copy for an entire aggregate or for only the backup of the activity log and instruction data set.

Changing an aggregate's IDS information

You can use the **ABARS Events** panel's **C** line command to change the information stored for an aggregate in the IDS. You can also use the **I** line command to view aggregate information from both ABARS Manager and SMS.

1. Type 1 in the **Command** line on the **Backup and Recovery Management** panel, and then press Enter to display the **ABARS Events Search Setup** panel.
2. Use the **ABARS Events Search Setup** panel's online help to complete the fields that constitute the search criteria, and then press Enter to display the **ABARS Events** panel, which shows a list of all ABARS events that meet the criteria you specified.
3. Type the appropriate line command in the **C** column next to the appropriate aggregate's name and press Enter to display the **Aggregate Controls** panel:
   - Type C to display the **Aggregate Controls** panel, where you can make the necessary changes.
   - Type I to display the Aggregate Inquiry panel, where you can view aggregate information from both ABARS Manager and SMS.
4. Use the online help for the **Aggregate Controls** or **Aggregate Inquiry** panel to guide you as you modify the information shown on that panel.
5. Save your changes.

Monitoring ABACKUPs and ARECOVERs as they execute

Use the **ABARS Events** panel's **M** line command to monitor ABACKUPs or ARECOVERs as they execute and to associate the ABARS started task with the actual aggregate and job executing it.

1. Type 1 in the **Command** line on the **Backup and Recovery Management** panel, and then press Enter to display the **ABARS Events Search Setup** panel.
2. Use the **ABARS Events Search Setup** panel's online help to complete the fields that constitute the search criteria, and then press Enter to display the **ABARS Events** panel, which shows a list of all ABARS events that meet the criteria you specified.
3. Type M in the **C** column next to the appropriate aggregate's name and press Enter to display the **ABARS Monitor** panel.
4. Use the **ABARS Monitor** panel's online help to guide you as you complete the input fields and use primary commands, and then press Enter.

Searching for ABACKUP data set overlap

Use the **ABARS Events** panel's **0** line command to identify data sets that are included in multiple ABACKUP events. You can search for overlap by event or by data set name.

1. Type 1 in the **Command** line on the **Backup and Recovery Management** panel, and then press Enter to display the **ABARS Events Search Setup** panel.
2. Use the **ABARS Events Search Setup** panel's online help to complete the fields that constitute the search criteria, and then press Enter to display the **ABARS Events** panel, which shows a list of all ABARS events that meet the criteria you specified.
3. If you want to search for overlap by data set name, go directly to step 4. Otherwise, to search for overlap by event, type 0 in the **C** column next to the appropriate aggregate's name on the **ABARS Events** panel and press Enter to display the **Overlap Search by Event** panel, where you can determine which data sets appear in more than one aggregate.
4. If you want to search for overlap by data set name, do the following:
   a) Type N in the **C** column next to the appropriate aggregate's name on the **ABARS Events** panel and press Enter to display the **View Data Set Name List** panel, which displays a list of the data sets included in that aggregate.
   b) From the list on the lower half of the **View Data Set Name List** panel, type 0 in the **C** column next to a data set's name, and then press Enter to search the IDS for other aggregates that have backed up the selected data set.

Submitting a non-incremental ARECOVER

Use the **ABARS Events** panel's **R** line command to submit a non-incremental ARECOVER for an ABACKUP event.

Only one ARECOVER or ABACKUP can be run against a given aggregate at any one point in time.

1. Type 1 in the **Command** line on the **Backup and Recovery Management** panel, and then press Enter to display the **ABARS Events Search Setup** panel.
2. Use the **ABARS Events Search Setup** panel's online help to complete the fields that constitute the search criteria, and then press Enter to display the **ABARS Events** panel, which shows a list of all ABARS events that meet the criteria you specified.
3. Type R in the **C** column next to the appropriate ABACKUP event (the **PGM** field has a value of AB and the **Type** field has a value of B) and press Enter to display the **Submit Recovery** panel.
4. Use the **Submit Recovery** panel's online help to guide you as you complete the input fields and use primary commands, and then press Enter. Specifications in the following fields cause additional panels or pop-up windows to appear when you press Enter on the **Submit Recovery** panel:
   a) If you specified Y in the **Recover Specific Data Sets** field on the **Submit Recovery** panel, the **View Data Set Name List** panel appears. Type R in the **C** column next to the data sets you want included in the recovery; type X in the **C** column next to the data sets you want excluded from the recovery.

b) If you specified Y in the **RECOVERNEWNAMELEVEL** field on the **Submit Recovery** panel, use the **New HLQ** fields to specify new high-level qualifiers for the corresponding old high-level qualifiers.

c) If you specified Y in the **RECOVERNEWNAMEALL** field on the **Submit Recovery** panel, use the **Original DSN/Mask** and **New DSN/Mask** fields to rename all data sets to be recovered using the new high level qualifier specified for **HLQ**.

Submitting an incremental ARECOVER

Use the **ABARS Events** panel's **R** line command to submit an incremental ARECOVER for an ABACKUP event.

Only one ARECOVER or ABACKUP can be run against a given aggregate at any one point in time.

1. Type 1 in the **Command** line on the **Backup and Recovery Management** panel, and then press Enter to display the **ABARS Events Search Setup** panel.
2. Use the **ABARS Events Search Setup** panel's online help to complete the fields that constitute the search criteria, and then press Enter to display the **ABARS Events** panel, which shows a list of all ABARS events that meet the criteria you specified.
3. Type R in the **C** column next to the appropriate ABACKUP event (the **PGM** field has a value of AB and the **Type** field has a value of I) and press Enter to display the **Check Recovery of Incremental** panel.
4. Use the **Check Recovery of Incremental** panel's online help to complete the fields on that panel and to select the generation at which you want the incremental recovery to start, and then press Enter to display the **Submit Recovery** panel.
5. Use the **Submit Recovery** panel's online help to guide you as you complete the input fields and use primary commands, and then press Enter. Specifications in the following fields cause additional panels or pop-up windows to appear when you press Enter on the **Submit Recovery** panel:

   a) If you specified Y in the **Recover Specific Data Sets** field on the **Submit Recovery** panel, the **View Data Set Name List** panel appears. Type R in the **C** column next to the data sets you want included in the recovery; type X in the **C** column next to the data sets you want excluded from the recovery.

   b) If you specified Y in the **RECOVERNEWNAMELEVEL** field on the **Submit Recovery** panel, use the **New HLQ** fields to specify new high-level qualifiers for the corresponding old high-level qualifiers.

   c) If you specified Y in the **RECOVERNEWNAMEALL** field on the **Submit Recovery** panel, use the **Original DSN/Mask** and **New DSN/Mask** fields to rename all data sets to be recovered using the new high level qualifier specified for **HLQ**.

About using the FullRename feature with ARECOVER

Native ABARS ARECOVER allows only a limited set of data set rename capabilities. In fact, you can change data set high level qualifiers only using ABARS parameters **RECOVERNEWNAMELEVEL** or **RECOVERNEWNAMEALL**. ABARS Manager offers a more robust rename capability. With ABARS Manager, you can rename data sets in almost any way.

However, you cannot use the ABARS Manager FullRename option to rename the following types of data sets:

- ICF catalogs
- ACCOMPANY data sets
- Tape data sets
- Migrated VSAM

ABARS Manager allows users to restore data sets to new names without some of the restrictions imposed by native ABARS. ABARS Manager allows any number of data sets to be renamed, and qualifiers other than the first may be changed. A full selection criterion allows changing any qualifier in any individual data set or group of data sets, or replacement of the entire data set name if desired.

After the ABARS ARECOVER event is complete, and the original data sets first restored, ABARS Manager produces a file containing a series of IDCAMS ALTER control statements. You can set keyword values in the BKMINI member of the product PARMLIB to specify whether IDCAMS will be invoked automatically

by ABARS Manager or you want to submit a subsequent IDCAMS ALTER job step to complete the rename operation.

FullRename restrictions

ABARS Manager Data Set Rename Extensions are subject to limitations, most based on DSS restrictions and some that are due to restrictions in native ABARS processing.

Data Set Rename limitations:

- You cannot rename ICF catalogs.
- Because they are on tape, you cannot rename data sets in the ACCOMPANY list.
- You cannot rename tape data sets.
- Migrated data sets are recalled to rename them. You cannot rename migrated VSAM data sets because they are known to DFSMShsm as non-VSAM by the cluster name. The data and index names are unknown.
- Each qualifier in the filter fully replaces a qualifier in the original DSN.
- The FullRename function is accomplished after the ARECOVER has completed by using IDCAMS ALTER statements. If you try to rename a data set to one that is already cataloged, the rename fails.
- Because ABARS Manager Data Set Rename processing renames data sets only after they have been recovered by ARECOVER, the RECOVERNEWNAMEALL or RECOVERNEWNAMELEVEL parameters might be necessary in order to restore the data sets using a temporary, or intermediate, high level qualifier.
- No check is made to determine whether the same data set name is to be created multiple times by the renaming criteria the user specifies. If two or more data sets are to be renamed to the same output name, only the first renaming succeeds.
- The Output mask cannot have a partially qualified node. For example, 'PROD.A*.**' is not allowed.
- You can rename a GDS by changing multiple qualifiers only *after the new GDG base has been defined*. Otherwise, the renamed GDS is converted to a non-GDS PS file.

User catalog and SMS restrictions

If you are renaming a previously recovered data set by changing its high level qualifier value, the final renamed data set could be re-cataloged in a different user catalog other than the original data set name. In this case, the rename function succeeds, but the user data set is not accessible from the original catalog. In addition, certain installation SMS restrictions might prevent a complete renaming of data set entries. With VSAM, the rename fails if the value specified for the **Rename to DSN/Mask** field is in a catalog other than the catalog of the value specified in the **Original DSN/Mask** field.

Usage tips

Because the FullRename function is accomplished after the ARECOVER has completed by using IDCAMS ALTER statements, restore the data sets first. You can use either ARECOVER with the REPLACE command, or ABARS RECOVERNEWNAMEALL or RECOVERNEWNAMELEVEL except for VSAM files.

Ensure the target data sets do not exist on the system unless they are GDG data sets and the **RENAME_GDG_DELETE** or **RENAME_GDG_SCRUB** parameters are set to Y in the product's BKMINI member of the product parameter library.

Read all the rules and ensure all data is appropriately planned for to ensure a successful FullRename.

Filter rules for the original data set name mask

The data set name mask you specify in the **Original DSN/Mask** field selects the data sets against which to apply the new data set name filter mask (as specified in the **Rename to DSN/Mask** field). The allowable filter characters are:

Mask Character	Interpretation
*	A single asterisk represents exactly one DSN qualifier of any value. * combined with valid DSN characters or % means 0 to nn characters of any value. For example, *. or .*. or .* .
**	A double asterisk represents 0 to nn DSN qualifiers of any value. ** may only be used once in a mask. For example, **. or .**. or .** are valid masks; **.abcd.** is not valid.
%	A percent sign represents one non-blank character.
!	An exclamation point represents one national character: @, #, or $.
<	A less-than sign represents one nonnumeric character, national symbols included.
>	A greater-than sign represents one numeric character.

For example, an original data set name mask of *.PAYROLL*.%%%%%23*.**.DATA would match DSN=TLQ050.PAYROLL.CYCLE23.YEAREND.DATA.

Filter rules for the new data set name mask

The data set name mask you specify in the **Rename to DSN/Mask** field renames the data sets selected by the original data set name filter mask (as specified in the **Original DSN/Mask** field). The allowable filter characters are:

Mask Character	Interpretation
*	A single asterisk represents exactly one DSN qualifier. You cannot use * as a partial qualifier in the new data set name mask. For example, aa.*bb.** is not a valid new data set name mask.
**	A double asterisk represents 0 to nn DSN qualifiers. You can specify the double asterisk only once in a mask. For example, **.abc.** is not a valid new data set name mask.
%	A percent sign represents one non-blank character.
+CCCCCC	A plus sign followed by one to eight characters means 'insert this new qualifier.'
-	A minus sign means 'remove this qualifier from the new name.'

For example, if the original data set name mask is ACME.LAB9.DEMO1.SDS.ORIG, and the new data set name mask is ACME.+TEST.XX%%.-.**, the renamed data set name would be ACME.TEST.XXB9.SDS.ORIG.

Examples

This example shows the renaming of a single fully qualified data set:

```
FULLRENAMEMASKS(AN.OLD.DATA.SET,A.NEW.DATA.SET)
```

If the original data set name mask is AN.OLD.DATA.SET and the new data set name mask is A.NEW.DATA.SET, renaming AN.OLD.DATA.SET results in a data set named A.NEW.DATA.SET.

This example shows the renaming of any three-qualifier DSN having the mid-level qualifier PAY to a new name that begins with ACME:

```
FULLRENAMEMASKS(*.PAY.*,ACME.PAY.*)
```

The rename results are as follows:

```
NONVSAM DS: BKM.PAY.TEST1
             >> MATCHING FILTER:  *.PAY.*
             >> NEW NAME FILTER:  ACME.PAY.*
```

```
             >> NEW NAME       :  ACME.PAY.TEST1
NONVSAM DS: BKM.PAY.TEST2
             >> MATCHING FILTER:  *.PAY.*
             >> NEW NAME FILTER:  ACME.PAY.*
             >> NEW NAME       :  ACME.PAY.TEST2
NONVSAM DS: BKM.PAY.TEST3
             >> MATCHING FILTER:  *.PAY.*
             >> NEW NAME FILTER:  ACME.PAY.*
             >> NEW NAME       :  ACME.PAY.TEST3
NONVSAM DS: BKM.PAY.TEST4
             >> MATCHING FILTER:  *.PAY.*
             >> NEW NAME FILTER:  ACME.PAY.*
             >> NEW NAME       :  ACME.PAY.TEST4
```

This example shows the renaming of data sets beginning with BKM and ending with MAY so that they end with JUNE:

```
FULLRENAMEMASKS(BKM.**.MAY,**.JUNE)
```

The rename results are as follows:

```
NONVSAM DS: BKM.QUAL1.A.MAY
             >> MATCHING FILTER:  BKM.**.MAY
             >> NEW NAME FILTER:  **.JUNE
             >> NEW NAME       :  BKM.QUAL1.A.JUNE
NONVSAM DS: BKM.QUAL1.MAY
             >> MATCHING FILTER:  BKM.**.MAY
             >> NEW NAME FILTER:  **.JUNE
             >> NEW NAME       :  BKM.QUAL1.JUNE
NONVSAM DS: BKM.QUAL1.QUAL2.MAY
             >> MATCHING FILTER:  BKM.**.MAY
             >> NEW NAME FILTER:  **.JUNE
             >> NEW NAME       :  BKM.QUAL1.QUAL2.JUNE
NONVSAM DS: BKM.QUAL2.MAY
             >> MATCHING FILTER:  BKM.**.MAY
             >> NEW NAME FILTER:  **.JUNE
             >> NEW NAME       :  BKM.QUAL2.JUNE
```

This example shows the renaming of any three-qualifier data set with XYZ in the first three characters of the first qualifier, and DATA as the low level qualifier, to IVP as the high level qualifier, and DATA2 as the last qualifier:

```
FULLRENAMEMASKS(XYZ*.*.DATA,IVP.*.DATA2)
```

The results are as follows:

```
NONVSAM DS: XYZI.GEN1.DATA
             >> MATCHING FILTER:  XYZ*.*.DATA
             >> NEW NAME FILTER:  IVP.*.DATA2
             >> NEW NAME       :  IVP.GEN1.DATA2
NONVSAM DS: XYZI.GEN2.DATA
             >> MATCHING FILTER:  XYZ*.*.DATA
             >> NEW NAME FILTER:  IVP.*.DATA2
             >> NEW NAME       :  IVP.GEN2.DATA2
NONVSAM DS: XYZI.GEN3.DATA
             >> MATCHING FILTER:  XYZ*.*.DATA
             >> NEW NAME FILTER:  IVP.*.DATA2
             >> NEW NAME       :  IVP.GEN3.DATA2
NONVSAM DS: XYZI.GEN4.DATA
             >> MATCHING FILTER:  XYZ*.*.DATA
             >> NEW NAME FILTER:  IVP.*.DATA2
             >> NEW NAME       :  IVP.GEN4.DATA2
```

This example shows the removal of ABC.DEF from a data set name:

```
FULLRENAMEMASKS(XYZI.ABC.DEF.TEST,XYZI.-.-.TEST)
```

The results are as follows:

```
NONVSAM DS: XYZI.D.ABC.DEF.TEST
             >> MATCHING FILTER:  XYZI.D.*.*.TEST
             >> NEW NAME FILTER:  XYZI.D.-.-.TEST
             >> NEW NAME       :  XYZI.D.TEST
```

This example adds the NEW qualifier to each data set name:

```
FULLRENAMEMASKS(XYZI.PAY.JUNE,XYZI.+NEW.PAY.**)
```

The results are as follows:

```
NONVSAM DS:  XYZI.PAY.JUNE
              >> MATCHING FILTER:  XYZI.PAY.JUNE
              >> NEW NAME FILTER:  XYZI.+NEW.PAY.**
              >> NEW NAME       :  XYZI.NEW.PAY.JUNE
```

Here is another example of adding the NEW qualifier to each data set name:

```
FULLRENAMEMASKS(XYZI.%.APRIL,XYZI.%.+NEW.APRIL)
```

The results are as follows:

```
NONVSAM DS:  XYZI.B.APRIL
              >> MATCHING FILTER:  XYZI.%.APRIL
              >> NEW NAME FILTER:  XYZI.%.+NEW.APRIL
              >> NEW NAME       :  XYZI.B.NEW.APRIL
```

This example shows the use of an ambiguous filter, resulting in a failed rename:

```
FULLRENAMEMASKS(OLD.DATA.SET,NEW.**.SET.**)
```

The results of this ambiguous filter are as follows:

```
BKMB1313E NEW NAME MASK IS AMBIGUOUS :  NEW.**.SET.**
BKMB1414E IDCAMS RENAME NOT DONE - ERRORS OR WARNINGS FOUND
```

This example shows the renaming of a GDS by adding two new qualifiers:

```
FULLRENAMEMASKS(A.B.C.D.G0001V00,A.+NEWQUAL.%.%.%.+NEW.*)
```

If the original data set name mask is A.B.C.D.G0001V00 and the new data set name mask is A.+NEWQUAL.%.%.%.+NEW.*, renaming A.B.C.D.G0001V00 results in a new GDS named A.NEWQUAL.B.C.D.NEW.G0001V00.

Using FullRename for VSAM clusters

If an ICF VSAM cluster is renamed, all components of the cluster that match your specification in the **Original DSN/Mask** field are renamed using the value specified in the **Rename to DSN/Mask** field for the cluster. For more information about generated names, see the IBM publication *DFSMS/MVS Using Data Sets*.

When you use the FullRename masks for VSAM data sets, note the following:

- If the value specified for **Rename to DSN/Mask** for a VSAM cluster or for the **Rename to DSN/Mask** for any of its components already exists, no alters are generated for the cluster or for its components.
- New names for any AIXs cannot be resolved from the new name of the base cluster. They must be filtered (**Original DSN/Mask**, **Rename to DSN/Mask**) in the same way as the base cluster, and their components are renamed in the same way.

Here is an example in which a VSAM sphere is rename. Keep in mind that when you rename a VSAM data set, the new name for all components, excluding any AIXs, is the same as the cluster name, with the last node of the component appended.

```
Cluster:    A.VSAM.DATASET
Data:        A.VSAM.DATASET.DATA
Index:        A.VSAM.DATASET.INDX
Path:        A.VSAM.DATASET.PATH1
```

If, however, the last node of the new name for the cluster is CL or CLUS, the new name for the cluster, minus the last node, will be used to rename the components. Using the VSAM data set above, if the cluster has a new name of A.TEST.VSAM.DATASET.CL, the component new names are as follows:

```
Cluster:    A.TEST.VSAM.DATASET.CL
Data:        A.TEST.VSAM.DATASET.DATA
Index:      A.TEST.VSAM.DATASET.INDX
Path:        A.TEST.VSAM.DATASET.PATH1
```

The following example shows a single fully qualified VSAM cluster being renamed:

```
FULLRENAMEMASKS(XYZI.XAA.PROD.AUG,PROD.XAA.PROD.AUG)
```

The results are as follows:

```
VSAM SPHERE:  XYZI.XAA.PROD.AUG
              >> MATCHING FILTER:  XYZI.XAA.PROD.AUG
              >> NEW NAME FILTER:  PROD.XAA.PROD.AUG
              >> NEW NAME       :  PROD.XAA.PROD.AUG
  DATA :      XYZI.XAA.PROD.AUG.DATA
              >> NEW NAME       :  PROD.XAA.PROD.AUG.DATA
  INDEX:      XYZI.XAA.PROD.AUG.INDEX
              >> NEW NAME       :  PROD.XAA.PROD.AUG.INDEX
```

This example shows the qualifier XAC being removed as a single VSAM cluster ending with CL is renamed:

```
FULLRENAMEMASKS(XYZI.XAC.PROD.AUG.CL,XYZI.-.PROD.AUG.CL)
```

The results are as follows:

```
VSAM SPHERE: XYZI.XAC.PROD.AUG.CL
               >> MATCHING FILTER:  XYZI.XAC.PROD.AUG.CL
               >> NEW NAME FILTER:  XYZI.-.PROD.AUG.CL
               >> NEW NAME       :  XYZI.PROD.AUG.CL
   DATA :      XYZI.XAC.PROD.AUG.CL.DATA
               >> NEW NAME       :  XYZI.PROD.AUG.DATA
   INDEX:      XYZI.XAC.PROD.AUG.CL.INDEX
               >> NEW NAME       :  XYZI.PROD.AUG.INDEX
```

This example shows the qualifier NEW being added to the data set name:

```
FULLRENAMEMASKS(XYZI.XAB.PROD.AUG,XYZI.+NEW.XAB.PROD.AUG)
```

The results are as follows:

```
VSAM SPHERE:  XYZI.XAB.PROD.AUG
              >> MATCHING FILTER:  XYZI.XAB.PROD.AUG
              >> NEW NAME FILTER:  XYZI.+NEW.XAB.PROD.AUG
              >> NEW NAME       :  XYZI.NEW.XAB.PROD.AUG
  DATA :      XYZI.XAB.PROD.AUG.DATA
              >> NEW NAME       :  XYZI.NEW.XAB.PROD.AUG.DATA
  INDEX:      XYZI.XAB.PROD.AUG.INDEX
              >> NEW NAME       :  XYZI.NEW.XAB.PROD.AUG.INDEX
```

This example shows all data sets with a high level qualifier of BKM renamed to a high level qualifier of PROD.

Note: This rename consists of both VSAM and non-VSAM data sets.

```
FULLRENAMEMASKS(BKM.**,PROD.**)
```

The results are as follows:

```
NONVSAM DS:  BKM.GEN1.DATA
              >> MATCHING FILTER:  BKM.**
              >> NEW NAME FILTER:  PROD.**
              >> NEW NAME       :  PROD.GEN1.DATA
NONVSAM DS:  BKM.PAY.TEST4
              >> MATCHING FILTER:  BKM.**
              >> NEW NAME FILTER:  PROD.**
              >> NEW NAME       :  PROD.PAY.TEST4
VSAM SPHERE:  BKM.PROD.AUG.CL
              >> MATCHING FILTER:  BKM.**
              >> NEW NAME FILTER:  PROD.**
              >> NEW NAME       :  PROD.PROD.AUG.CL
  DATA :      BKM.PROD.AUG.DATA
```

```
              >> NEW NAME       :  PROD.PROD.AUG.DATA
  INDEX:      BKM.PROD.AUG.INDEX
              >> NEW NAME       :  PROD.PROD.AUG.INDEX
VSAM SPHERE:  BKM.XAA.PROD.AUG
              >> MATCHING FILTER:  BKM.**
              >> NEW NAME FILTER:  PROD.**
              >> NEW NAME       :  PROD.XAA.PROD.AUG
  DATA :      BKM.XAA.PROD.AUG.DATA
              >> NEW NAME       :  PROD.XAA.PROD.AUG.DATA
  INDEX:      BKM.XAA.PROD.AUG.INDEX
              >> NEW NAME       :  PROD.XAA.PROD.AUG.INDEX
```

This example shows all data sets with a high level qualifier of BKM renamed to a high level qualifier of PROD.NEW.

Note: This rename consists of both VSAM and non-VSAM data sets.

```
FULLRENAMEMASKS(BKM.**,PROD.+NEW.**)
```

The results are as follows:

```
NONVSAM DS:  BKM.GEN1.DATA
              >> MATCHING FILTER:  BKM.**
              >> NEW NAME FILTER:  PROD.+NEW.**
              >> NEW NAME       :  PROD.NEW.GEN1.DATA
NONVSAM DS:  BKM.PAY.TEST4
              >> MATCHING FILTER:  BKM.**
              >> NEW NAME FILTER:  PROD.+NEW.**
              >> NEW NAME       :  PROD.NEW.PAY.TEST4
VSAM SPHERE:  BKM.PROD.AUG.CL
              >> MATCHING FILTER:  BKM.**
              >> NEW NAME FILTER:  PROD.+NEW.**
              >> NEW NAME       :  PROD.NEW.PROD.AUG.CL
  DATA :      BKM.PROD.AUG.DATA
              >> NEW NAME       :  PROD.NEW.PROD.AUG.DATA
  INDEX:      BKM.PROD.AUG.INDEX
              >> NEW NAME       :  PROD.NEW.PROD.AUG.INDEX
VSAM SPHERE:  BKM.XAA.PROD.AUG
              >> MATCHING FILTER:  BKM.**
              >> NEW NAME FILTER:  PROD.+NEW.**
              >> NEW NAME       :  PROD.NEW.XAA.PROD.AUG
  DATA :      BKM.XAA.PROD.AUG.DATA
              >> NEW NAME       :  PROD.NEW.XAA.PROD.AUG.DATA
  INDEX:      BKM.XAA.PROD.AUG.INDEX
              >> NEW NAME       :  PROD.NEW.XAA.PROD.AUG.INDEX
```

This example shows a batch ARECOVER using the FULLRENAME feature. Use a comma or a space to separate data set names within the keyword FULLRENAMEMASKS. The example shows a space being used to separate the data set names.

```
//STEPNAME  EXEC  BKMBPROC
//BKMBATCH.SYSIN  DD  *
    ARECOVER AG(IVPSTD)                                          +
             GEN(0)                                              +
             EXEC                                                +
             TERMMSGS(YES)                                       +
             NOSTACK                                             +
             INCLUDE(                                            +
                    XYZI.A.TEST1                                 +
                    XYZI.B.TEST2                                 +
                    XYZI.C.TEST3                                 +
                    XYZI.D.MINUS.MINUS.TEST4                     +
                    )                                            +
             FULLRENAMEMASKS(                                    +
              XYZI.A.TEST1      XYZI.+NEW.A.**                   +
              XYZI.%.TEST2      XYZI.%.+NEW.TEST2                +
              XYZI.C.TEST*      XYZI.X.**                        +
              XYZI.*.TEST4      XYZI.D.-.-.TEST4                 +
                              )                                  +
     FINALRENAMEACTION(EXEC)                                     +
     OUTPUTRENAMEDSN(                                            +
      &HSMUID.&HOSTID.&AGGNAME.&DATE.&TIME.&USERID               +
                             )
```

Submitting an ARECOVER using the FullRename feature

With ABARS Manager, you can rename data sets in almost any way as part of an ARECOVER.

1. Modify the settings in the BKMINI member of the product parameter library for the parameters **ALLOWFULLRENAME**, **OUTPUTRENAMEDSN**, **FINALRENAMEACTION**, and **RENAMEDSNDISP**.
2. Type 1 in the **Command** line on the **Backup and Recovery Management** panel, and then press Enter to display the **ABARS Events Search Setup** panel.
3. Use the **ABARS Events Search Setup** panel's online help to complete the fields that constitute the search criteria, and then press Enter to display the **ABARS Events** panel, which shows a list of all ABARS events that meet the criteria you specified.
4. Type R in the **C** column next to the appropriate ABACKUP event (the **PGM** field has a value of AB) and press Enter to display the **Submit Recovery** panel.
5. Type Y in the **FULLRENAMEMASKS** field, and then press Enter to display the **FULLRENAMEMASKS** panel.
6. Use the online help to guide you as you specify the appropriate information in the **Original DSN/Mask** and **Rename to DSN/Mask** fields.
   - If you specified EXEC for the FINALRENAMEACTION keyword in the BKMINI member of the product parameter library, the ALTER commands for the data set renames are executing directly by ABARS Manager without creating or using an output data set. If you change the value of **FINALRENAMEACTION** on this panel, the change is in effect for this one ARECOVER only.
   - If you specified FILE for the FINALRENAMEACTION keyword in the product parameter library's BKMINI member, the ALTER control statements are created and written to an output file, designated by the **OUTPUTRENAMEDSN** field's value, which by default uses these symbolic values: &HSMUID.&HOSTID.&AGGNAME.&DATE.&TIME.&.USERID. If you change the value of **OUTPUTRENAMEDSN** on this panel, the change is in effect for this one ARECOVER only.
7. Press PF3 to re-display the **Submit Recovery** panel. Use the **Submit Recovery** panel's online help to guide you as you make specifications on this panel.
8. If you specified EXEC as the **FINALRENAMEACTION**, press Enter to submit the ARECOVER. Otherwise, if you specified FILE as the **FINALRENAMEACTION**, add the IDCAMS program to the output file and then press Enter.

The following example shows the final contents of the ALTER file created by ABARS Manager during FullRename processing. Use this file as input to an IDCAMS step execution to perform the final data set rename operations.

```
ALTER  AB.KSDS NEWNAME(NEW.KSDSA)
ALTER  AB.KSDS.DATA NEWNAME(NEW.KSDSA.DATA)
ALTER  AB.KSDS.INDEX NEWNAME(NEW.KSDSA.INDEX)
ALTER  AB.SEQFILE NEWNAME(NEW.SEQFILE)
```

Submitting a RECOVER of ABARS 'D' file data using DFSMSdss

Use the **ABARS Events** panel's **RD** line command to submit a DFSMSdss RECOVER of ABARS D file data.

Only primary DASD data sets are eligible for this type of recovery. Controlling allocation specifics can be important for handling unusual conditions during recovery, such as the following:

- Directing an SMS data set to a specific volser – Guaranteed Space
- Bypassing SMS ACS routines and assigning different classes
- Directing an SMS managed data set to a non-SMS volume
- Bypassing TMS

Consider directing SMS data sets to handle specific situations during recovery.

For example, if a data set at the primary site was assigned a STORCLAS of guaranteed space and the same volser has not been initialized at the recovery site, you might need to add the DFSMSdss supported SMS Keyword STORCLAS to the recovery.

Consider another example in which the SMS ACS is set up to direct data set X to a specific pool. The pool is now full and data set X is very large. In this situation, you might want to direct data set X to a specific volume but still want it to be SMS managed. Use the STORCLAS parameter to direct data set X to a specific volser

Note:

- This feature is an advanced function for storage administrators or systems programmers. ABARS Manager does no syntax checking; it is your responsibility to verify that you have specified the appropriate keywords.
- Only one ABACKUP or ARECOVER can be run against an aggregate at any one point in time.

1. Type 1 in the **Command** line on the **Backup and Recovery Management** panel, and then press Enter to display the **ABARS Events Search Setup** panel.
2. Use the **ABARS Events Search Setup** panel's online help to complete the fields that constitute the search criteria, and then press Enter to display the **ABARS Events** panel, which shows a list of all ABARS events that meet the criteria you specified.
3. Type RD in the **C** column next to the appropriate ABACKUP event (the **PGM** field has a value of AB) and press Enter to display the **Submit Recovery** panel.
4. On the **Submit Recovery** panel, type Y in the **Recover Specific Data Sets** field and press Enter to display the **View Data Set Name List** panel.
5. On the **View Data Set Name List** panel, do the following:
   a) Type R in the **C** column next to every data set you want included in the recovery.
   b) Type X in the **C** column next to every data set you want excluded from the recovery.
   c) Press PF3 to re-display the **Submit Recovery** panel.
6. On the **Submit Recovery** panel, select the DFSMSdss keywords you want and type S in the **Build/View JCL for Submit** field to review the JCL and submit it.

Browsing an aggregate's instruction data set

You can use the **ABARS Events** panel's **BI** line command to browse the instruction data set for an aggregate.

Be aware that you cannot browse an instruction data set until it has been defined to the aggregate in ISMF, activated through SMS, and been cataloged.

1. Type 1 in the **Command** line on the **Backup and Recovery Management** panel, and then press Enter to display the **ABARS Events Search Setup** panel.
2. Use the **ABARS Events Search Setup** panel's online help to complete the fields that constitute the search criteria, and then press Enter to display the **ABARS Events** panel, which shows a list of all ABARS events that meet the criteria you specified.
3. On the **ABARS Events** panel, type BI in the **C** column next to the aggregate whose instruction data set you want to browse, and then press Enter to show that data set in an ISPF Browse panel.

Editing an aggregate's ARECOVER conflict data set

Use the **ABARS Events** panel's **EC** line command to edit a cataloged ARECOVER conflict data set for a selected aggregate.

1. Type 1 in the **Command** line on the **Backup and Recovery Management** panel, and then press Enter to display the **ABARS Events Search Setup** panel.
2. Use the **ABARS Events Search Setup** panel's online help to complete the fields that constitute the search criteria, and then press Enter to display the **ABARS Events** panel, which shows a list of all ABARS events that meet the criteria you specified.
3. On the **ABARS Events** panel, type EC in the **C** column next to the ARECOVER event (**PGM** column shows a value of AR) whose conflict data set you want to edit, and then press Enter.

If a conflict data set was created during the selected ARECOVER event and the conflict data set is cataloged, that data set is displayed in an ISPF Edit panel. If no conflict data set was created during the ARECOVER event or the conflict data set is not cataloged, the system issues a message to that effect.

4. On the ISPF Edit panel, make the appropriate changes and save them.

Browsing an aggregate's SDSL parameters

You can use the **ABARS Events** panel's **BL** line command to browse the instruction data set for an aggregate.

1. Type 1 in the **Command** line on the **Backup and Recovery Management** panel, and then press Enter to display the **ABARS Events Search Setup** panel.
2. Use the **ABARS Events Search Setup** panel's online help to complete the fields that constitute the search criteria, and then press Enter to display the **ABARS Events** panel, which shows a list of all ABARS events that meet the criteria you specified.
3. On the **ABARS Events** panel, type BL in the **C** column next to the aggregate whose SDSL parameters you want to browse, and then press Enter.

   If SDSL parameters exist for the selected aggregate, they appear in an ISPF Browse panel; otherwise, the message `NO SDSL PARMS EXIST` is issued.

Editing an aggregate's SDSL parameters

You can use the **ABARS Events** panel's **EL** line command to edit the SDSL parameters for an aggregate.

1. Type 1 in the **Command** line on the **Backup and Recovery Management** panel, and then press Enter to display the **ABARS Events Search Setup** panel.
2. Use the **ABARS Events Search Setup** panel's online help to complete the fields that constitute the search criteria, and then press Enter to display the **ABARS Events** panel, which shows a list of all ABARS events that meet the criteria you specified.
3. On the **ABARS Events** panel, type EL in the **C** column next to the aggregate whose SDSL parameters you want to edit, and then press Enter.

   If SDSL parameters exist for the selected aggregate, they appear in an ISPF Edit panel; otherwise, the message `NO SDSL PARMS EXISTS` is issued.

Browsing an aggregate's Selection Data Set

You can use the **ABARS Events** panel's **BS** line command to browse the instruction data set for an aggregate.

Be aware that you cannot browse a Selection Data Set until it has been defined to ISMF, activated through SMS, and cataloged.

1. Type 1 in the **Command** line on the **Backup and Recovery Management** panel, and then press Enter to display the **ABARS Events Search Setup** panel.
2. Use the **ABARS Events Search Setup** panel's online help to complete the fields that constitute the search criteria, and then press Enter to display the **ABARS Events** panel, which shows a list of all ABARS events that meet the criteria you specified.
3. On the **ABARS Events** panel, type BS in the **C** column next to the aggregate whose Selection Data Set you want to browse, and then press Enter.

   If there is a single Selection Data Set for the selected aggregate, that data set appears in an ISPF Browse panel. However, if there are multiple Selection Data Sets for the selected aggregate, the **View Data Set Name List** panel appears, listing the Selection Data Sets for the aggregate. Type B in the **C** column next to the Selection Data Set you want to browse, and then press Enter to open an ISPF Browse panel that shows the data set.

Editing an aggregate's Selection Data Set

You can use the **ABARS Events** panel's **ES** line command to open an ISPF Edit session on a Selection Data Set for an aggregate.

Be aware that before you can edit a Selection Data Set, that data set must be defined to ISMF, activated through SMS, and cataloged.

1. Type 1 in the **Command** line on the **Backup and Recovery Management** panel, and then press Enter to display the **ABARS Events Search Setup** panel.
2. Use the **ABARS Events Search Setup** panel's online help to complete the fields that constitute the search criteria, and then press Enter to display the **ABARS Events** panel, which shows a list of all ABARS events that meet the criteria you specified.
3. On the **ABARS Events** panel, type ES in the **C** column next to the aggregate whose Selection Data Set you want to edit, and then press Enter.

   If there is a single Selection Data Set for the selected aggregate, that data set appears in an ISPF Edit panel. However, if there are multiple Selection Data Sets for the selected aggregate, the View SDS(s) Used for Aggregate panel appears, listing the Selection Data Sets for the aggregate. Type E in the **C** column next to the Selection Data Set you want to edit, and then press Enter to open an ISPF Edit panel that shows the data set.
4. On the ISPF Edit panel, make the appropriate changes and save them.

Browsing an aggregate's cataloged ARECOVER restart data set

Use the **ABARS Events** panel's **BR** line command to browse a cataloged ARECOVER restart data set for a selected aggregate.

1. Type 1 in the **Command** line on the **Backup and Recovery Management** panel, and then press Enter to display the **ABARS Events Search Setup** panel.
2. Use the **ABARS Events Search Setup** panel's online help to complete the fields that constitute the search criteria, and then press Enter to display the **ABARS Events** panel, which shows a list of all ABARS events that meet the criteria you specified.
3. On the **ABARS Events** panel, type BR in the **C** column next to the ARECOVER event (**PGM** column shows a value of AR) whose restart data set you want to browse, and then press Enter.

   If a restart data set was created during the selected ARECOVER event and the restart data set is cataloged, that data set is displayed in an ISPF Browse panel. If no restart data set was created during the ARECOVER event or the restart data set is not cataloged, the system issues a message to that effect.

Editing an aggregate's cataloged ARECOVER restart data set

Use the **ABARS Events** panel's **ER** line command to edit a cataloged ARECOVER restart data set for a selected aggregate.

1. Type 1 in the **Command** line on the **Backup and Recovery Management** panel, and then press Enter to display the **ABARS Events Search Setup** panel.
2. Use the **ABARS Events Search Setup** panel's online help to complete the fields that constitute the search criteria, and then press Enter to display the **ABARS Events** panel, which shows a list of all ABARS events that meet the criteria you specified.
3. On the ABARS Events panel, type ER in the **C** column next to the ARECOVER event (**PGM** column shows a value of AR) whose restart data set you want to edit, and then press Enter.

   If a restart data set was created during the selected ARECOVER event and the restart data set is cataloged, that data set is displayed in an ISPF Edit panel. If no restart data set was created during the ARECOVER event or the restart data set is not cataloged, the system issues a message to that effect.
4. On the ISPF Edit panel, make the appropriate changes and save them.

Browsing an aggregate's ARECOVER conflict data set

Use the **ABARS Events** panel's **BC** line command to browse a cataloged ARECOVER restart data set for a selected aggregate.

1. Type 1 in the **Command** line on the **Backup and Recovery Management** panel, and then press Enter to display the **ABARS Events Search Setup** panel.
2. Use the **ABARS Events Search Setup** panel's online help to complete the fields that constitute the search criteria, and then press Enter to display the **ABARS Events** panel, which shows a list of all ABARS events that meet the criteria you specified.
3. On the **ABARS Events** panel, type BC in the **C** column next to the ARECOVER event (**PGM** column shows a value of AR) whose conflict data set you want to browse, and then press Enter.

   If a conflict data set was created during the selected ARECOVER event and the conflict data set is cataloged, that data set is displayed in an ISPF Browse panel. If no conflict data set was created during the ARECOVER event or the conflict data set is not cataloged, the system issues a message to that effect.

Editing an aggregate's ARECOVER conflict data set

Use the **ABARS Events** panel's **EC** line command to edit a cataloged ARECOVER conflict data set for a selected aggregate.

1. Type 1 in the **Command** line on the **Backup and Recovery Management** panel, and then press Enter to display the **ABARS Events Search Setup** panel.
2. Use the **ABARS Events Search Setup** panel's online help to complete the fields that constitute the search criteria, and then press Enter to display the **ABARS Events** panel, which shows a list of all ABARS events that meet the criteria you specified.
3. On the **ABARS Events** panel, type EC in the **C** column next to the ARECOVER event (**PGM** column shows a value of AR) whose conflict data set you want to edit, and then press Enter.

   If a conflict data set was created during the selected ARECOVER event and the conflict data set is cataloged, that data set is displayed in an ISPF Edit panel. If no conflict data set was created during the ARECOVER event or the conflict data set is not cataloged, the system issues a message to that effect.
4. On the ISPF Edit panel, make the appropriate changes and save them.

About logical aggregate management

Logical aggregate management is a way to manage or prioritize a group of aggregates by using a single ABACKUP or a single ARECOVER command. A logical aggregate is a logical name that groups, and can be used to manage physical aggregates.

Often, multiple aggregates need to be grouped for tracking or prioritizing recovery. Logical Aggregate Management can help prioritize critical aggregate recovery by creating a logical group called CRITICAL1 for example. All critical applications can be assigned to this logical aggregate and only the Logical Aggregate needs to be submitted for recovery. It in turn submits the ARECOVERs for all the physical aggregates defined to this Logical Aggregate, based on the number of ABARS address spaces available.

This feature helps you prioritize your recovery so that anyone can submit the logical aggregate. No longer will you need to wade through a list of applications at your recovery site to determine which are the most critical, and then submit them individually. Any group of aggregates can be logically grouped together within ABARS Manager, and can be monitored, tracked, executed and managed from that single, overall point of view.

Accessing the logical aggregate management feature

You can access the logical aggregate set-up and management panels from the Advanced Backup and Recovery for z/OS **Main Menu** panel.

The **Logical Aggregate Management Setup** panel appears the first time you select the Logical Aggregates option from the **Backup and Recovery Management** panel. If you specify / in the **Don't show this panel again** field, the Advanced Backup and Recovery for z/OS bypasses the setup panel on subsequent access to the **Logical Aggregate Management** panel.

1. Type 2 in the **Command** line on the **Main Menu** panel, and then press Enter to display the **Backup and Recovery Management** panel.
2. Type 2 in the **Command** line on the **Backup and Recovery Management** panel, and then press Enter to display the **Logical Aggregate Management Setup** panel.
3. Do either of the following:
   - If you want to create a filtered list of logical aggregates upon which you want to perform management tasks, use the **Logical Aggregate Management Setup** panel's online help to guide you as you type entries in the fields, and then press Enter to display the **Logical Aggregate Management** panel.
   - If you want to display a list of all logical aggregates, press Enter to display the **Logical Aggregate Management** panel.

Creating a logical aggregate

You can create a new logical aggregate from either the **Logical Aggregate Management Setup** panel or from the **Logical Aggregate Management** panel.

1. Do either of the following:
   - From the **Logical Aggregate Management Setup** panel, type Y in the **Add NEW Logical Aggregate** field and then press Enter to pop up the **Create Logical Aggregate** window.
   - From the **Logical Aggregate Management** panel, type ADD in the **Command** line and then press Enter to pop up the **Create Logical Aggregate** window.
2. On the **Create Logical Aggregate** window, do the following:
   a) In the **Specify Logical Aggregate** field, type a valid name for the new logical aggregate. The name you specify cannot exceed 30 characters in length and cannot be the same as any existing physical or logical aggregate.
   b) Press Enter to display the **Add Aggregate to Logical Aggregate** panel, which shows a list of physical aggregates.
3. On the **Add Aggregate to Logical Aggregate** panel, do the following:
   a) Type S in the **C** column next to the name of each aggregate you want included in your new logical aggregate.
   b) When you have selected all the aggregates you want included in the logical aggregate, type SAVE in the **Command** line and press Enter.

   The new logical aggregate is saved to the IDS.

Adding an aggregate to a logical aggregate

You can use the **Logical Aggregate Management** panel's **A** line command to add a physical aggregate to a logical aggregate definition.

1. Type 2 in the **Command** line on the **Main Menu** panel, and then press Enter to display the **Backup and Recovery Management** panel.
2. Type 2 in the **Command** line on the **Backup and Recovery Management** panel, and then press Enter to display the **Logical Aggregate Management Setup** panel.
3. Do either of the following:

- If you want to create a filtered list of logical aggregates upon which you want to perform management tasks, use the **Logical Aggregate Management Setup** panel's online help to guide you as you type entries in the fields, and then press Enter to display the **Logical Aggregate Management** panel.
- If you want to display a list of all logical aggregates, press Enter to display the **Logical Aggregate Management** panel.

4. On the **Logical Aggregate Management** panel, type A in the **C** column next to the name of the logical aggregate to which you want to add a physical aggregate, and then press Enter to display the **Add Aggregate to Logical Aggregate** panel.
5. On the **Add Aggregate to Logical Aggregate** panel, type S in the **C** column next to each aggregate you want added to the selected logical aggregate.
6. Type SAVE in the **Command** line, and then press Enter.

Deleting a logical aggregate

You can use the **Logical Aggregate Management** panel's **DL** line command to remove a logical aggregate definition from ABARS Manager.

Deleting a logical aggregate does not delete any physical aggregates from either ABARS Manager or from ISMF.

1. Type 2 in the **Command** line on the **Main Menu** panel, and then press Enter to display the **Backup and Recovery Management** panel.
2. Type 2 in the **Command** line on the **Backup and Recovery Management** panel, and then press Enter to display the **Logical Aggregate Management Setup** panel.
3. Do either of the following:
   - If you want to create a filtered list of logical aggregates from which you can select the ones you want to delete, use the **Logical Aggregate Management Setup** panel's online help to guide you as you type entries in the fields, and then press Enter to display the **Logical Aggregate Management** panel.
   - If you want to display a list of all logical aggregates, press Enter to display the **Logical Aggregate Management** panel.
4. On the **Logical Aggregate Management** panel, type DL in the **C** column for the logical aggregate you want to delete, and then press Enter to display the **Delete Logical Aggregate** pop-up window.
5. On the **Delete Logical Aggregate** pop-up window, type Y in the **REPLY** field and press Enter.

Deleting a physical aggregate from a logical aggregate

You can use the **Logical Aggregate Management** panel's **DA** line command to remove a physical aggregate from a logical aggregate.

1. Type 2 in the **Command** line on the **Main Menu** panel, and then press Enter to display the **Backup and Recovery Management** panel.
2. Type 2 in the **Command** line on the **Backup and Recovery Management** panel, and then press Enter to display the **Logical Aggregate Management Setup** panel.
3. Do either of the following:
   - If you want to create a filtered list of logical aggregates upon which you want to perform management tasks, use the **Logical Aggregate Management Setup** panel's online help to guide you as you type entries in the fields, and then press Enter to display the **Logical Aggregate Management** panel.
   - If you want to display a list of all logical aggregates, press Enter to display the **Logical Aggregate Management** panel.
4. On the **Logical Aggregate Management** panel, type DA in the **C** column for the physical aggregate you want to delete from the logical aggregate, and then press Enter to display the **Remove Aggregate from List** pop-up window.

5. On the **Remove Aggregate from List** pop-up window, type Y in the **REPLY** field and press Enter.

Selection Data Set language (SDSL)

Selection Data Set Language (SDSL) is a powerful language facility that enhances the native ABARS selection criteria of INCLUDE, EXCLUDE, ALLOCATE, and ACCOMPANY.

Selection Data Set Language provides nearly 30 additional attributes. Use file name, management class, storage group, DSORG, LRECL, RECFM, migration status, change date, and so on, to include or exclude data. SDSL also supports extended masking of data set names and predefined filter groups. Predefined filter groups are provided as a convenience and maintenance enhancement so that common filtering done in many different aggregates can be defined once and used where required.

Support in SDSL for the Boolean operators AND, OR, XOR, and NOT permits complex conditional logic. By using SDSL, you get dynamic, real-time analysis of data sets for selection in an aggregate. Based on the parameters you specify, the results update the Selection Data Set prior to backup processing with a list of specific data sets matching the coded criteria. Use this feature to standardize frequent changes to your installation's data sets by automatically examining them each time an **ABACKUP EXECUTE** or **CONVSDSL** command is executed to include the matching file names in an aggregate's Selection Data Set.

The **ABACKUP EXECUTE** command gives you dynamic real-time analysis at the time of backup. When the **ABACKUP** is executed, the SDSL code is analyzed and the data sets are written to the Selection Data Set for immediate ABARS processing. The **CONVSDSL** command builds the Selection Data Set in real time and can be used in the following ways:

- Use it to examine the resulting Selection Data Set to ensure you achieve the results you want.
- Use it as immediate input to ABARS Aggregate LoadBalancer to split up the Selection Data Set. The ABARS Aggregate LoadBalancer job can submit the **ABACKUP** job if desired.

Note: SDSL does not accept Selection Data Sets as a means of input. Selection Data Sets built by Automated Critical Data Identification cannot become input to SDSL.

How does SDSL processing work?

You might find it helpful to understand the sequence of events involved in SDSL processing.

1. You use ISMF to define an aggregate and activate it into SMS.
2. You add the aggregate to the list of aggregates managed by ABARS Manager.
3. You code SDSL into the aggregate's Selection Data Set. This Selection Data Set can contain a filter name defined by using Advanced Backup and Recovery for z/OS Group Filter Management feature. The filter name is associated with a list of data sets to filter.
4. When you save the Selection Data Set containing the SDSL code, you are prompted to set a switch that signifies that SDSL is active for this aggregate.
5. When you submit an **ABACKUP EXECUTE** command for this aggregate, the following events occur:
   a. ABARS Manager stores a copy of the original SDSL-coded Selection Data Set in the IDS.
   b. ABARS Manager reads the SDSL coded in the aggregate's Selection Data Set, filters the entries, and dynamically rewrites the aggregate's Selection Data Set with the list of data sets requested by the SDSL code.

   Note: If you want to use aggregate load balancing to split up the SDSL aggregate, you can use the **CONVSDSL** command instead of **ABACKUP EXECUTE**. **CONVSDSL** provides the same functionality (except for the **ABACKUP**) and builds the Selection Data Set in real-time for immediate input to ABARS Aggregate LoadBalancer.

   a. The Aggregate LoadBalancer splits up the Selection Data Set into the specified number of Selection Data Sets, or aggregates.
   b. After the splitting is complete, the Selection Data Sets produced can be used by one or more **ABACKUP** jobs.

6. ABARS reads the updated Selection Data Set.
7. When **ABACKUP** completes successfully, the Selection Data Set is restored to the original SDSL Selection Data Set you coded.

SDSL keywords

SDSL processing involves two main keywords: **CANDIDATES** and **FILTERS**. To use SDSL successfully, you must understand how these two keywords work.

Use the **CANDIDATES** keyword to specify selection criteria that builds a list of data sets to back up, and use the **FILTERS** keyword to eliminate from the **CANDIDATES** list those data sets you do not want included in the backup.

The following rules apply to the use of these keywords:

- **FILTERS** always overrides **CANDIDATES**. ABARS Manager examines the **CANDIDATES** list and then refines it according to the **FILTERS**.
- The first hit ends filter processing for the data set.

Some examples might help you understand how the **CANDIDATES** list works.

Example 1

This example establishes a candidate list consisting of all cataloged data sets matching the mask XYMI.IVP.**. The ABARS backup category will be INCLUDE unless changed by filtering.

```
CANDIDATES(
           INCLUDE_DSN(XYMI.IVP.** )
           )
```

Example 2

This example establishes a candidate list consisting of all cataloged data sets matching the mask XYMI.IVP.**. The ABARS backup category will be INCLUDE unless changed by filtering. The **FILTERS** statements moves all data sets beginning with XYMI.IVP.FB.** to the allocate list.

```
CANDIDATES(
           INCLUDE_DSN(XYMI.IVP.**)
           )
FILTERS(
        ALLOCATE(DSN EQ XYMI.IVP.FB.**)
        )
```

Example 3

This example illustrates how the FILTERS statement works. Consider a situation in which you want to establish a list of data sets that are migrated to ML1, and match mask X23.**, you can use the syntax shown below.

- The **CANDIDATES** statement establishes a candidate list of all cataloged data sets matching X23.**.
- The **FILTERS** statement moves all ML1 data sets to the INCLUDE list.
- There might be additional data sets matching the mask that are not migrated to ML1. To remove these, use the EXCLUDE statement

Note: Because the **CANDIDATES** list collects data from the catalog based on masks, the **CANDIDATES** list includes GDGBASE and PATH information. If GDGBASE and PATH information is found, you need to filter these out as well.

```
CANDIDATES(
           INCLUDE_DSN(X23.**)
           )
FILTERS(
        INCLUDE(ML1 EQ YES)
```

```
        EXCLUDE(DSN EQ X23.**)
        )
```

Constructing SDSL instructions

SDSL instructions are entered as keywords and operands. ABARS Manager uses the ISPF Editor for entry and modification of the instructions.

When you save your entries and exit the ISPF Edit panel, the SDSL instructions are saved to the Advanced Backup and Recovery for z/OS Inventory Data Set (IDS). You can use the standard ISPF Edit commands **COPY**, **CREATE**, and **REPLACE** to save the instructions elsewhere, or to insert instructions copied from another source.

Constructing a list of data sets is a 2-step process:

1. Establish a list of data set names called a *candidate list* using the **CANDIDATES** keyword.
2. Tailor the list using the **FILTERS** keyword, which allows you to remove or change the data in the candidate list.

Rules for coding CANDIDATES and FILTERS instructions

Your SDSL **CANDIDATE** and **FILTERS** instructions must be specified as follows:

- Instructions must be specified using columns 1-72.
- Continuation statements are unnecessary.
- Use /* to start a comment and */ to terminate a comment. You can nest comments within other comments.
- Case is ignored when the instructions are parsed.

Example

```
CANDIDATES(
           INCLUDE_DSN(PAYROLL.**)
           )
FILTERS(
        ACCOMPANY(UNIT NE DASD)
        ALLOCATE(DSN EQ **.WORK.**)
        EXCLUDE((RECFM EQ FBA OR FBM) AND (LRECL EQ 121 OR 133))
        )
```

This example establishes a candidate list consisting of all cataloged data sets matching the mask PAYROLL.**. The ABARS backup category will be INCLUDE unless it is changed by filtering. In the example, filters are being used.

- The first filter causes all data sets with a unit name not equal (NE) to DASD, found on a list other than ACCOMPANY, to be removed from such list and placed on the ACCOMPANY list.
- The second filter causes any data set not matching the first filter, with a data set name that matches (EQ) the mask **.WORK.**, found on a list other than ALLOCATE, to be removed from such list and added to the ALLOCATE list.
- The third filter excludes any data sets not matching the first or second filter, with a RECFM equal (EQ) to either FBA or FBM, and whose LRECL is equal (EQ) to either 121 or 133 (data sets that appear to be reports). Data sets matching the mask are excluded from all lists they might have been on.

Filters are processed for each data set in the candidate list. Because a filter match causes the filter action to be taken without examining subsequent filters for the data set being processed, data set PAYROLL.ABC.WORK.DATA with RECFM FBA and LRECL 133 will be ALLOCATED, rather than EXCLUDED, because it matches the ALLOCATE filter first. The desired logic can be achieved by manipulating the filter sequence and/or by using Boolean expressions.

Processing SDSL instructions

A general understanding of the logic used, data set attributes tested, and Boolean operator evaluation in filters is important if you want to take full advantage of the SDSL feature.

Filtering logic

Each candidate data set from each candidate list (INCLUDE, ALLOCATE, ACCOMPANY) is compared to the filter list. Filters are compared in the order they are entered in the SDSL instructions. Filters from a group filter definition are processed in entry sequence, with the group of filters processed when the GROUP keyword is encountered relative to local filters. If a filter test is true, the action requested by the filter keyword applies, and subsequent filters are not examined. If the filter test is not true, the next filter is examined; hence, the process is referred to as 'first match, you're out'. For a given data set, if no filter test is true, the initially chosen ABARS category from the candidate list specification remains in effect.

Consider the following example:

```
FILTERS(
          EXCLUDE(DSN EQ SYS1.**)
          EXCLUDE(LRECL EQ 133)
        )
```

Because any data set that matches the filter SYS1.** is excluded from further consideration by the 'first match, you're out' logic, the test for LRECL is never applied to SYS1.** data sets.

Which data set attributes are tested by filters?

Because not all attributes used in filters are applicable to all data set types, when ABARS Manager encounters a situation such as an attempt to filter the data set list using CISIZE for a non-VSAM data set, it issues a descriptive message and sets the return code to 4.

If the data in the attribute is invalid (due to a broken catalog or some other such error) ABARS Manager issues a message indicating that the data is in error and sets the return code to 4.

In either of these cases, any remaining filters are applied as though the offending filter did not exist.

To resolve situations such as these, you can use the DSNTYPE parameter in conjunction with attributes that might not be applicable to all data set types in the list.

Instead of specifying the following EXCLUDE in the filter:

```
EXCLUDE(CISIZE GT 4096)
```

you can use the DSNTYPE parameter in the EXCLUDE:

```
EXCLUDE(DSNTYPE EQ VSAM AND CISIZE GT 4096)
```

SDSL supports the following values for the DSNTYPE parameter:

AIX®	GDGACT	LDSC	RRDSD
AIXC	GDGBASE	LDSD	USERCAT
AIXD	KSDS	NONVSAM	VRRDS
AIXI	KSDSC	PATH	VRRDSC
ESDS	KSDSD	POSIX	VRRDSD
ESDSC	KSDSI	RRDS	
ESDSD	LDS	RRDSC	

Using Boolean operators in filters

You can achieve many filtering objectives by using the 'first match you're out' logic on its own. Filters can be viewed as a nested IF-THEN/ELSE structure, meaning that subsequent filters are ignored once a filter tests true. If you want to use complex Boolean expressions in filters, you need to be familiar with Boolean logic, especially the XOR and NOT keywords.

The following rules apply to the use of Boolean operators in filters:

- Tests can be preceded by the NOT operator. If an expression is preceded by the keyword NOT, that expression's true/false evaluation is reversed. Note that you can accomplish tests for NOT conditions by using the EQ and NE comparators. Tests that use only the EQ comparator can evaluate as YES or NO. Hence, it is advisable to limit the use of the NOT keyword by using a native test for a NOT condition. For example, you could specify RECFM NE FB instead of NOT(RECFM EQ FB). Avoid double negative logic whenever possible.
- Use parentheses to control the order in which the operators are evaluated.
- You can use OR, AND, and XOR (exclusive OR) to join tests.

You can use Boolean operators to test multiple values without having to repeat the subject (DSN or attribute) of the test, or the comparator, or both.

For example:

```
EXCLUDE(LRECL EQ 121 OR 133)
```

The keyword LRECL and the comparator EQ are specified once only. In the following example, note that the test subject (CREDT) is specified once, but the comparators are specified for both tests:

```
EXCLUDE(CREDT LT *-30 or GT*-2)
```

ABARS Manager evaluates tests delimited by parentheses before evaluating tests outside parentheses. When parenthetical expressions are nested, ABARS Manager evaluates the expressions from the innermost to the outermost. Consider the following example:

```
EXCLUDE((RECFM EQ FBA OR FBM OR VBA OR VBM) +
           AND (LRECL EQ 121 OR 133))
```

ABARS Manager evaluates the RECFM and LRECL tests before the AND condition, using the results of the RECFM and LRECL tests. This test might be used to EXCLUDE data sets that appear to be reports recorded on DASD. Omitting the parentheses in this example would result in a different outcome than including them does.

About the SDSL CANDIDATES instruction

Candidate specifications are used to create three lists of data sets: INCLUDE, ALLOCATE, and ACCOMPANY. Initially, a data set can be on more than one of these lists. However, after filtering, any given data set can be on only one list. If a data set appears on more than one list after filtering, an error occurs and the ABARS backup is not launched. Data set names can be obtained from the catalog or from a volume VTOC (or from both sources).

SDSL CANDIDATES instruction syntax

This topic shows the **CANDIDATES** instruction syntax, describes its keywords, and provides notes on its processing.

```
CANDIDATES( { [ ACCOMPANY_DSN( dsn1 |  dsnsmask1 ...dsnn | dsnmaskn ) ALLOCATE_DSN( dsn1
| dsnsmask1 ...dsnn | dsnmaskn ) INCLUDE_DSN( dsn1 | dsnsmask1 ...dsnn | dsnmaskn ) ] }
DCOLLECT( { [ EXCLUDEVOLUMES ( volser1 ...volsern ) ] [ STORAGEGROUP( stgrp1 ...stgrpn ) ]
[ VOLUMES( volser1 ...volsern ) ] } )
```

Keywords

CANDIDATES(...)
This high-level keyword distinguishes candidates from filters. All candidates are specified in parentheses under the high-level keyword **CANDIDATES**.

ACCOMPANY_DSN(...)
Use this keyword to select data set names from the catalog and place them on the ACCOMPANY list. You can specify discrete data set names or masks to capture cataloged data sets that match the mask. Masking uses Enhanced ACS masking.

Your **CANDIDATES** instruction specification must include at least one of the **ACCOMPANY_DSN**, **ALLOCATE_DSN**, or **INCLUDE_DSN** keywords.

Note: Specifying a DASD data set for the ACCOMPANY_DSN keyword causes an ABARS failure. To avoid this error, set the aggregate set-up option **Change ACCOMPANY of DASD data set to INCLUDE** to Y (yes).

ALLOCATE_DSN(...)
Use this keyword to select data set names from the catalog and place them on the ALLOCATE list. You can specify discrete data set names or masks to capture cataloged data sets that match the mask. Masking uses Enhanced ACS masking.

Your **CANDIDATES** instruction specification must include at least one of the **ACCOMPANY_DSN**, **ALLOCATE_DSN**, or **INCLUDE_DSN** keywords.

Specifying a tape data set for the **ALLOCATE_DSN** keyword causes an ABARS failure. To avoid this error, set the aggregate set-up option **Change ALLOCATE of Tape data set to INCLUDE** is set to Y (yes).

INCLUDE_DSN(...)
Use this keyword to select data set names from the catalog and place them on the INCLUDE list. You can specify discrete data set names or masks to capture cataloged data sets that match the mask. Masking uses Enhanced ACS masking.

Your **CANDIDATES** instruction specification must include at least one of the **ACCOMPANY_DSN**, **ALLOCATE_DSN**, or **INCLUDE_DSN** keywords.

DCOLLECT({ [EXCLUDEVOLUMES (*volser1 ...volsern*)] [STORAGEGROUP(*stgrp1 ...stgrpn*)] [VOLUMES(*volser1 ...volsern*)] })
Use this keyword to specify the IDCAMS DCOLLECT parameters that select volumes from which data set names are obtained from the VTOCs. All data set names obtained using DCOLLECT are added to the INCLUDE list. If you specify the **DCOLLECT** keyword, you must also specify for it at least one **EXCLUDEVOLUMES**, **STORAGEGROUPS**, or **VOLUMES** keyword.

- **VOLUMES** specifies one or more volumes from which data set names are to be obtained.
- **EXCLUDEVOLUMES** specifies one or more volumes to exclude.
- **STORAGEGROUPS** specifies one or more storage groups from which data set names are to be obtained.

Note: These parameters are passed to IDCAMS without modification. You can specify DCOLLECT only once within a **CANDIDATES** instruction. For more information about **DCOLLECT** consult IBM documentation.

Processing notes

For each data set collected through **DCOLLECT**, a catalog lookup is performed. If a data set found on a volume is not cataloged, its removal from the list depends on how the aggregate setup option **Remove uncataloged data sets** is set.

If a catalog entry for a data set name obtained using **DCOLLECT** exists, but the **DCOLLECT** volume is not one of the volumes for the cataloged data set, SDSL generates a return code 8 and the ABARS backup is not launched. This situation might occur when an uncataloged data set that is not managed by SMS

is present on the volume chosen for **DCOLLECT** and a catalog entry for the data set name points to a different volume.

Candidates selected by their fully qualified data set name or a data set name mask are obtained from the catalogs. Candidates selected from VTOCs are obtained using IDCAMS **DCOLLECT**.

Similar to ABARS, SDSL processing initially allows a data set to exist on more than one of the three lists: INCLUDE, ALLOCATE, and ACCOMPANY. Native ABARS processing requires that after filtering, a given data set can exist on one *and only one* of these lists. This cannot be a requirement with SDSL generated Selection Data Sets. ABARS Manager processing of Selection Data Sets, whether native or constructed by SDSL, includes the option to resolve multiple category conflicts. Therefore, if a data set is included on more than one list after filtering, this might not cause an ABARS error if the ABARS Manager backup option to resolve multiple category conflicts is selected.

Because SDSL filtering allows an initially-chosen data set backup category to be changed, adding a data set to the candidate list in more than one category is not necessary, and might cause more confusion.

The primary reason SDSL allows a data set to be on more than one backup category list is to support familiar functionality for native ABARS users. Note, however, that the mechanism to remove a data set name from one list is slightly different in SDSL than it is in native ABARS.

For example, in native ABARS, `ALLOCATEEXCLUDE(APPLE.BAKER.CHARLIE)` means 'exclude the data set `APPLE.BAKER.CHARLIE` from the list of ALLOCATE data sets.' To accomplish the same thing in SDSL, you could set up this filter: `EXCLUDE(DSN EQ APPLE.BAKER.CHARLIE AND CANDCAT EQ ALLOCATE)`

Note: Candidates can only be selected based on cataloged data set names or data set names from DASD volumes. You cannot select candidates by both name criteria and attribute criteria. If list criteria must use data set attributes, build a candidate list of all potential data sets, and subsequently filter unwanted data sets based on attributes.

About the SDSL FILTERS instruction

You can use filters to exclude names from the candidates list and to change the ABARS backup category initially established for a data set (or both). The FILTERS specification always overrides the CANDIDATES specification; processing examines the **CANDIDATES** list, then modifies the list according to the **FILTERS** instruction.

Filters that are commonly used in multiple aggregates can be established as a filter group and given a group name. An aggregate's SDSL instructions can include one or more filter groups. Physical placement of the request to invoke a group of filters relative to individual local filters allows the choice of whether group filters are to take precedence over local filters or the other way around.

SDSL FILTERS instruction syntax

The **FILTERS** instruction includes action keywords, filter keywords, and filter criteria. This topic describes both types of keywords and the filter criteria.

The **FILTERS** instruction takes an action keyword (**EXCLUDE**, **INCLUDE**, **ALLOCATE**, **ACCOMPANY**), followed by one or both of the following:

- A **GROUP** keyword and its associated filter group name.
- A **FILTER** keyword and its associated filter criteria.

```
FILTERS(...)
```

Keywords

ACCOMPANY|ALLOCATE|EXCLUDE|INCLUDE
Use the action keywords to specify whether a data set matching the filter criteria should be included, excluded, or have its ABARS category (INCLUDE, ALLOCATE, ACCOMPANY) changed. Changing a category means the data set is placed on the list for the specified category list and removed from any other lists it may might have been on when the **CANDIDATES** were processed.

You must specify at least one action keyword in the **FILTERS** instruction.

FILTERS action keywords are as follows:

- **INCLUDE** – If the criteria are true, the ABARS category of the data set being tested is changed to INCLUDE.
- **EXCLUDE** – If the criteria are true, the data set being tested is excluded from the Selection Data Set.
- **ACCOMPANY** – If the criteria are true, the ABARS category of the data set being tested is changed to ACCOMPANY.
- **ALLOCATE** – If the criteria are true, the ABARS category of the data set being tested is changed to ALLOCATE.

filter keyword(*filter criteria*)
Use a filter keyword to specify an individual (local) filter keyword and filter criteria. If the filter evaluates to TRUE, the implied action of the specified action keyword (**ACCOMPANY**, **ALLOCATE**, **EXCLUDE**, or **INCLUDE**) is taken.

The filter keyword (filter criteria) specification defines the tests to be made against a data set. Filter criteria can be a single comparison or a complex Boolean expression. The entire expression must be true for the action implied by the action keyword to be applied to the data set being tested. Detailed descriptions of the available filter keywords and filter criteria are provided in the next section of this topic.

Your **FILTER** instruction must include at least one action keyword plus at least one filter keyword and its filter criteria or one **GROUP** keyword and its associated group name.

GROUP(*groupname*)
Specify the GROUP keyword and the group name of a filter group to be processed. If the filter evaluates to TRUE, the implied action of the specified action keyword (**ACCOMPANY**, **ALLOCATE**, **EXCLUDE**, or **INCLUDE**) is taken. The GROUP(*group name*) specification identifies a group of filters to be processed.

Your **FILTER** instruction must include at least one action keyword plus at least one filter keyword and its filter criteria or one **GROUP** keyword and its associated group name.

Note: Physical placement of the **GROUP** keyword specifies where the GROUP filters are to be processed relative to other filters in the list.

Filter keywords and criteria

Filter criteria comprise one or more conditional expressions that can be joined with Boolean operators. In general, single conditions use the following syntax: `keyword operator value`.

Table 5. Filter keywords and criteria

Keyword	Valid Operators	Valid Values	Description
ALLOCTYPE	EQ | NE	CYL | TRK | BLK	Compares the data set allocation units.
BLKSZ	EQ | NE | GE | GT | LE | LT	*nnnnn*	Compares the data set block size.
BYTEA	EQ | NE | GE | GT | LE | LT	*bytes allocated*	Compare the data set bytes allocated. Use any of the following formats to specify *bytes allocated*: nnnnnnnnnnnnnn (bytes) nnnnnnnnnnnnnnB nnnnnnnnnnK nnnnnnnnM nnnnnG

Table 5. Filter keywords and criteria (continued)

Keyword	Valid Operators	Valid Values	Description
BYTEU	EQ | NE | GE | GT | LE | LT	*bytes used*	Compare the data set bytes used. Use any of the following formats to specify *bytes used*: nnnnnnnnnnnnnn (bytes) nnnnnnnnnnnnnnB nnnnnnnnnnK nnnnnnnnM nnnnnG
CANDCAT	EQ	INCLUDE | ALLOCATE | ACCOMPANY	Compare the data set's current ABARS backup category. Ordinarily, this test is used in conjunction with at least one other test. A filter that has no CANDCAT test applies to all lists. Use CANDCAT if a test is to be applied to one list only. This keyword is used primarily when the native ABARS functionality is employed where a data set is on more than one list and must be subsequently removed from all but one list. **Note:** Because SDSL filters can be used to change an ABARS backup category, you can use this technique instead of placing a data set on multiple lists and then excluding the data set from appropriate lists. For example, `EXCLUE(DSN EQ APPLE.BAKER.CHARLIE AND CANDCAT EQ ALLOCATE)` tests TRUE for data set APPLE.BAKER.CHARLIE if, and only if, the data set is on the ALLOCATE list.
CREDT	EQ | NE | GE | GT | LE | LT	*mmddyy* | * | *-*nnn*	Compare the data set's creation date. Specify an absolute date in the format mmddyy, * to represent the current system date, or *-nnn to represent a date relative to the current system date. For example, `CREDT LT *-30` tests for a data set created more than 30 days before the present.
DATACLAS	EQ | NE	*dataclas* | *mask* | ''	Compare the data set SMS data class. Specify a data class, a data class mask, or a blank space (which represents 'none'). Extended ACS filtering pattern masking applies.
DEVN	EQ | NE	*devicename*	Compare the data set device name. For example, if you specify DEVN EQ 3380, any 3400-nn device is considered a match.
DSN	EQ | NE	*data set name* | *mask*	Compare the data set name. The comparison uses enhanced ACS masking.

Table 5. Filter keywords and criteria (continued)

Keyword	Valid Operators	Valid Values	Description
DSNTYPE	EQ | NE	AIX AIXC AIXD AIXI ESDS ESDSC GDGACT GDGBASE KSDS KSDSC KSDSD KSDSI LDS LDSC LDSD NONVSAM PATH POSIX RRDS RRDSC RRDSD USERCAT VRRDS VRRDSC VRRDSD VRRDSI VSAM	Specify the type of data set that must be processed to perform the remainder of the test within the filter. AIX–Any and all components AIXC–Cluster only AIXD–Data component only AIXI–Index component only ESDS–any and all components ESDSC–Cluster only GDGACT–Active GDG entries, but not GDG Base entries GDGBASE–GDG Base entry KSDS–Any and all components KSDSC–Cluster only KSDSD–Data component only KSDSI–Index component only LDS–Any and all components LDSC–Cluster only LDSD–Data component only NONVSAM–Any non-VSAM data set PATH–Path only POSIX–Open Edition (Linux®) files RRDS–Any and all components RRDSC–Cluster only RRDSD–Data component only USERCAT–A user catalog VRRDS–Any and all components VRRDSC–Cluster only VRRDSD–Data component only VRRDSI–Index component only VSAM–KSDS, ESDS, RRDS, or anything else that is non-VSAM
DSORG	EQ | NE	*dsorg*	Perform a 2-character data set organization comparison. **Note:** A VS data set organization is considered a match with any VSAM data set organization value that can be returned from the catalog (VE, VI, and so forth).
EXPDT	EQ | NE | GE | GT | LE | LT	*mmddyy* | * | *+*nnn* | NEVER	Compare the data set expiration date. You can specify any of the following: • An absolute date, using the mmddyy format • The current system date, by specifying a single asterisk • A date relative to the current system date, by specifying a single asterisk, followed by a plus symbol, then a number of days. For example, specify `EXPDT GT *+30` to mean 'any data set with an expiration date beyond 30 days from the present.' • NEVER, indicating the data set does not have an expiration date.
GDS	EQ	YES | NO	Test whether the data set is a generation data set (absolute (G0000V00) or relative).
GROUP (*groupname*)			GROUP specifies a group of filters to be processed; *groupname* is a previously defined group of filters. Group filters are processed where the GROUP keyword is physically placed relative to other filters. groupname can be a maximum of 30 characters. Define group filters using the Advanced Backup and Recovery for z/OS **Setup and Configuration** panel. Filters can be nested, which allows you to define sets and subsets of filters. With careful consideration, you can avoid redundant typing and maintenance for common filters and minimize cross-aggregate overlaps.
LRECL	EQ | NE | GE | GT | LE | LT	*nnnnn*	Compare the data set logical record length.

Table 5. Filter keywords and criteria (continued)

Keyword	Valid Operators	Valid Values	Description
MIGRATE	EQ	YES | NO	Test whether a data set is migrated to either migration level 1 or migration level 2..
ML0	EQ	YES | NO	Test whether a data set resides on primary DASD (has not been migrated).
ML1	EQ	YES | NO	Test whether a data set has been migrated to migration level 1.
ML2	EQ	YES | NO	Test whether a data set has been migrated to migration level 2.
MGMTCLAS	EQ | NE	*mgmtclas* | *mask* | ' '	Compare the data set's SMS management class. You can specify a management class, a management class mask, or a blank space (which represents 'none'). The comparison uses enhanced ACS masking.
RECFM	EQ | NE	*recfm*	Compare the data set's record format. You can specify any of these filter values: • U • FB • FS • FA • FM • VB • VS • VA • VM • FBA • FBM • FBS • FSA • FSM • VBA • VBM • VBS • VSA • VSM • FBSA • FBSM • VBSA • VBSM
REFDT	EQ | NE | GE | GT | LE | LT	*mmddyy* | * | **-nnn*	Compare the data set's last reference date.
SMS	EQ	YES | NO	Test whether the data set is managed by SMS.
STORCLAS	EQ | NE	*storclas* | *mask* | ' '	Compare the data set's SMS storage class. You can specify a storage class, a storage class mask, or a blank space (which represents 'none'). The comparison uses enhanced ACS masking.
STOGRP	EQ | NE	*storgrp* | *mask* | ' '	Compare the data set's SMS storage group. You can specify a storage group, a storage group mask, or a blank space (which represents 'none'). The comparison uses enhanced ACS masking.
UNIT	EQ | NE	CART | DASD | TAPE	Compare the data set's unit. Your specification can be any of the following: • CART–Any non-3400 device • TAPE–Any 3400 device. Because the CANDIDATES list uses masks to collect data from the catalog, the list includes GDGBASE and PATH information. GDGBASE and PATH entries do not have VOL or UNIT information and cause a failure if you use the UNIT or VOL filter keyword and do not also explicitly exclude GDGBASE and PATH information. For example, `EXCLUDE(DSNTYPE NE GDGBASE OR DSNTYPE NE PATH AND (VOL EQ ARCH2))`.

Table 5. Filter keywords and criteria (continued)			
Keyword	**Valid Operators**	**Valid Values**	**Description**
VOL	EQ | NE	*volser* | *mask*	Compare the volumes on which the data set is allocated. For multi-volume data sets: • An EQ condition is TRUE whenever at least one volume matches your *volser* or *mask* specification. • A NE condition is TRUE whenever all volumes match your *volser* or *mask* specification. Consider an example in which data set APPLE.BAKER.CHARLIE resides on volumes WRK001, TSO002, and WRK003. • If you specify a filter of `INCLUDE(VOL EQ TSO*)`, the test is TRUE and APPLE.BAKER.CHARLIE is included. • If you specify a filter of `EXCLUDE(VOL NE TSO*)`, the test is FALSE and APPLE.BAKER.CHARLIE is *not* excluded by this filter. Because the CANDIDATES list uses masks to collect data from the catalog, the list includes GDGBASE and PATH information. GDGBASE and PATH entries do not have VOL or UNIT information and cause a failure if you use the UNIT or VOL filter keyword and do not also explicitly exclude GDGBASE and PATH information. For example, `EXCLUDE(DSNTYPE NE GDGBASE OR DSNTYPE NE PATH AND (VOL EQ ARCH2))`

CANDIDATES and FILTERS usage examples

SDSL's **CANDIDATES** and **FILTERS** instructions significantly enhance the native ABARS selection criteria. Use the examples in this topic to guide you as you build your own SDSL selection criteria.

The selection criteria shown in this first example establish a candidates list that includes all data sets that match mask BKMI.IVP.**, but excludes those that were last referenced before March 1, 2013.

```
CANDIDATES(
           INCLUDE_DSN(BKMI.IVP.**)
           )
FILTERS(
        EXCLUDE(REFDT LT 030113)
       )
```

Now consider an example in which the selection criteria establish a candidates list that includes BKMI.IVP.CNTL and allocates all USERCAT.** data sets. The **FILTERS** statement is used to exclude USERCAT.EDP.** data sets from the USERCAT.** data sets put in the ALLOCATE list. As a result, BKMI.IVP.CNTL is put in the INCLUDE list, and all user catalogs matching USERCAT.** are put in the ALLOCATE list *except* for those matching mask USERCAT.EDP.**.

```
CANDIDATES(
           INCLUDE_DSN(BKMI.IVP.CNTL)
           ALLOCATE_DSN(USERCAT.**)
          )
 FILTERS(
         EXCLUDE((DSN EQ USERCAT.EDP.**))
        )
```

In the following example, the CANDIDATES list's DCOLLECT option is used to include all data sets on volume TSO001. The **FILTERS** statement moves any data sets matching mask DA1.** to the ALLOCATE list and excludes all data sets matching DCD.**.

```
CANDIDATES(
           DCOLLECT(VOL(TSO001))
           )
FILTERS(
        ALLOCATE(DSN EQ DA1.**)
        EXCLUDE(DSN EQ DCD.**)
       )
```

Now consider an example in which the **DCOLLECT** option of the CANDIDATES list is used to include all data sets on volumes SMS001 and SMS002. The **FILTERS** statement is used to exclude any data sets that match the masks shown in the EXCLUDE list.

```
CANDIDATES(DCOLLECT(VOLUMES(SMS001 SMS002)))
FILTERS(
        EXCLUDE(DSN EQ AMP.TEST.**)
        EXCLUDE(DSN EQ BKMI.**)
        EXCLUDE(DSN EQ SIS.**)
        EXCLUDE(DSN EQ SYS3.**)
        EXCLUDE(GDS EQ YES)
       )
```

In this example, the **DCOLLECT** option of the **CANDIDATES** list is used to include all data sets on volumes SMS001 and SMS002. The FILTERS statement with the **GROUP** option is used to exclude data sets using filters that were previously defined under the name of PROD. These group filters can be defined through the **Group Filter Management** option on the **Setup and Configuration** panel.

```
CANDIDATES(DCOLLECT(VOLUMES(SMS001 SMS002)))
FILTERS(GROUP(PROD))
```

The instructions in following example back up some of the files necessary to re-allocate the Db2 system databases (as opposed to application databases) and restore the current system image copies and recent archive logs, along with some related files. This environment has three sharing Db2 systems, which is why three sets of image copies and four sets of archive logs are included.

The **FILTERS** section uses OR statements. Use AND or OR statements carefully. Unique **EXCLUDE** filters help identify what is excluded far more easily than using AND or OR statements.

Note: When referencing relative generations within a mask, you cannot follow a double asterisk directly with a relative value. For example, A.B.**(0) is not a valid mask specification; however, you could specify A.B.**.*(0) or A.B.**.LAST(0).

```
CANDIDATES(
           ALLOCATE_DSN(HDB2V.DSNDBC.DSNDB*.**)
/* Allocate the "system" databases      */
           ALLOCATE_DSN(CDB2V.DSNDBC.DSNDB*.**)
           ALLOCATE_DSN(CDB2V.DSNDBC.CDB2WORK.**)
           ALLOCATE_DSN(CDB2V.DSNDBC.CDB3WORK.**)
           INCLUDE_DSN(CDB2.IMAGCOPY.DSNDB*.**.*(0))
/* Include latest "system" Image Copies  */
           INCLUDE_DSN(DB2.IMAGCOPY.DSNDB*.**.*(0))
           INCLUDE_DSN(HDB2.IMAGCOPY.DSNDB*.**.*(0))
           INCLUDE_DSN(CDB2ARCH.ARCHLOG1.D*.**)
/* Include the "system" Archive Logs     */
           INCLUDE_DSN(CDB3ARCH.ARCHLOG1.D*.**)
           INCLUDE_DSN(DB2ARCH.ARCHLOG1.D*.**)
           INCLUDE_DSN(HDB2ARCH.ARCHLOG1.D*.**)
           INCLUDE_DSN(DB2.PROD.CONTROL)
/* Include several DB2 related libraries */
           INCLUDE_DSN(DB2.PROD.LINKLIB)
           INCLUDE_DSN(DB2.PRODUCTS.CLIST)
           INCLUDE_DSN(DB2.RUNLIB.LOAD)
          )
FILTERS(
/*  Filter the Archive Logs  */
        EXCLUDE((CREDT LT *-0) AND +
/*  Exclude any Archive Logs */
/*  not created today        */
         (DSN EQ CDB2ARCH.ARCHLOG1.D*.** +
          OR CDB3ARCH.ARCHLOG1.D*.** +
          OR DB2ARCH.ARCHLOG1.D*.** +
          OR HDB2ARCH.ARCHLOG1.D*.**))
       )
```

The following example illustrates the backing up of tape files without including generation data sets. The **CANDIDATES** statement establishes a candidates list that includes all data sets that match mask DCCICSPN.** and DCCICSPV.**. The **FILTERS** statement moves tape data sets to the include list except

for generation data sets and data that is migrated. In addition, the **FILTERS** statement excludes all data from the list that matches the data set masks found in the exclude list.

```
CANDIDATES(
           INCLUDE_DSN(DCCICSPN.** DCCICSPV.**)
          )
FILTERS(
        INCLUDE(UNIT EQ TAPE AND GDS EQ NO AND MIGRATE EQ NO)
        EXCLUDE(DSN EQ DCCICSPN.**)
        EXCLUDE(DSN EQ DCCICSPV.** )
        EXCLUDE(DSN EQ DCCICSPV.*.DFHLOG.**)
        EXCLUDE)DSN EQ DCCICSPV.*.DFHSHUNT.**)
        EXCLUDE(DSN EQ DCCICSPV.*.*.JOURNAL.**)
        EXCLUDE(DSN EQ DCCICSTV.*.DFHLOG.**)
        EXCLUDE(DSN EQ DCCICSTV.*.DFHSHUNT.**)
        EXCLUDE(DSN EQ DCCICSTV.*.*.JOURNAL.**)
       )
```

The following example illustrates the backing up of only non-tape generation data sets.

```
CANDIDATES(
           INCLUDE_DSN(DCCICSPN.** DCCICSPV.**)
          )
FILTERS(
        EXCLUDE(GDS EQ NO)
        EXCLUDE(UNIT EQ TAPE OR CART)
        EXCLUDE(DSN EQ DCCICSPV.*.DFHLOG.**)
        EXCLUDE(DSN EQ DCCICSPV.*.DFHSHUNT.**)
        EXCLUDE(DSN EQ DCCICSPV.*.*.JOURNAL.**)
        EXCLUDE(DSN EQ DCCICSTV.*.DFHLOG.**)
        EXCLUDE(DSN EQ DCCICSTV.*.DFHSHUNT.**)
        EXCLUDE(DSN EQ DCCICSTV.*.*.JOURNAL.** )
       )
```

This example shows the backing up of tape data sets only.

```
CANDIDATES(
           INCLUDE_DSN(DC.DMGT.ABARS.PROD.ORCA0*.**)
          )
FILTERS(
        EXCLUDE(UNIT NE CART)     /* INCLUDE( UNIT EQ TAPE ) */
       )
```

The instructions in the following example establish a candidates list that includes all data sets matching the INCLUDE_DSN list and accompanies all data sets matching BACKUP.**.

The first **FILTERS** statement, (`INCLUDE(DSN EQ DS01.DUASMI.**`) matches a mask in the **CANDIDATES** list. Because filter processing works on a ‘first hit and you are out’ basis, we want to make sure that data sets that match `DS01.DUASMI.**` are included and not excluded because of a following **EXCLUDE** statement. In essence, this is interpreted as 'take the **FILTER** list and exclude everything that matches an EXCLUDE statement except these.'

The **CANDIDATES** list collects data from the catalog based on masks; therefore, the **CANDIDATES** list includes GDGBASE and PATH information. GDGBASE and PATH entries do not have VOL or UNIT information and will result in a failure if VOL or UNIT keywords are used, unless GDGBASE and PATH entries are excluded. Therefore, `EXCLUDE(DSNTYPE NE GDGBASE OR DSNTYPE NE PATH AND (VOL EQ ARCHV2))` is coded to ignore GDGBASE and PATH information so that the correct data sets are found.

```
CANDIDATES(
            INCLUDE_DSN(DEMOAGG.**)
            INCLUDE_DSN(RLS1.**)
            INCLUDE_DSN(SYS3.**)
            INCLUDE_DSN(DS03.**)
            INCLUDE_DSN(DS01.DUASMI.**)
            INCLUDE_DSN(DS01.ASAP.GDG(-2))
            ACCOMPANY_DSN(BACKUP.**)
           )
FILTERS(
        INCLUDE(DSN EQ DS01.DUASMI.**)
        EXCLUDE(CREDT LT *-80)
        EXCLUDE(DSN EQ DEMOAGG1.**.SDS)
        EXCLUDE(DSN EQ DEMOAGG.**.C.**)
        EXCLUDE(DSN EQ DEMOAGG.**.D.**)
        EXCLUDE(DSN EQ DEMOAGG.**.I.**)
```

```
        EXCLUDE(DSN EQ DEMOAGG.**.O.**)
        EXCLUDE(DSNTYPE NE GDGBASE OR DSNTYPE NE PATH AND (VOL EQ
                ARCHV2)
        )
```

The following example illustrates a mask that identifies the selected relative gens of GDSs. Note that the relative value cannot directly follow a double asterisk. Therefore, A.B.**(0) is invalid, but A.B.**.*(0) is valid.

```
CANDIDATES(
            INCLUDE(A.B.**)) /* THE BROAD MASK YOU WANT TO USE */

FILTERS(
        INCLUDE(DSN EQ A.B.**.*(0))     /* INCLUDE THESE */
        EXCLUDE(DSN EQ A.B.**.G>>>>.V>>)/* EXCLUDE ALL OTHER GENS */
        INCLUDE(DSN EQ *.**)            /* INCLUDE ALL OTHERS THAT*/
                                        /* MATCH
                                        /* THE CANDIDATES MASK */
        )
```

Aggregate LoadBalancer

You can use the Aggregate LoadBalancer to distribute the backup and restore load across multiple Selection Data Sets, combine multiple Selection Data Sets into one, and tailor individual Selection Data Sets. The Selection Data Sets can contain data set names in any backup control card format created by the Automated Critical Data Identification APPLEND function.

The Aggregate LoadBalancer is a batch procedure that provides the following functionality:

- You can use it to distribute the backup or restore processing required for the data sets in a single Selection Data Set across multiple Selection Data Sets to take advantage of concurrent backup capabilities and minimize the impact of a backup failure.
- You can also use the Aggregate LoadBalancer program to combine multiple Selection Data Sets into a single Selection Data Set, which can reduce tape mounts that result from processing numerous small backups.
- You can use the Aggregate LoadBalancer program to alter the data set name list for one or more Selection Data Sets. This method is similar to using the Edit Selection Data Set line command, but using the LoadBalancer can help you avoid making syntax errors in the Selection Data Set and you can use it in conjunction with Selection Data Set balancing or combination (or both).

Note: Be advised that when you use the LoadBalancer to split or combine Selection Data Sets, other Advanced Backup and Recovery for z/OS features, such as Automated Critical Data Identification, are unaware of the changes.

About the Aggregate LoadBalancer and Incremental ABARS

Aggregate LoadBalancer and Incremental ABARS are incompatible. Incremental ABARS is specified at the aggregate level and once specified, it tracks all data sets for that aggregate. Aggregate LoadBalancer moves these data sets to other aggregates, based on size or number of data sets. Once moved, Incremental ABARS has no way to associate the data sets with the original aggregate.

Using the Aggregate LoadBalancer

To use the Aggregate LoadBalancer batch procedure successfully, you need to understand its job stream and JCL.

About the Aggregate LoadBalancer job stream

The BKMALB01 job stream executes procedure BKMALBP, which consists of two required steps and one optional step. These steps are described in this topic, along with some usage notes for the program.

- Step 1 executes program BKM00001 and processes user control statements from DD name SYSIN, specifying whether to balance or combine (or both), how balancing or combining should be done, and

the input and output aggregates or Selection Data Sets (or all). An extract file is created containing the entries from one or more input Selection Data Sets in a readily modifiable format.

- Step 2 is an optional step that is executed *only* if in-stream user code will be used to modify the extract file. Sample REXX code is provided to demonstrate possible distribution techniques. If you are using this step, change the value for symbolic parameter ALBPGNM from IEFBR14 or IKJEFT01 to execute a batch TSO session and change SYSEXEC to the name of your REXX EXEC library.
- Step 3 executes program BKM00002 and reads the extract file from step 1, or the modified extract file from step 2, to create one or more Selection Data Sets. The extract file input to step 3 can be any file you want to use, if the format is valid.

Modifying the extract file

Step 1 produces an extract file, which can be modified by user code in the optional Step 2. Any modifications made to the extract file during Step 2 will affect the Selection Data Sets subsequently created during Step 3 processing. The point of altering the extract file, or re-creating it, is to force the data set names to specific Selection Data Sets.

Keep in mind that changes made during Step 2 processing are not reflected in the Automated Critical Data Identification database. Also be aware that if you want to filter out certain data sets, it is better to do so using the Automated Critical Data Identification filter options because any filtering done using these options is repeated on subsequent verifications.

Note: If the modifications you make to the extract file in some way eliminate data set names, checking the integrity of the backup is your responsibility. You can find REXX examples, used to modify the extract file, in the Advanced Backup and Recovery for z/OS SBKMRSMP library; however, be aware that these samples are not meant to be used unmodified; they are provided as samples only.

Extract file record format

The input record has been parsed into the following variables and format:

Table 6. Input record variables and format

Column	Length	Variable	Variable Description
1	1	REC_TYPE	1–Control 2–Include 3–Allocate 4–Accompany
3	44	REC_DSN	Data Set Name, 44 bytes
48	1	REC_NVSAM_OR_VSAM	N–Non-VSAM V–VSAM
49	1	REC_DISK_OR_TAPE	D–DASD, including ML1 DASD T–TAPE 2–ML2
53	8	REC_LAST_REFDAT	Date on which the data set was last referenced, in YYYYDDD format
62	6	REC_VOLSER	Volser, 6 bytes

Table 6. Input record variables and format (continued)			
Column	**Length**	**Variable**	**Variable Description**
68	1	REC_MULTIVOL	Multivolume flag, 1 byte. An asterisk in this position means REC_VOLSER is the first of a multivolume data set.
70	8	REC_SIZE	Size of DASD, migrated, and user tape data sets, in kilobytes
78	1	Constant that should not be used.	Character K
80	54	REC_SELNAME	Output Selection Data Set
135	8	REC_AGGNAME	Output aggregate name
144	2	REC_FROM_TYPE	A: –An A followed by a colon indicates an aggregate. D: –A D followed by a colon indicates a Selection Data Set.
146	54	REC_FROM_NAME	Aggregate name or Selection Data Set name
201	8	REC_REMAINDER	Unused
210	2	REC_DSORG	Data set organization (IS, PS, DA, PO, VK, VR, VL, VE, or U)
213	4	REC_RECFM	Data set record format (U, F, FB, FBA, FBM, V, VB, or VBS)
218	8	REC_LRECL	Data set record length, in bytes
227	8	REC_BLKSIZE	Data set block size, in bytes
236	8	REC_EXPIRE_DATE	Data set expiration date, in YYYYDDD format

How does BKM00001 handle data set size values?

During Step 1, the current size of each data set is retrieved. If the DFSMShsm MCDSs are made available to the BKM00001 program, the data set size for each migrated data set is the size while migrated. If the DFSMShsm MCDSs are not available, the size of the migrated data sets is set to zero.

How does BKM00002 handle ML2 data sets from the same ML2 tape volume?

The BKM00002 program attempts to keep all data set names residing on the same ML2 tape volume together in a single output Selection Data Set. Similarly, data set names for tape data sets residing on the same tape volume are kept together in the same Selection Data Set. These situations can cause an imbalance when the data set names are being distributed by count or by total size.

How does BKM00002 handle multifile tape volumes?

BKM00002 attempts to keep all tape data set names from the same multifile tape volume together in a single output Selection Data Set.

How does BKM00002 handle Generation Data Sets?

BKM00002 attempts to keep all Generation Data Set names related to the same Generation Data Group together in a single output Selection Data Set.

Modifying the Aggregate LoadBalancer JCL

You need to modify the job card parameters and provide the appropriate control cards before you submit BKMALB01 for execution. Refer to the text in BKMALB01 for the change options and sample control cards. This job uses the JCL procedure BKMALBP in the product JCL library.

The following figure shows the BKMALBP procedure:

```
******************************** Top of Data ************************
//BKMALBP  PROC ALBPGNM=IEFBR14,
//       DISK=SYSALLDA,
//       LOAD1=?hlq.SBKMLOAD
//       PARMLIB=?hlq.SBKMPARM
//       REXXEXEC=BKMREXX1,
//       SOUT='*',
//       SYSEXEC=NULLFILE
//*
//S1       EXEC PGM=BKM00001,REGION=0M
//SYSPRINT DD SYSOUT=&SOUT
//STEPLIB  DD DISP=SHR,DSN=&LOAD
//MSCPARM  DD DISP=SHR,DSN=&PARMLIB(BKMINI)
//SYSPRINT DD SYSOUT=&SOUT
//WORKLOG  DD SYSOUT=*
//SYSUDUMP DD SYSOUT=&SOUT
//SYSUT2   DD DSN=&EXTRDSN,
//            UNIT=&DISK,SPACE=(TRK,(10,3)),
//            DISP=(NEW,PASS,DELETE)
//*
//***********************
//* OPTIONAL STEP       *
//***********************
//S2       EXEC PGM=&ALBPGNM,PARM='REXXEXEC'
//STEPLIB  DD  DISP=SHR,DSN=&LOAD
//SYSTSIN  DD  DUMMY
//SYSPRINT DD  SYSOUT=&SOUT
//SYSTSPRT DD  SYSOUT=&SOUT
//SYSEXEC  DD  DISP=SHR,DSN=&SYSEXEC       <== USED FOR REXX CODE LIB
//INDD     DD  DSN=&EXTRDSN,DISP=(OLD,PASS)
//*
//S3       EXEC PGM=BKM00002
//STEPLIB  DD  DISP=SHR,DSN=&LOAD1
//MSCPARM  DD  DISP=SHR,DSN=&PARMLIB(BKMINI)
//SYSPRINT DD  SYSOUT=&SOUT
//SYSOUT   DD  SYSOUT=&SOUT
//SYSUDUMP DD  SYSOUT=&SOUT
//SORTIN   DD  DSN=&EXTRDSN,DISP=(OLD,DELETE)
//         PEND
//*
```

The following figure shows an example of the Aggregate LoadBalancer job JCL:

```
//JOBCARD  JOB (ACCT),'BKMALB01',CLASS=A           <<< SEE #1
// JCLLIB ORDER=?hlq.SBKMJCL                 <<< SEE #2
//*******************************************************************/
//* SAMPLE JCL FOR AGGREGATE LOAD BALANCER                          *
//*******************************************************************/
//* 1. CHANGE JOB STATEMENT TO LOCAL STANDARDS                      */
//* 2. CHANGE ALL ?HLQ. TO LOCAL STANDARDS                          */
//*******************************************************************/
//* STEPS 1 AND 3 ARE REQUIRED.                                     *
//* STEP  2 IS OPTIONAL:                                            *
//*         IT CAN BE USED IF ADDITIONAL MANIPULATION OF THE        *
//*         BALANCING PROCESS ID REQUIRED.                          *
//*         NOTE: THERE ARE SAMPLE REXX EXECS IN THE PRODUCT        *
//*               SBKMRSMP LIBRARY                                  *
//*******************************************************************/
//BKMALB01 EXEC BKMALBP
//S1.SYSIN     DD  *
DISTRIBUTE(BY_TOTAL_KB (1000000))               -                                                -
INPUT(  AGGR (AGGR1                             -
              AGGR2    )                        -
       SELDSN(SELDSN1  )    )                   -
                                                -
OUTPUT( AGGR (AGGR3                             -
              AGGR4    )                        -
       SELDSN(SELDSN2                           -
              SELDSN3  )    )
******************************** Bottom of Data ********************
```

DISTRIBUTE control statements

Control statements are read using the DD name SYSIN.

DISTRIBUTE control statement syntax

All parameters are specified by keywords with their associated values. Enclose the keyword values in parentheses. Because the entire parameter list constitutes one logical statement, you must end each physical line (except the last) with a dash as the continuation character.

```
DISTRIBUTE ({BY_TOTAL_KB|MB|GB|TB(nnnnnnnn)|BY_COUNT})     +
           INPUT(SELDSN(seldsn1…seldnsn)|                   +
                AGGR(agg-name1…agg-namen))|                +
           OUTPUT(SELDSN(seldsn1…seldnsn)|                  +
                AGGR(agg-name1…agg-namen))|                +
           OUTFORMAT(ABARS|DFDSS|DFDSSL(DSSL)|DMSCADISK|FDR|LIST)   +
           RC4TORC0
```

Keyword descriptions

DISTRIBUTE

This is a required keyword for which you must specify either the **BY_COUNT** or the **BY_TOTAL_KB|MB|GB|TB** subparameter.

BY_COUNT

Use this parameter to specify that you want the data set names from all input Selection Data Sets distributed evenly across all the output Selection Data Sets. In situations where there are fewer data set names than there are output Selection Data Sets, not all output Selection Data Sets are used. Because all Generation Data Set names related to the same Generation Data Group and data set names for tape data sets residing on the same tape volume are kept together in a single output Selection Data Set, a Selection Data Set count imbalance might occur the data set names are distributed by count.

You must specify either **BY_COUNT** or **BY_TOTAL_KB|MB|GB|TB** as a subparameter of the **DISTRIBUTE** keyword.

BY_TOTAL_KB|MB|GB|TB

Use this parameter to specify the maximum number of data set kilobytes, megabytes, gigabytes, or terabytes to be placed in each output Selection Data Set. By relating size to time (estimate how much data can be copied to tape in a specific amount of time), using this balancing option confines each backup to some specific duration. The size limit per Selection Data Set will be exceeded if you specify an insufficient number of output Selection Data Sets to keep the distribution processing within the requested size limits. The output Selection Data Sets are 'topped off' by adding the names of smaller and smaller data sets to each Selection Data Set. Because all Generation Data Set names related to the same Generation Data Group and data set names for tape data sets residing on the same tape volume are kept together in a single output Selection Data Set, a Selection Data Set size imbalance might occur when the data set names are distributed by total size.

You must specify either **BY_COUNT** or **BY_TOTAL_KB|MB|GB|TB** as a subparameter of the **DISTRIBUTE** keyword.

Note: Regardless of whether balancing is done by count or size limit, if an aggregate or Selection Data Set name is placed in the extract file by the optional Step 2 REXX routine, the INCLUDE data set is forced to the chosen output aggregate or Selection Data Set.

INPUT(...)

Use this keyword to specify the input aggregates, Selection Data Sets, or a combination of these. Be aware that your installation's version of ABARS might not permit the use of PDSEs for Selection Data Sets; however, PDSEs are supported by programs BKM00001 and BKM00002.

Note: You must specify at least one **AGGR** or one **SELDSN** subparameter for the **INPUT** keyword. For **SELDSN**, the data set name list in the Selection Data Set can be in any format produced by the Automated Critical Data Identification APPLEND function (ABARS, DFDSS, DFDSSL (DSSL), DMS, CADISK, FDR, or LIST).

OUTPUT(...)
Use this keyword to specify the output aggregate names or Selection Data Set names.

Note: You must specify at least one **AGGR** or one **SELDSN** subparameter for the **OUTPUT** keyword.

OUTFORMAT(ABARS|DFDSS|DFDSSL (DSSL)||DMSCADISK|FDR|LIST)
Use this keyword to specify the output Selection Data Set format. Each Selection Data Set written contains the data set names in the format supported by the chosen backup product.

RC4TORC0
Use this keyword to override an RC=4 returned from the S1 step (program BKM00001) with an RC=0 to avoid interfering with subsequent processing. This keyword requires no sub-parameters. **RC4TORC0** cannot be used to override return codes with values other than 4.

AGGR(*agg-name1 ...agg-namen*)
Use this subparameter of the **INPUT** or **OUTPUT** keyword to identify the input or output (respectively) aggregate names. You must specify at least one aggregate name for this subparameter. Use one or more blank spaces to separate multiple aggregate names from each other. At least one **AGGR** or **SELDSN** subparameter must be specified for both INPUT and OUTPUT keywords.

AGGR and **SELDSN** may be specified together as sub-parameters of both the **INPUT** and **OUTPUT** keywords.

Note: For either **INPUT** or **OUTPUT**, the entire list of aggregate names and Selection Data Set names you specify cannot exceed 4000 bytes in length.

SELDSN(*sel-dsn1 ...sel-dsnn*)
Use this subparameter of the **INPUT** or **OUTPUT** keyword to identify the input or output (respectively) Selection Data Set names. You must specify at least one Selection Data Set name for this subparameter. Use one or more blank spaces to separate multiple Selection Data Set names from each other. At least one **AGGR** or **SELDSN** subparameter must be specified for both **INPUT** and **OUTPUT** keywords.

SELDSN and **AGGR** may be specified together as sub-parameters of both the **INPUT** and **OUTPUT** keywords only.

Note: For either **INPUT** or **OUTPUT**, the entire list of Selection Data Set names and aggregate names you specify cannot exceed 4000 bytes in length.

Usage notes for INPUT and OUTPUT aggregate names

Aggregate names must resolve to Selection Data Set names.

Both BKM00001 and BKM00002 programs look up SMS aggregate group definitions to resolve to Selection Data Set names. You must define the aggregate names to SMS before you can use them with the INPUT and OUTPUT keywords.

Using the same Selection Data Set for INPUT and OUTPUT is not recommended.

Note:

- The Aggregate LoadBalancer allows you to specify the same Selection Data Set name for both the **INPUT** and **OUTPUT** keywords. If there is only one output Selection Data Set, no combining or splitting of Selection Data Sets occurs. In this situation, the LoadBalancer determines data set sizes, volsers, migration status, and checks for uncataloged data sets. Because native ABARS fails a backup if an uncataloged data set is referenced in the Selection Data Set, this technique can improve the reliability of the ABACKUP process, as well as providing additional information (data set size, location, and migration status) that is useful during backup processing analysis.

  However, if the only Aggregate LoadBalancer function needed is the catalog check to eliminate uncataloged data sets from Selection Data Sets, it is recommended that you use the ABARS Manager Setup and Configuration option for removing uncataloged data sets. You can set this option for specific aggregates or let it default for all aggregates using a BKMINI token. For best results, catalog locates, which ABARS must do anyway, are time-consuming and should be avoided whenever possible.

- If the extract file from Step 1 is not altered, all INCLUDE data sets are accounted for in the OUTPUT, unless the program reports an error and terminates with a nonzero return code.

  However, if the extract file is altered, it is recommended that you keep the INPUT Selection Data Sets intact, should you choose to verify that all INCLUDE data sets are accounted for in the OUTPUT.

Using the sample Aggregate LoadBalancer REXX EXEC

You can use the sample REXX EXEC (BKMREXX1, BKMREXX2, or BKMREXX3), which is supplied in the SBKMRSMP library, as a prototype for other EXECs you might want to write.

Sample program BKMREXX1 contains an internal subroutine called PROCESS_INCLUDE. This subroutine receives control for each INCLUDE data set in the extract file. The sample forces data sets matching a mask of 'RMP.ACDISMF.**' to the first output Selection Data Set by using an IBM-supplied ACS (Automatic Class Selection) filter routine. Note the output Selection Data Sets and aggregate names are stored in stem variables OUTPUT.1 through OUTPUT.*nn*. By using multiple ACS masks, you can key on one or more masks. Although the example forces data sets to just the first Selection Data Set, you can use OUTPUT.1 through OUTPUT.*nn* (OUTPUT.0 = total number of output Selection Data Sets) to spread the INCLUDE data sets to your choice of Selection Data Sets.

Please call our support group for assistance if you want to use this user routine. We are interested in uses you might find for this program. We might also be able to provide you with other sample REXX EXECs that perform balancing with exceptions that others have found useful.

Performing incremental backups and recoveries

The Advanced Backup and Recovery for z/OS ABARS Manager feature includes incremental backup and recovery capability.

An incremental backup takes a full (base) backup of an aggregate and then backs up those data sets that have changed (or are new to the aggregate) since the last full or incremental backup. This means that the full backup and all incremental backups must be retained for recovery. An aggregate with incremental backups must be recovered in such a way that the latest copy of any data set takes precedence over any previous backups.

Advantages and disadvantages

Incremental backups are useful if ignoring unchanged data sets during a backup can provide a significant reduction in backup time. This time savings comes at the expense of vaulting tape copies, which by design contain obsolete versions of data sets, and at the expense of increased recovery time. Incremental backup tapes might not be efficiently used if the amount of changed data is small, and this inefficiency might impact your organization's device usage objectives.

If the bulk of changes to an aggregate are new data sets created in each cycle, incremental backup times are probably the best you can get. Although many incremental backup tapes must be retained until the next base backup, the majority of data sets within the incremental backups are not wasted (obsolete) space. Hence, the only drawback to performing an incremental backup in this circumstance is the physical number of tapes required to maintain a current base and all intervening incremental backups.

Using ABARS Manager to run incremental backups

This topic provides a procedure for performing incremental backups using ABARS Manager and cleaning up the Data Set Activity Database (DAD).

Before you can run an incremental backup, you must set keyword values in the `:PRODUCT_SECURITY` and `:BKMSTSKD_OPTIONS` sections of the BKMINI member of the Advanced Backup and Recovery for z/OS product parameter library.

1. Define an aggregate to ISMF.
2. Activate the aggregate to SMS.
3. Add the aggregate to ABARS Manager.

4. Identify the aggregate as incremental and set the **Backup Gens/Groups to Retain** value.
5. Submit the **ABACKUP**.
6. Remove obsolete entries from the Data Set Activity Database (DAD).

Step 1: Defining an aggregate to ISMF

The first step in running an incremental backup using ABARS Manager is to create a new SMS management class for the aggregate and set its retention attributes appropriately.

You cannot use ABARS Manager to run incremental backups until you set appropriate keyword values in the `:PRODUCT_SECURITY` and `:BKMSTSKD_OPTIONS` sections of the BKMINI member in the product parameter library.

For incremental backups, ABARS Manager expires from the IDS aggregates as a group (base and incrementals) when the retention attribute limit is reached. In addition, ABARS Manager uses the DFSMShsm **EXPIREBV** command to expire the rolled-off ABARS backup versions. The DFSMS (SMS) management class assigned to this aggregate must have the retain versions set to 9999 or NOLIMIT.

The following figure shows an example of such an SMS management class.

```
CDS Name  . . . . . . . . . . : ACTIVE
       Management Class Name . . . : ABARSNOL

       AGGREGATE Backup Attributes:
           # Versions  . . . . . . . : 9999
           Retain only Version . . :
             Unit  . . . . . . . . . :
           Retain extra Version  . :
             Unit  . . . . . . . . . :
           Copy Serialization  . . : CONTINUE
           ABackup Copy Technique  : STANDARD
```

Step 2: Activating the aggregate to SMS

After you have defined the aggregate to ISMF, activate it to SMS.

Step 3: Adding the aggregate to ABARS Manager

After you have defined the aggregate to ISMF and activated it to SMS, use the **Aggregate Management with SMS Interface** panel to add the aggregate to ABARS Manager.

1. Type 1 in the **Command** line on the Advanced Backup and Recovery for z/OS **Main Menu**, and then press Enter to display the **Setup and Configuration Menu** panel.
2. Type 1 in the **Command** line on the **Setup and Configuration Menu** panel, and then press Enter to display the **Aggregate Management with SMS Interface** panel.
3. On the **Aggregate Management with SMS Interface** panel, type A in the **C** column next to the name of the aggregate for which you intend to run an incremental backup and then press Enter.

   The software displays the **ABARS Manager Aggregate Controls** panel.

Step 4: Identifying the aggregate as incremental and setting the Backup Gens/Groups to Retain value

After you have defined the aggregate to ISMF, activated it to SMS, and added it to ABARS Manager, use the **Aggregate Controls** panel to identify the aggregate as incremental and set its **Backup Gens/Groups to Retain** value.

Be aware that you cannot complete this step until after the aggregate has been added to ABARS Manager and it appears on the list displayed by the **Aggregate Management with SMS Interface** panel.

1. If the **Aggregate Controls** panel is already displayed, go directly to step 2. Otherwise, do the following:
   a) On the Advanced Backup and Recovery for z/OS **Main Menu** panel, type 1 in the **Command** line and press Enter to display the **Setup and Configuration Menu** panel.

b) On the **Setup and Configuration Menu** panel, type 1 in the **Command** line and press Enter to display the **Aggregate Management with SMS Interface** panel.

c) On the Aggregate Management with SMS Interface Setup panel, type an asterisk (*) in the **Aggregate Name or Mask** field and press Enter to display the **Aggregate Management with SMS Interface** panel.

d) In the **C** column next to the appropriate aggregate's name, type C and press Enter to display the **Aggregate Controls** panel.

2. On the **Aggregate Controls** panel, do the following:

a) In the **Identify as Incremental** field, type Y.

b) Use the **Backup Gens/Groups to Retain** field to specify the number of generations or groups you want retained before ABARS Manager expires the oldest.

Be aware that ABARS Manager considers the base backup plus all of its incremental backups to be one generation or group.

Consider an example in which you specify that you want two generations or groups retained by ABARS Manager. ABARS Manager would retain the first base backup its five subsequent incremental backups as one generation or group. ABARS Manager would retain the second base backup and all of its subsequent incremental backups as the second generation or group. When the third base backup is run, ABARS Manager expires the first base backup and all of its associated incremental backups; the second and third base backups and their incremental backups are retained.

c) Save your specifications and close the **Aggregate Controls** panel.

When the **Aggregate Management with SMS Interface** panel is displayed, it shows Y in the INCR column for the aggregate.

Step 5: Submitting the ABACKUP

After you have defined the aggregate to ISMF, activated it to SMS, added it to ABARS Manager, and identified it as an incremental aggregate and set is Backup Gens/Groups to Retain attribute appropriately, you can submit the ABACKUP.

Before you can submit the ABACKUP, you must have data in the Selection Data Set for the aggregate. You can create the Selection Data Set, or you can use one created by the Automated Critical Data Identification feature.

A base backup must be taken for the aggregate before you can take an incremental backup. When the base backup is run, the data sets in the Selection Data Set are recorded in the Data Set Activity Database (DAD) so that any changes to those data sets and any new data sets that are added to the aggregate as a result of a modified Selection Data Set are tracked by ABARS Manager. These changed and new data sets are included in the next incremental data sets.

1. On the **Aggregate Management with SMS Interface** panel, type B in the **C** column next to the aggregate against which you want to run an incremental backup and press Enter.
2. When the **Submit Backup** panel appears, type B in the **Incremental or Base** field.
3. Use the **Submit Backup** panel's online help to guide you as you complete the rest of the fields on this panel and then press Enter.

   After a base backup has been run successfully for this aggregate, you can run incremental backups for it.

Removing obsolete entries from the DAD database

Data set names are recorded in the DAD database when they are found to be new or to have changed. Some of these entries in the DAD database become obsolete and should be cleaned up as part of normal maintenance.

Obsolete entries can exist in the Data Set Activity Database (DAD) for any of these reasons:

- They are not in the current Base/Incremental (Incremental ABARS).

- They are old generation data sets (GDG) that haven't been deleted. For Incremental ABARS, they are no longer in the current Base/Incremental.
- They have been deleted.

The PURGE_UNCATLG = Y parameter in the BKMINI member of the product parameter library automatically deletes entries from the DAD database when an SMF delete is issued for the data set, but note that the other entries are not deleted automatically.

The BKM9254M JCL member shown in the following figure is provided in the Advanced Backup and Recovery for z/OS JCL library. BKM9254M removes obsolete entries from the DAD database. Use this program on a regular basis with DELETE OLDGDGS to remove obsolete GDG entries from the DAD database. The DAD database does not need to be reorganized after you run BKM9254M.

```
//*******************************************************************
//*  MAKE THE FOLLOWING CHANGES:
//*    1. CHANGE JOB CARD TO LOCAL STANDARDS
//*    2. CHANGE PARMLIB= THE BKM PARMLIB NAME
//*    2. CHANGE LOAD1= TO THE BKM AUTHORIZED LIB NAME
//*******************************************************************
//JOBNAME  JOB ,'...',CLASS=?,MSGCLASS=?         <=== SEE NOTE 1
//BKM9254M PROC PARMLIB=prefix.value.SBKMPARM,   <=== SEE NOTE 2
//        LOAD1=prefix.value.SBKMLOAD            <=== SEE NOTE 3
//S1      EXEC PGM=BKM9254M
//STEPLIB  DD  DISP=SHR,DSN=&LOAD1
//MSCPARM  DD  DISP=SHR,DSN=&PARMLIB(BKMINI)
//SYSPRINT DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//        PEND
//*
//BKM9254M EXEC BKM9254M
//SYSUDUMP DD  SYSOUT=*
//SYSIN    DD  *
```

The valid control cards are as follows:

```
DELETE OLDGDGS
DELETE NOT-IN-CAT
DELETE SCRATCHED
DELETE NAME(...)
DELETE NAME(...) NOT-IN-CAT
DELETE NAME(...) SCRATCHED
```

DELETE
This is a required keyword. You must also specify at least one of the other keywords.

OLDGDGS
Specify this keyword to remove uncataloged GDSs. This is an optional keyword.

NOT-IN-CAT
Specify this keyword to remove uncataloged data sets.

SCRATCHED
Specify this keyword to remove deleted data sets.

NAME
Specify this keyword to identify a data set name, either as a mask (`NAME(PAY.WEEK01.G>>>>V00)NOT-IN-CAT`, for example) or as a literal name.

Backing up the IDS

It is strongly recommended that you copy the Advanced Backup and Recovery for z/OS Inventory Data Set (IDS) and ship it with each vault shipment. For electronic vaulting scenarios, copy the IDS after each aggregate backup and vault the copy along with the aggregate.

Although you can use native ABARS to restore aggregates backed up under the control of ABARS Manager, using that method to recover incremental backups is awkward and error-prone. Without a corresponding backup inventory that is current with respect to incremental backups present at the vault, incremental recovery is impractical. The correct restoration of generation data sets is especially

challenging when restoring from multiple generations. The IDS contains information and related code to ensure active generations are restored successfully.

Incremental backups are merely backup generations of the aggregate. When the incrementally backed up data sets are restored under the control of ABARS Manager, the sequence of the restores is known, and subsequently the **ARECOVERs** are done in the correct sequence. As with non-incremental ABARS, you specify the appropriate **REPLACE** or **NOREPLACE** option because the recovery analysis starts with the most current incremental backup and works backward to the base backup. This allows ABARS Manager to select the most current backup for the data sets to be recovered and skip the older ones, eliminating the **REPLACE** methodology that occurs when the base is restored first and then the incrementals are applied. The **ARECOVER** execution restores the data from the base backup to the incremental, ensuring that generation data sets are restored in the correct order; the recovery analysis eliminated all the old data sets.

You can recover generations of an aggregate directly with ABARS; however, if you use this method, ensuring that these generations are recovered in the correct sequence is your responsibility.

Chapter 20. Using CATSCRUB to synchronize catalogs and DASD volumes

Use the **CATSCRUB** command to synchronize quickly one or more BCS catalogs with the associated DASD volumes at your disaster recovery site during a full volume restore. You can use ABARS Manager to help you build your **CATSCRUB** command.

About CATSCRUB

The **CATSCRUB** command includes a comprehensive facility to ensure that ICF catalogs and data volumes are synchronized when a system is recovered at a disaster recovery site.

The command's simulation capability lets you see the expected deletions before you run **CATSCRUB** in EXECUTE mode. Using simulation in conjunction with MATCH-VOLSER creates an environment at the home site to ensure that the process functions as expected at the disaster recovery site.

Note:

- Running **CATSCRUB** in SIMULATE mode before using it in EXECUTE mode is always advisable. It is also important that you read and understand the information in the next section before you use **CATSCRUB** in EXECUTE mode.
- When you run **CATSCRUB** in BATCH mode, for each user catalog selected by user keywords, CONTROL access is checked using the catalog's VOLSER. If a catalog cannot be located or does not have CONTROL access, a status message is written and that catalog is not scrubbed.

How does CATSCRUB processing work?

Every time you run **CATSCRUB** in EXECUTE mode, the following events occur:

- All of the anomaly keyword values used, as specified by the command, the contents of BKMINI, and the program defaults, are listed.
- When MSGTEXT(FULL) is specified, all of the DASD volumes to be specified for MATCH-VOLSER or EXCLUDE-MATCHVOLSER are listed.
- For each catalog to be scrubbed, the following information is listed:
  - The catalog entries with RC value > 0 in corresponding anomaly keywords
  - The catalog entries deleted with EXECUTE or simulated to be deleted with SIMULATE
  - A summary of catalog entries deleted.
- A summary of catalog entries scrubbed from all catalogs processed is generated.

Building a CATSCRUB command

You can use the ABARS Manager **Submit CATSCRUB** panel to help you build a **CATSCRUB** command.

1. Type 4 in the **Command** line on the **Main Menu** panel and then press Enter to display the Features panel.
2. Type 1 in the **Command** line on the **Features** panel and then press Enter to display the **Submit CATSCRUB** panel.
3. Use the **Submit CATSCRUB** panel's online help to guide you as you complete the panel's fields.

About the Submit CATSCRUB panel

Use the **Submit CATSCRUB** panel to help you to build a **CATSCRUB** command.

The following figure shows the **Submit CATSCRUB** panel.

```
                      BKM - Submit CATSCRUB
 Command ===>

 (S) SELECT AN OPTION:
    Build JCL/Edit for Submit
                                                                  More:     +
Required keywords:
  (S) Selected List    => No catalog(s) selected
SIMULATE               => Y      Y or N (Execute)

Optional keywords:
EXCLUDE-BCS(..)        => N      Y or N
EXCLUDE-KEEP-DSN       => N      Y or N
EXCLUDE-KEEP-TAPE      => N      Y or N
EXCLUDE-KEEP-VOLSER    => N      Y or N
EXCLUDE-MATCH-VOLSER   => N      Y or N
KEEP-DSN               => N      Y or N
KEEP-TAPE              => N      Y or N
KEEP-VOLSER            => N      Y or N
MATCH-VOLSER           => N      Y or N
MESSAGE-TEXT           => F      A(bbreviated) or F(ull)

Anomalies:                       D(elete) or       Return        C(ontinue)
                                   K(eep)           Code         or E(nd)
FATAL-CATALOG-ERR:                                                   E
GDGBASE-NO-ACTIVE-GENS:             K               4                C
GDS-MIGRATED:                       K               4                C
GDS-MIGRATED-DASD:
GDS-MIGRATED-TAPE:
GDS-MULTI-VOL-ERR:                  K               4                C
GDS-NOT-FOUND:                      K               4                C
GDS-TAPE:                           K               4                C
NONVSAM-ALIAS-NO-REALNAME:          K               4                C
NONVSAM-MIGRATED:                   K               4                C
NONVSAM-MIGRATED-DASD
NONVSAM-MIGRATED-TAPE
NONVSAM-MULTI-VOL-ERR:              K               4                C
NONVSAM-NOT-FOUND:                  K               4                C
NONVSAM-TAPE:                       K               4                C
OBJECT-ACCESS-METHOD:               K               4                C
VSAM-SPHERE-MIGRATED:               K               4                C
VSAM-SPHERE-MIGRATED-DASD:
VSAM-SPHERE-MIGRATED-TAPE:
VSAM-SPHERE-MULTI-VOL-ERR:          K               4                C
VSAM-SPHERE-NOT-FOUND:              K               4                C
```

The following fields and controls are available on the **Submit CATSCRUB** panel:

Build JCL/Edit for Submit
Use this field to specify whether you want to just validate the other data in the dialog or to validate the data and create the JCL. Type S to generate the JCL based on the dialog data you choose on the panel. You can edit the JCL before submitting it.

(S) Selected List
Type S to select from a pop-up list of BCSs connected to the current master catalog. You can select one or more BCSs from the list. When you dismiss the pop-up dialog, `No catalog(s) selected` is replaced by the number of catalogs selected (for example, `3 Catalog(s) selected`). If you do not type S in this field, the required keyword is generated in the JCL; however, no arguments are generated. Instead, when the JCL is displayed for editing or submission, you can add argument(s) at that time.

SIMULATE
Use this field to specify whether you want to execute the JCL or use it to simulate a **CATSCRUB**. Specify N (default) to submit the **CATSCRUB**. Specify Y to generate the required keyword in the JCL so you can simulate the **CATSCRUB** without having the process actually performed.

EXCLUDE-BCS(. .)
N is the default. Specify Y to generate the required keyword in the JCL, without generating any arguments. Instead, when the JCL is displayed for editing or submission, you can add argument(s) at that time.

Code your argument as data set names or masks. You can specify a mask as a value for the BCS name, and then selectively exclude unwanted catalogs from the list of catalogs that match your mask. You

can specify standard mask characters as well as the extended ACS mask characters. Only BCSs that have been selected by your mask can then be effectively excluded. A SIMULATE mode of execution shows which catalogs would be processed by an actual execution.

EXCLUDE-KEEP-DSN
EXCLUDE-KEEP-DSN works in conjunction with the KEEP-DSN keyword. EXCLUDE-KEEP-DSN specifies one or more data set names or masks that are to be excluded from the list of data set names as specified by the KEEP-DSN keyword. EXCLUDE-KEEP-DSN is processed after KEEP-DSN.

N is the default value. Specify Y to generate the required keyword in the JCL, without generating any arguments. When the JCL is shown for editing or submission, you can add arguments at that time. You can specify standard mask characters as well as the extended ACS mask characters.

EXCLUDE-KEEP-TAPE
EXCLUDE-KEEP-TAPE works in conjunction with the KEEP-TAPE keyword. EXCLUDE-KEEP-TAPE specifies one or more tape serial numbers or masks that are to be excluded from the list of serial numbers as specified by the KEEP-TAPE keyword. EXCLUDE-KEEP-TAPE will be processed after KEEP-TAPE.

N is the default value. Specify Y to generate the required keyword in the JCL, without generating any arguments; when the JCL is shown for editing or submission, you can add arguments at that time. You can specify standard mask characters as well as the extended ACS mask characters.

EXCLUDE-KEEP-VOLSER
EXCLUDE-KEEP-VOLSER works in conjunction with the KEEP-VOLSER keyword. EXCLUDE-KEEP-VOLSER specifies one or more disk serial numbers or masks that are to be excluded from the list of serial numbers as specified by the KEEP-VOLSER keyword. EXCLUDE-KEEP-VOLSER is processed after KEEP-VOLSER.

N is the default value. Specify Y to generate the required keyword in the JCL, without generating any arguments; when the JCL is shown for editing or submission, you can add arguments at that time. You can specify standard mask characters as well as the extended ACS mask characters.

EXCLUDE-MATCH-VOLSER
EXCLUDE-MATCH-VOLSER works in conjunction with the MATCH-VOLSER keyword. MATCH-VOLSER specifies those volumes to be synchronized with the catalogs. This feature can be very useful when testing the DR process in SIMULATE mode before moving to the DR site. EXCLUDE-MATCH-VOLSER specifies one or more disk serial numbers or masks that are to be excluded from the list of serial numbers as specified by the MATCH-VOLSER keyword. You can specify standard mask characters as well as the extended ACS mask characters. EXCLUDE-MATCH-VOLSER is processed after MATCH-VOLSER.

N is the default value. Specify Y to generate the required keyword in the JCL, without generating any arguments; when the JCL is shown for editing or submission, you can add arguments at that time.

KEEP-DSN
KEEP-DSN works in conjunction with the EXCLUDE-KEEP-DSN keyword, if specified. KEEP-DSN specifies one or more data set names or masks that are to be excluded from **CATSCRUB** delete processing. You can specify the standard mask characters as well as the extended ACS mask characters.

N is the default value. Specify Y to generate the required keyword in the JCL, without generating any arguments; when the JCL is shown for editing or submission, you can add arguments at that time.

KEEP-TAPE
KEEP-TAPE works in conjunction with the EXCLUDE-KEEP-TAPE keyword, if specified. KEEP-TAPE specifies one or more tape serial numbers or masks that are to be excluded from NONVSAM-TAPE(DELETE) and GDS-TAPE(DELETE) processing. You can specify standard mask characters as well as the extended ACS mask characters.

N is the default value. Specify Y to generate the required keyword in the JCL, without generating any arguments; when the JCL is shown for editing or submission, you can add arguments at that time.

KEEP-VOLSER
KEEP-VOLSER works in conjunction with the EXCLUDE-KEEP-VOLSER keyword, if specified. KEEP-VOLSER specifies one or more disk serial numbers or masks that are to be excluded from the delete

processing of the disk-oriented anomalies. You can specify standard mask characters as well as the extended ACS mask characters.

N is the default value. Specify Y to generate the required keyword in the JCL, without generating any arguments; when the JCL is shown for Edit/Submission, you can add arguments at that time.

MATCH-VOLSER
MATCH-VOLSER specifies those volumes to be synchronized with the catalogs. This feature can be very useful when testing the DR process in SIMULATE mode before moving to the DR site.

N is the default value. Specify Y to generate the required keyword in the JCL, without generating any arguments; when the JCL is shown for editing or submission, you can add argument) at that time.

MESSAGE-TEXT
The MESSAGE-TEXT field controls the volume of IDCAMS messages printed during the execution of **CATSCRUB**. Enter one of the following values:

- Specify F (the default) if you want a listing of all the matched volumes, a complete process log of all actions taken by the **CATSCRUB** command, indicating the name of every data set record that is deleted from the catalogs processed by **CATSCRUB**, and a summary report (at the end of the listing) that provides a count of each type of catalog record deleted.
- Specify A if you want a summary report only.

Note: The remaining fields are the anomaly keywords. These anomalies control **CATSCRUB** processing of input catalogs when specific conditions are encountered. Each anomaly, with one exception (FATAL-CATALOG-EXCEPTION), provides the following:

- DISPOSITION - Specify what you want done with the catalog entry if it meets the condition: DELETE or KEEP the entry.
- RETURN CODE - Specify the return code with which CATSCRUB terminates should a catalog entry meet the condition.
- ACTION - Specify what you want CATSCRUB to do when an entry meets the condition: CONTINUE or END.

FATAL-CATALOG-ERROR supports an ACTION only.

FATAL-CATALOG-ERR
This anomaly controls **CATSCRUB** processing when a fatal error occurs when processing a catalog. The default ACTION is to END **CATSCRUB** processing. Circumvention is to EXCLUDE-BCS those catalogs causing fatal errors. The following BKMINI token in the CATSCRUB_OPTIONS section will control the initial dialog setting for this anomaly:

Token: FATAL_CATALOG_ERR_PROC

GDGBASE-NO-ACTIVE-GENS
This anomaly controls **CATSCRUB** processing when the GAT (Generation Aging Table) within a GDG record indicates that there are no GDS associated with the GDG.

This can occur either naturally or because **CATSCRUB** emptied the GAT as a result of other **CATSCRUB** processing.

The default DISPOSITION is to DELETE the GDG entry. The default RETURN CODE is 4. The default ACTION is to CONTINUE **CATSCRUB** processing.

The following BKMINI tokens in the CATSCRUB_OPTIONS section will control the initial dialog settings for this anomaly:

Tokens:

GDGBASE_NO_ACTIVE_GENS_DISP

GDGBASE_NO_ACTIVE_GENS_RC

GDGBASE_NO_ACTIVE_GENS_PROC

GDS-MIGRATED
This anomaly controls **CATSCRUB** processing when an individual GDS subrecord indicates that the entry is in migrated status.

This anomaly is incompatible with GDS-MIGRATED-DASD and GDS-MIGRATED-TAPE.

Be careful if DELETE is specified for the DISPOSITION, as this affects the number of entries within the GAT and, potentially, produce an unintended result when a relative generation number is specified in the JCL.

The default DISPOSITION is to KEEP the GDS entry.

The default RETURN CODE is 4.

The default ACTION is to CONTINUE **CATSCRUB** processing.

The following BKMINI tokens in the CATSCRUB_OPTIONS section will control the initial dialog settings for this anomaly:

Tokens:

GDS_MIGRATED_DISP

GDS_MIGRATED_RC

GDS_MIGRATED_PROC

GDS-MIGRATED-DASD

This anomaly controls **CATSCRUB** processing when an individual GDS subrecord indicates that the entry is in migrated status where the device type is disk.

This anomaly is incompatible with GDS-MIGRATED.

If this anomaly is specified, GDS-MIGRATED-TAPE must also be specified.

Be careful if DELETE is specified for the DISPOSITION as this will affect the number of entries within the GAT and, potentially, produce an unintended result when a relative generation number is specified in the JCL.

The default DISPOSITION is to KEEP the GDS entry.

The default RETURN CODE is 4.

The default ACTION is to CONTINUE **CATSCRUB** processing.

GDS-MIGRATED-TAPE

This anomaly controls **CATSCRUB** processing when an individual GDS subrecord indicates that the entry is in migrated status where the device type is tape.

This anomaly is incompatible with GDS-MIGRATED.

If this anomaly is specified, GDS-MIGRATED-DASD must also be specified.

Be careful if DELETE is specified for the DISPOSITION as this will affect the number of entries within the GAT and, potentially, produce an unintended result when a relative generation number is specified in the JCL.

The default DISPOSITION is to KEEP the GDS entry.

The default RETURN CODE is 4.

The default ACTION is to CONTINUE **CATSCRUB** processing.

GDS-MULTI-VOL-ERR

This anomaly controls **CATSCRUB** processing when an individual GDS subrecord indicates that, for an active multivolume generation, one or more of the referenced volumes is not being processed by **CATSCRUB**. Care should be taken if DELETE is specified for the DISPOSITION as this will affect the number of entries within the GAT and, potentially, produce an unintended result when a relative generation number is specified in the JCL.

This keyword will be applied to those entries having a volume cell matching a volser in the MATCH-VOLSER list.

The default DISPOSITION is to KEEP the GDS entry.

The default RETURN CODE is 4.

The default ACTION is to CONTINUE **CATSCRUB** processing.

The following BKMINI tokens in the CATSCRUB_OPTIONS section will control the initial dialog settings for this anomaly:

Tokens:

GDS_MULTI_VOL_ERR_DISP

GDS_MULTI_VOL_ERR_RC

GDS_MULTI_VOL_ERR_PROC

GDS-NOT-FOUND

This anomaly controls **CATSCRUB** processing when an individual GDS subrecord does not physically exist on any of the volumes being processed by the **CATSCRUB** command. Be careful if DELETE is specified for the DISPOSITION as this will affect the number of entries within the GAT and, potentially, produce an unintended result when a relative generation number is specified in the JCL.

The default DISPOSITION is to KEEP the GDS entry.

The default RETURN CODE is 4.

The default ACTION is to CONTINUE **CATSCRUB** processing.

The following BKMINI tokens in the CATSCRUB_OPTIONS section will control the initial dialog settings for this anomaly:

Tokens:

GDS_NOT_FOUND_DISP

GDS_NOT_FOUND_RC

GDS_NOT_FOUND_PROC

GDS-TAPE

This anomaly controls CATSCRUB processing when an individual GDS subrecord is tape resident. Be aware that if DELETE is specified for the DISPOSITION, it affects the number of entries within the GAT and might produce an unintended result when a relative generation number is specified in the JCL.

The default DISPOSITION is to KEEP the GDS entry.

The default RETURN CODE is 4.

The default ACTION is to CONTINUE **CATSCRUB** processing.

The following BKMINI tokens in the CATSCRUB_OPTIONS section will control the initial dialog settings for this anomaly:

Tokens:

GDS_TAPE_DISP

GDS_TAPE_RC

GDS_TAPE_PROC

NONVSAM-ALIAS-NO-REALNAME

This anomaly controls CATSCRUB processing of the alias entry for a non-VSAM object where the true-name is not found within the volumes being processed by the **CATSCRUB** command.

The default DISPOSITION is to KEEP the Alias entry.

The default RETURN CODE is 4.

The default ACTION is to CONTINUE **CATSCRUB** processing.

The following BKMINI tokens in the CATSCRUB_OPTIONS section will control the initial dialog settings for this anomaly:

Tokens:

NONVSAM_ALIAS_NO_REALNAME_DISP

NONVSAM_ALIAS_NO_REALNAME_RC

NONVSAM_ALIAS_NO_REALNAME_PROC

NONVSAM-MIGRATED

This anomaly controls **CATSCRUB** processing when the entry for a non-VSAM data set indicates that the entry is in migrated status. Note that this controls only "A" records (standard non-VSAM objects), not GDS (Generation Data Sets).

This anomaly is incompatible with NONVSAM-MIGRATED-DASD and NONVSAM-MIGRATED-TAPE.

The default DISPOSITION is to KEEP the non-VSAM entry.

The default RETURN CODE is 4.

The default ACTION is to CONTINUE **CATSCRUB** processing.

The following BKMINI tokens in the CATSCRUB_OPTIONS section will control the initial dialog settings for this anomaly:

Tokens:

NONVSAM_MIGRATED_DISP

NONVSAM_MIGRATED_RC

NONVSAM_MIGRATED_PROC K, 4, C

NONVSAM-MIGRATED-DASD

This anomaly controls **CATSCRUB** processing when the entry for a non-VSAM data set indicates that the entry is in migrated status where the device type is disk. Note that this controls only "A" records (standard non-VSAM objects), not GDS (Generation Data Sets).

This anomaly is incompatible with NONVSAM-MIGRATED. If this anomaly is specified, NONVSAM-MIGRATED-TAPE must also be specified.

The default DISPOSITION is to KEEP the non-VSAM entry.

The default RETURN CODE is 4.

The default ACTION is to CONTINUE **CATSCRUB** processing.

NONVSAM-MIGRATED-TAPE

This anomaly controls **CATSCRUB** processing when the entry for a non-VSAM data set indicates that the entry is in migrated status where the device type is tape. Note that this controls only "A" records (standard non-VSAM objects), not GDS (Generation Data Sets).

This anomaly is incompatible with NONVSAM-MIGRATED. If this anomaly is specified, NONVSAM-MIGRATED-DASD must also be specified.

The default DISPOSITION is to KEEP the non-VSAM entry.

The default RETURN CODE is 4.

The default ACTION is to CONTINUE **CATSCRUB** processing.

NONVSAM-MULTI-VOL-ERR

This anomaly controls **CATSCRUB** processing when the entry for a non-VSAM data set indicates that, for an active multivolume generation, one or more of the referenced volumes is not being processed by **CATSCRUB**.

Note that this controls only "A" records (standard non-VSAM objects), not GDS (Generation Data Sets).

This keyword will be applied to those entries having a volume cell matching a volser in the MATCH-VOLSER list.

The default DISPOSITION is to KEEP the non-VSAM entry.

The default RETURN CODE is 4.

The default ACTION is to CONTINUE **CATSCRUB** processing.

The following BKMINI tokens in the CATSCRUB_OPTIONS section will control the initial dialog settings for this anomaly:

Tokens:

NONVSAM_MULTI_VOL_ERR_DISP

NONVSAM_MULTI_VOL_ERR_RC

NONVSAM_MULTI_VOL_ERR_PROC

NONVSAM-NOT-FOUND

This anomaly controls **CATSCRUB** processing when a non-VSAM data set does not physically exist on any of the volumes being processed by the **CATSCRUB** command, and the NONVSAM-MIGRATE, NONVSAM-MULTI-VOL-ERR and NONVSAM-TAPE conditions do not apply.

Note that this controls only "A" records (standard non-VSAM objects), not GDS (Generation Data Sets).

The default DISPOSITION is to KEEP the non-VSAM entry.

The default RETURN CODE is 4.

The default ACTION is to CONTINUE **CATSCRUB** processing.

The following BKMINI tokens in the CATSCRUB_OPTIONS section will control the initial dialog settings for this anomaly:

Tokens:

NONVSAM_NOT_FOUND_DISP

NONVSAM_NOT_FOUND_RC

NONVSAM_NOT_FOUND_PROC

NONVSAM-TAPE

This anomaly controls **CATSCRUB** processing when a non-VSAM data set is determined to be tape resident. Note that this controls only "A" records (standard non-VSAM objects), not GDS (Generation Data Sets).

The default DISPOSITION is to KEEP the non-VSAM entry.

The default RETURN CODE is 4.

The default ACTION is to CONTINUE **CATSCRUB** processing.

The following BKMINI tokens in the CATSCRUB_OPTIONS section will control the initial dialog settings for this anomaly:

Tokens:

NONVSAM_TAPE_DISP

NONVSAM_TAPE_RC

NONVSAM_TAPE_PROC

OBJECT-ACCESS-METHOD

This anomaly controls **CATSCRUB** processing when an entry using OAM is met.

The default DISPOSITION is to KEEP the OAM-using entry.

The default RETURN CODE is 4.

The default ACTION is to CONTINUE **CATSCRUB** processing.

The following BKMINI tokens in the CATSCRUB_OPTIONS section will control the initial dialog settings for this anomaly:

Tokens:

OBJECT_ACCESS_METHOD_DISP

OBJECT_ACCESS_METHOD_RC

OBJECT_ACCESS_METHOD_PROC

VSAM-SPHERE-MIGRATED

This anomaly controls **CATSCRUB** processing when a VSAM cluster sphere record indicates that the entry is in migrated status.

This anomaly is incompatible with VSAMSPHERE-MIGRATED-DASD and VSAMSPHERE-MIGRATED-TAPE.

The default DISPOSITION is to KEEP the cluster entry.

The default RETURN CODE is 4.

The default ACTION is to CONTINUE **CATSCRUB** processing.

The following BKMINI tokens in the CATSCRUB_OPTIONS section will control the initial dialog settings for this anomaly:

Tokens:

VSAMSPHERE_MIGRATED_DISP

VSAMSPHERE_MIGRATED_RC

VSAMSPHERE_MIGRATED_PROC

VSAM-SPHERE-MIGRATED-DASD

This anomaly controls **CATSCRUB** processing when a VSAM cluster sphere record indicates that the entry is in migrated status where the device type is disk.

This anomaly is incompatible with VSAMSPHERE-MIGRATED.

If this anomaly is specified, VSAMSPHERE-MIGRATED-TAPE must also be specified.

The default DISPOSITION is to KEEP the VSAM sphere entry.

The default RETURN CODE is 4.

The default ACTION is to CONTINUE **CATSCRUB** processing.

VSAM-SPHERE-MIGRATED-TAPE

This anomaly controls **CATSCRUB** processing when a VSAM cluster sphere record indicates that the entry is in migrated status where the device type is tape.

This anomaly is incompatible with VSAMSPHERE-MIGRATED.

If this anomaly is specified, VSAMSPHERE-MIGRATED-DASD must also be specified.

The default DISPOSITION is to KEEP the VSAM sphere entry.

The default RETURN CODE is 4.

The default ACTION is to CONTINUE **CATSCRUB** processing.

VSAM-SPHERE-MULTI-VOL-ERR

This anomaly controls **CATSCRUB** processing when a VSAM cluster sphere record indicates that, for an active multivolume object, one or more of the referenced volumes is not being processed by **CATSCRUB**.

This keyword is applied to those entries having a volume cell matching a volser in the MATCH-VOLSER list.

The default DISPOSITION is to KEEP the VSAM sphere entry.

The default RETURN CODE is 4.

The default ACTION is to CONTINUE **CATSCRUB** processing.

The following BKMINI tokens in the CATSCRUB_OPTIONS section will control the initial dialog settings for this anomaly:

Tokens:

VSAMSPHERE_MULTI_VOL_ERR_DISP

VSAMSPHERE_MULTI_VOL_ERR_RC

VSAMSPHERE_MULTI_VOL_ERR_PROC

VSAM-SPHERE-NOT-FOUND

This anomaly controls **CATSCRUB** processing when a VSAM cluster does not physically exist on any of the volumes being processed by the **CATSCRUB** command.

The default DISPOSITION is to KEEP the VSAM sphere entry.

The default RETURN CODE is 4.

The default ACTION is to CONTINUE **CATSCRUB** processing.

The following BKMINI tokens in the CATSCRUB_OPTIONS section will control the initial dialog settings for this anomaly:

Tokens:

VSAMSPHERE_NOT_FOUND_DISP

VSAMSPHERE_NOT_FOUND_RC

VSAMSPHERE_NOT_FOUND_PROC

When you specify S for Selected List on the **Submit CATSCRUB** panel, the software displays a list of catalogs from which you can choose the values for the BCS keyword. The following figure shows an example of such a list:

```
                                          Row 22 to 30 of 30 |
| Command ===>                             Scroll ===> PAGE  |
|                                                            |
|   Master Catalog:                                          |
| Primary Commands: S(elect) mask, C(lear) mask              |
|                                                            |
| C Enter "S" to select Name(s)                              |
| ---------------------------------------------------------- |
|   USERCAT.SCD                                              |
|   USERCAT.UCAT1                                            |
|   USERCAT.UCAT2                                            |
|   USERCAT.UCAT3                                            |
|   USERCAT.VCR01                                            |
|   USERCAT.VCR02                                            |
|   USERCAT.VCR03                                            |
|   USERCAT.VCR04                                            |
|   USERCAT.VIPLD91                                          |
| ********************* Bottom of data ********************** |
```

Chapter 21. Searching for data sets

Use the Search for Data Set feature to locate backup copies for, or references by applications to, specific data sets.

You can search on an explicit data set name or a mask. You can also apply further search criteria to restrict the search results to only those data sets whose attributes match your criteria.

Advanced Backup and Recovery for z/OS supports three types of data set search:

- A full search of the IDS for data sets/Unix files displays on the **Search for Dataset** panel a list of all generations of every data set/Unix file that matches the search criteria. If your site uses CA Disk to perform backups, you may specify that the search include both the IDS and the CA Disk backup inventory for data sets that match your search criteria. By applying line commands to backup data sets in the search results list, you can perform these tasks:
  - Display additional information about the data set/Unix file
  - Show overlaps in backup coverage
  - Recover a data set
  - Show information about the event that backed up the data set/Unix file
- A quick search of the IDS for data sets displays on the Quick Search for Dataset panel only generation (0) for each data set that matches the search criteria. By applying line commands to backup data sets in the search results list, you can perform these tasks:
  - Display additional information about the data set
  - (Any data set) Show all generations of the selected data set
  - Show overlaps in backup coverage
  - Recover a data set
  - Show information about the event that backed up the data set
- A search of the Historical Dataset Usage database displays on the Historical Dataset Record panel all references by applications to those data sets whose attributes match the search criteria. By applying line commands to the data sets in the search results list, you can perform these tasks:
  - Display SMF data for a selected data set
  - Show evaluation results for the selected data set
  - Show information about the most recent backup for a selected data set
  - Show information about all backups for the selected data set
  - Show information about the event that backed up the data set

To access the data set search facility, choose the **Search for Dataset** option from the **Main Menu** panel.

Searching for all generations of a data set

The data set search facility **Search for Datasets** option returns a list of all generations of the data sets whose attributes meet your search criteria.

1. From the **Search Selection** panel, select the **Search for Datasets** option.
2. Required: In the **DSN or Mask** field on the **Search for Dataset Setup** panel, specify a data set name (a maximum of 44 characters). You can specify a data set name mask.

   Note: If you specify * (search all data sets), a search warning message indicates that the search may take a considerable amount of time. Type Y and press Enter to continue the search or type N and press Enter to return to the **Search for Dataset Setup** panel.
3. In the **Auto Expand List** field specify one of these options:

Option	Description
Y	Display previous versions of backups of each data set.
N	Do not display previous versions of backups for each data set.

4. In the **Sort by** field specify one of these options:

Option	Description
N	Sort the list by data set name or type.
D	Sort the list by date.

5. In the **All DS Backups** field, specify one of these options:

Option	Description
Y	Display data sets, including individually backed up data sets. The list includes those data sets backed up using FDR/DSF.
N	Do not display data sets that were backed up individually.

6. In the **Overlaps Only** field specify one of these options:

Option	Description
Y	Display only those data sets that are backed up by more than one aggregate.
N	Display all data sets, including those data sets that are backed up by more than one aggregate.

7. In the **HSM Incrementals** field specify one of these options:

Option	Description
Y	Display HSM incremental or command backup data sets matching the data set name or mask specified
N	Do not display HSM incremental or command backup data sets matching the data set name or mask specified.

   Note: This option is valid for ABARS Manager and Critical Backup Tracking and Inventory. For Critical Backup Tracking and Inventory, DFSMShsm incremental backups do not have any generations associated with them in displays or reports because Critical Backup Tracking and Inventory tracks backups by job name. DFSMShsm incremental backups are displayed in date or time order and can be viewed in relationship to the rest of the data set backups and respectively restored.

   Tip: When specifying search criteria, be as specific as possible. The **Search for Data Set** panel calculates each generation number for each data set, which can be a time-consuming process. To avoid extremely long search processing times, only the first 50,000 data sets are displayed. Use the **Quick Search** option if you need to view more than 50,000 data sets.

8. (Critical Backup Tracking and Inventory only) In the **Full Volume Dumps** field specify whether to display full volume dump data sets matching the specified data set name or mask. With Critical Backup Tracking and Inventory, by default, data sets in full volume dumps are not displayed. Generation numbers are not displayed because full volume dump data sets are not generally restored at the data set level. To prevent inadvertent restores, the numbers are not included in the display.
   - If full volume dumps are tracked by Critical Backup Tracking and Inventory, specify Y to display full volume dump data sets.
   - If you do not want to display full volume dump data sets, specify N.
9. In the **Date Format** field, specify one of these options:

Option	Description
Y	Display dates in yy/mm/dd format.

Option	Description
N	Display dates in dd/mm/yy format.

10. In the **Store values to be used as Default for this USERID** field type a forward slash (/) to store the values specified on this panel. If selected, the next time the **Search for Dataset** panel is displayed for your user ID, the fields will contain the most recent values you specified. To view or modify your selections after they have been stored, type an S or a question mark in the **C** field on the **Search for Data Set** panel.
11. In the **Don't show this panel again** field type a forward slash (/) to display the **Search for Data Set** panel when you select Option 5 from the main menu.
12. Press Enter to display the **Search for Data Set** panel.

Search for Data Set field and command reference

The **Search for Dataset** panel displays all data sets/Unix files found in the that match the search criteria you specified on the **Search for Dataset Setup** panel.

Fields

Sort by
Specify the sort order:

- D — Sort the list by date.
- N — Sort the list by data set name or type.

Display All DS Bkups
Specify one of these values:

- Y — Display data sets, including data sets backed up individually using DFdss/LD and FDR/DSF.
- N — Do not display those data sets that were backed up individually.

Display overlaps only
Specify one of these values:

- Y — Display those data sets that are backed up by more than one aggregate.
- N — Do not display only those data sets that are backed up by more than one aggregate.

Display HSM Incr
Specify one of these values:

- Y — Display HSM incremental or command backup data sets matching the data set name or mask specified.
- N — Do not display HSM incremental or command backup data sets matching the data set name or mask specified.

Note: This option is valid for ABARS Manager and Critical Backup Tracking and Inventory. For Critical Backup Tracking and Inventory, DFSMShsm incremental backups do not have any generations associated with them in any displays or reports because Critical Backup Tracking and Inventory tracks backups by job name. DFSMShsm incremental backups will be displayed in date or time order and can be viewed in relationship to the rest of the data set backups and respectively restored.

Display Full Vol Dumps
(Critical Backup Tracking and Inventory only) In the **Full Volume Dumps** field specify whether or not to display full volume dump data sets matching the specified data set name or mask. With Critical Backup Tracking and Inventory, by default, data sets in full volume dumps are not displayed. It can appear that generation numbers have been skipped when displaying data sets in the **Search for Data Set** panel but in fact, they are accounted for. Generation numbers are not displayed because full volume dump data sets are not generally restored at the data set level. The numbers are not displayed in order to prevent inadvertent restores.

- Y — If full volume dumps are tracked by Critical Backup Tracking and Inventory, specify Y to display full volume dump data sets.
- N — Do not display full volume dump data sets.

Date Format
Specify one of these values:

- Y — Display dates in yy/mm/dd format.
- D — Display dates in dd/mm/yy format.

When you make entries in the fields on the upper portion of the panel and press Enter, those entries determine the values shown in the following fields on the lower portion of the panel:

C
Command input. The field next to an entry in the list enables you to specify a line command for that particular entry.

Dataset Name or Mask
Specify a data set name/Unix file index (a maximum of 44 characters). You can specify a data set name mask.

+/-
Display or hide previous backup versions:

- + — Display previous versions of the backup for the selected data set.
- - — Display only the latest version of the backup for the selected data set.

Note: ADRDSSU JCL for Unix files backup accepts full Unix File Path in PATH or WORKINGDIRECTORY parameters which can be up to 255 characters long for both. So, for searching Unix files/directories backed up through ADRDSSU utility, user needs to provide Unix File Index which is formatted in 44 characters as shown below:

```
<first 10 characters of Directory Name>~<Last 10 Characters of Directory
Name>|<first 10 characters of Unix File Name>~<Last 10 Characters of Unix
File Name>
```

The following is an example of Search for Dataset results display for Unix files present in sub-directories of the /u/ts3435 directory.

```
BKM5510                BKM   Search for Dataset           Row 41 to 50 of 345

Primary Commands: REFresh, RESet, R mask, X mask          Sort by => N  D,N
   Line Commands: +,-,I,O,R,S              Display DS LVL Bkups   => Y  Y or N
                                           Display overlaps only  => N  Y or N ou
                                           Display HSM Incr       => N  Y or N
                                           Display VOL LVL Dumps  => N  Y or N
                                           Date Format            => Y  Y or D
C Dataset Name or Mask                        +- Gen  PGM/Type Date     Time  O
  /u/ts3435*                                   -               yy/mm/dd
-------------------------------------------------------------------------------
  /u/ts3435/~_for_test|/uuld~55|001            +    0 DSS ZFS  23/02/23 02:50 *
  /u/ts3435/~_for_test|u-ts3435-~aracters|001  +    0 DSS ZFS  23/03/02 04:41 *
  /u/ts3435/~_for_test|uld~|001                +    0 DSS ZFS  23/03/01 01:44
  /u/ts3435/~aaaaaaaaa|/BKMFILE|001            +    0 DSS ZFS  22/12/15 06:32
  /u/ts3435/~characte|-name-sho~racters|001    +    0 DSS ZFS  23/03/29 23:20 *
  /u/ts3435/~characte|/a /b|001                +    0 DSS ZFS  23/04/06 02:58 *
  /u/ts3435/~characte|/a|001                   +    0 DSS ZFS  23/04/06 03:29
  /u/ts3435/~characte|/a~/b|001                     0 DSS ZFS  23/04/10 03:12
  /u/ts3435/~characte|/b|001                   +    0 DSS ZFS  23/04/10 00:44 *
  /u/ts3435/~characte|/u-ts3435~acters|001           0 DSS ZFS  23/04/06 05:26
```

Gen
(Display only) The relative generation of the backup.

PGM/Type
(Display only) The program or utility that performed the backup:

- ADR DLTA— ADARUN DELTA

- ADR FULL — ADARUN FULL
- LOG COPY—ADARUN PLCOPY
- AB VER— ABARS ABACKUP Verify
- AB EXEC—ABARS ABACKUP Execute
- AR VER— ABARS ARECOVER Verify
- AR EXEC— ABARS ARECOVER Execute
- AR PREP— ABARS ARECOVER Prepare
- BCS CR+— Catalog RecoveryPlus BCS Backup
- DSN CR+— Catalog RecoveryPlus DSN Backup
- VVDS CR+— Catalog RecoveryPlus VVDS Backup
- BCS ACM—Advanced Catalog Management BCS Backup
- DSN ACM—Advanced Catalog Management DSN Backup
- VVDS ACM—Advanced Catalog Management VVDS Backup
- CA-FAVER— CA—FAVER
- CADISK— CA-Disk
- CFC REP— CFCAMS REPRO
- CFC EXP— CFCAMS EXPORT
- COPYDUMP—DSS COPYDUMP
- DSS LD— DSS LOGICAL DUMP
- DSS PCDS— DSS PHYSICAL DUMP
- DSS PDDS—DSS FULL VOLUME COPY
- DSS FVD— DSS FULL VOLUME DUMP
- DSS ZFS— DSS Unix Files Backup
- FDRA ABR— FDR ABR ABR DUMP
- FDRA APPL— FDR APPL DUMP
- FDR DSF— FDR DSF DUMP
- FDRA DSF— FDR ABR DSF DUMP
- FDRA FDR—FDR ABR FDR DUMP
- FDRCOPY—FDR COPY
- FDR FVD — FDR FULL VOLUME DUMP
- FREDSNB— FAST REPLICATION DATA SET BACKUP
- HSM ARCB—HSM ARCINBAK
- HSMAUTO—HSM AUTODUMP
- HSM INCR— HSM INCREMENTAL BACKUP
- IDC EXP — IDCAMS EXPORT
- IDC REP— IDCAMS REPRO
- ICEGENER— ICEGENER
- ICEMAN— ICEMAN
- ICETOOL— ICETOOL
- IEBGENER — IEBGENER
- IEBCOPY — IEBCOPY
- SORT — SORT
- SYNCGENR— SYNCGENR

- SYNCSORT— SYNCSORT
- TAPECOPY—TAPECOPY
- USERDEF—USER DEFINED
- VSAMASST—VSAM Assist dump

Date
(Display only) The date of the backup.

Time
(Display only) The time of the backup.

O
An asterisk in this field indicates overlaps exist for the data set.

Primary commands

Specify primary commands at the **Command** prompt at the top of the panel.

REFresh
Refresh (update) the display.

RESet
Clear all selected data sets.

R mask
Specify a dsn or mask to select data sets for recovery. For example, specify ABC.** to select all data sets with a high-level qualifier of "ABC".

X mask
Clear any data sets selected for recovery. For example, specify XYZ.** to clear the selection of all data sets a high-level qualifier of " XYZ".

Line commands

Specify line commands in the **C** field next to an entry in the list.

+
Display previous versions of the backup for the selected event.

-
Display the latest version of the backup for the selected event.

I
Display data set or Unix file/directory detail information (backups only).

O
Displays the **Overlap Search by Dataset** panel which shows overlaps for the selected data set. The overlap options cross references the data set to determine if another job is backing up the selected data set (backups only).

R
Select a data set/Unix file for recovery.

S
Display backup summary detail (backups only).

Searching for gen(0) of a data set

The data set search facility **Quick Search** option returns a list of the most recent generation of the data set backups whose attributes meet your search criteria.

1. From the **Search Selection** panel, select the **Quick Search** option.
2. Required: In the **DSN or Mask** field on the **Quick Search for Dataset Setup** panel, specify a data set name or mask (a maximum of 44 characters).
3. In the **All DS Backups** field, specify one of these options:

Option	Description
Y	Display data sets, including data sets that were backed up individually. The data set list includes data sets backed up using FDR/DSF.
N	Do not display data sets that were backed up individually.

4. In the **HSM Incrementals** field, specify whether to display HSM incremental or command backup data sets matching the data set name or mask specified.

   Note: This option is valid for ABARS and Critical Backup Tracking and Inventory. For Critical Backup Tracking and Inventory, DFSMShsm incremental backups do not have any generations associated with them in any displays or reports because Critical Backup Tracking and Inventory tracks backups by job name. DFSMShsm incremental backups are displayed in date or time order and can be viewed in relationship to the rest of the data set backups and respectively restored.
5. In the **Full Volume Dumps** field specify whether or not to display full volume dump data sets matching the specified data set name or mask. With Critical Backup Tracking and Inventory, by default, data sets in full volume dumps are not displayed. Generation numbers are not displayed because full volume dump data sets are not generally restored at the data set level. The numbers are not displayed in order to prevent inadvertent restores.
6. In the **Date Format** field, specify one of these options:

Option	Description
Y	Display dates in yy/mm/dd format
N	Display dates in dd/mm/yy format

7. In the **Store values to be used as Default for this USERID** field, type a forward slash (/) to store the values specified on this panel. If selected, the next time the **Search for Dataset** panel is displayed for your user ID, the fields will contain the most recent values you specified. To view or modify your selections after they have been stored, type S or a question mark in the **C** field on the **Quick Search for Data Set** panel.
8. In the **Don't show this panel again** field type a forward slash (/) to display the **Quick Search for Data Set** panel when you select Option 5 from the main menu.
9. Press Enter to display the **Quick Search for Dataset** panel.

Quick Search for Dataset field and command reference

The **Quick Search for Dataset** panel displays all data sets found in the Advanced Backup and Recovery for z/OS Inventory Data Set that match the search criteria you specified on the **Quick Search for Dataset Setup** panel.

Fields

Display All DS Bkups
Specify one of these values:

- Y — Display data sets, including data sets that were backed up individually.
- N — Do not display data sets that were backed up individually.

Display HSM Incr
Specify one of these values:

- Y — Display HSM incremental or command backup data sets matching the data set name or mask specified.
- N — Do not display HSM incremental or command backup data sets matching the data set name or mask specified.

Note: This option is valid for ABARS Manager and Critical Backup Tracking and Inventory. For Critical Backup Tracking and Inventory, DFSMShsm incremental backups do not have any generations associated with them in any displays or reports because Critical Backup Tracking and Inventory tracks

backups by job name. DFSMShsm incremental backups will be displayed in date or time order and can be viewed in relationship to the rest of the data set backups and respectively restored.

Display Full Vol Dumps
(Critical Backup Tracking and Inventory only) In the **Full Volume Dumps** field specify whether or not to display full volume dump data sets matching the specified data set name or mask. With Critical Backup Tracking and Inventory, by default, data sets in full volume dumps are not displayed. It can appear that generation numbers have been skipped when displaying data sets in the **Quick Search for Data Set** panel but in fact, they are accounted for. Generation numbers are not displayed because full volume dump data sets are not generally restored at the data set level. The numbers are not displayed in order to prevent inadvertent restores.

- Y — If full volume dumps are tracked by Critical Backup Tracking and Inventory, specify Y to display full volume dump data sets.
- N — Do not display full volume dump data sets.

Date Format
Specify the appropriate option:

- Y — Display dates in yy/mm/dd format.
- D — Display dates in dd/mm/yy format.

C
Command input. The field next to an entry in the list enables you to specify a line command for that particular entry.

Dataset Name or Mask
Specify a data set name (a maximum of 44 characters). You can specify a data set name mask.

Jobname
(Display only) The name of the job.

PGM/Type
(Display only) The program or utility that performed the backup (for example ABACKUP).

Date
(Display only) The date of the backup.

Time
(Display only) The time of the backup.

Primary commands

Specify primary commands at the **Command** prompt at the top of the panel.

REFresh
Refresh (update) the display.

RESet
Clear all selected data sets.

Line commands

Specify line commands in the **C** field next to an entry in the list.

I
Display data set detail information (backups only).

L
Display previous versions of the backup for the selected event.

O
Show overlaps. Cross reference the data set to determine if another job is backing up the selected data set (backups only).

R
Select a data set for recovery.

S
Display backup summary detail (backups only).

Searching for historical dataset usage

The ACDI Historical Dataset Usage feature provides a simple method for determining all places where a specific data set has been referenced during the time period you specify.

You can use filtering criteria to restrict the search results to only those data sets whose attributes match your criteria. For each reference that is found for the specified data set, the referencing data set's name, application ID, SMF record type, and the time stamp of the reference are displayed.

1. From the **Search Selection** panel, choose the **ACDI Historical Dataset Usage** option.
2. On the **Quick Search for Historical Dataset Record** panel, specify the filtering criteria that you want applied to the data set search; only those data sets whose attributes meet these criteria are displayed in the list on the **Historical Dataset Record** panel. Specify the following information:
   a) In the **DSN or mask** field, specify a name or a mask for the search data set (a maximum of 44 characters).
   b) In the **Application or mask** field, specify the name (or a mask for the application name) that references the search data set.
   c) In the **Jobname or mask** field, specify the name (or a mask for the jobname) that references the search data set.
   d) If you want to search for usage records within the most recent *n* days, specify that number of days in the **Days in History** field.
   e) If you want to search for usage records within a specific date range, specify those dates in the **Start Date** and **End Date** fields. Use the date format YYYY/MM/DD.
   f) Press Enter to run the specified search and display the results on the **Dataset History Record** panel.
3. To use one of the available line commands, type the appropriate letter in the **C** column immediately to the left of the data set name and press Enter.
   - Type S to show the SMF data for the selected data set. The following figure shows an example of the type of SMF data:

```
                HISTORICAL DATASET USAGE DETAIL RECORD
Command ==>                                              SCROLL==> PAGE


DATASET: RR02.$DEMO.ABRV23.CNTL
         APPLRR      Appl Name
         0000000033  Cycle
         APPLRRJ3    Jobname collected from
         RL          DDname collected from
         ********    Program name collected from
         OLD         Disposition
         INPUT       Access
         TST         System ID collected from
         14          SMF record type
         06/04/2013  Date collected
         15.46.57    Time collected
         DASD        Device
         TSTSM8      Volser (1st if multi-volume)
         PS          DSORG
         00000392    Dataset size in Kbytes
```

   - Type E to show evaluation results for the selected data set.

```
                                DSR DATA
Command ==>                                          APPLICATION ==> APPLRR

 DATASET NAME: RR02.ABR.APPLJ3.DC.SORTED.G0008V00
 STATUS: INCLUDE     REASON: EVALUATION
 FILTER HIT: INCLUDE    FILTER SCOPE: LOCAL      FILTER TYPE:
 FILTER:
 GDS: NEW GDG = INCL/ALLOC
                                                                More:     +
 SMF DATA   RECORD TYPE: 61
 SYSID   CYCLE NUMBER DATE      TIME
 TST     0000000034   2013.156  13.30.39.37
 JOBNAME   DDNAME    PROGRAM   ACCESS    DISP  UNIT  VOLSER  ORG  SIZE /K
 APPLRRJ3  ********  ********  DEFINE    NEW   DASD  TSO002  PS   00000000
 JOBSTEP   PROCSTEP  JCLGEN   SMFSRCE
 ********  ********  ****      RSP
 JCL DATA
 SYSID   CYCLE NUMBER RDRDATE   RDRTIME
 TST     0000000034   2013.156  13.30.39.29
                              SCHD  NORM  ABNM
 JOBNAME   DDNAME    PROGRAM   DISP  DISP  DISP JOBID
 APPLRRJ3  SORTOUT   SORT      NEW   CAT   CAT  J0043618
 JOBSTEP   PROCSTEP  JCLGEN    INDIRECT REFERENCE
 STEP1               +1        N
```

- Type L to show information about the most recent backup for the selected data set. When the Quick Search for Dataset panel appears, you can use any of the following line commands for a selected backup:

I	View the **Dataset Detail** panel, which shows management class, storage group and class, data class, allocation, and other information.
L	View information about the most recent backup for the selected data set.
O	Show other events that are also backing up the selected data set.
R	Restore the selected data set.
S	Show detailed information about the event that backed up the selected data set.

- Type A to show all backups for the selected data set. When the **Search for Dataset** panel appears, you can use any of the following line commands for a selected backup:

+	Expand the display to show earlier versions of the backup for the selected data set.
-	Collapse the display to show only the most recent version of the backup for the selected data set.
I	View the **Dataset Detail** panel, which shows management class, storage group and class, data class, allocation, and other information.
O	Show other events that are also backing up the selected data set.
R	Restore the selected data set.
S	Show detailed information about the event that backed up the selected data set.

Working with the search results

Directly from the data set search results list, you can use line commands to display additional information about a data set and to recover data sets from backups.

For full searches and quick data set searches, the **Search for Dataset** or **Quick Search for Dataset** panel displays a list of all data sets that match the search criteria you specified. From that list, you can perform the following tasks on individual data sets in the list:

- For any data set in the list, you can display additional detail about the data set.
- For backup data sets, you can perform these tasks:
  - Display a list of other aggregates or events that backed up the selected data set.

- Recover data sets that were backed up by an HSM incremental backup.
- Restore data sets that were backed up by an unknown utility. This functionality is not available from the **Quick Search for Dataset** panel

For historical data set usage searches, the **Dataset History Record** panel displays a list of application data sets that match the search criteria you specified. From that list, you can perform the following tasks on the data sets in the list:

- Display SMF data for the data set.
- Show the evaluation results for the data set.
- Display information about the most recent backup for the selected data set. Further, for that most recent backup, you can perform these tasks:
  - Display detailed information about the selected backup data set.
  - Display a list of other aggregates or events that backed up the selected data set.
  - Restore the application data set using the selected backup.
- Display information about all backups for the selected data set. Further, for any of those backups, you can perform these tasks:
  - Display detailed information about the selected backup data set.
  - Display a list of other aggregates or events that backed up the selected data set.
  - Restore the application data set using the selected backup.

Displaying detailed information for a search result data set

Directly from the data set search results list, you can use the **I** line command to display detailed information about any data set/Unix file in the list.

1. From the list of data sets/files shown on the **Search for Dataset** or **Quick Search for Dataset** panel, type `I` in the **C** column adjacent to the name of the data set/Unix file of interest, and then press Enter.
2. When you have finished viewing the information on the **View Dataset Detail** panel, press the END function key to display the **Search for Dataset** or **Quick Search for Dataset** panel.

   Detailed View Panel for Dataset

```
BKM5223                CBTI   View Dataset Detail

Jobname         => BKMAPP12 (0)        Date/Time     => 2022/04/20 04:37:07
Stepname/Task   => STEP001  / 1        Type          => IEBGENER
Procname        => N/A
RC ( GNR /AST)) => 0 / 0

Data Set Name     => BKM.BKM0604.GA.NEWAPP.SELECT
                                                                More:     +

GENERAL DATA                    ALLOCATION             Show Space as K
 Management Class => MCAGGT                            TB  GB  MB  KB
 Storage Group    => TSODATA     Allocated          =>             55.3
 Storage Class    => TSODATA     Used               =>             55.3
  Volume serial   => TSO002      Allocated Unit     => TRKS
  Device type     => DISK        Primary            => 0001
 Data Class       => ........    Secondary          => 0015
  Organization    => PS
  Record format   => FB
  Record length   => 00080
  Block size      => 27920
  Original Volume =>             (If migrated)

  Creation date   => 2022109    Referenced date     => 2022110

RECOVER DATA
 ABARS Category   =>
 Device Name      => 3390

VOLUME LIST (Up To The First 50)
 Volume Count     => 00001
 ===>  TSO002
```

Detailed View Panel for Unix file

```
BKM522Z            CBTI   View Unix Path and File name

Jobname         => DIRSPCZ0 (0)        Date/Time     => 2023/05/24 20:52:45
Stepname/Task   => LOGIC1   / 1        Type          => ADRDSSU  (ZFS     )
Procname        => N/A
RC ( DSZ /CBTI)) => 0 / 0               zFS files

File Index        => /u/ts3435/~5_charac|/u-ts3435~acters|001

Unix Dir Name  => '/u/ts3435/zfstest/test1/this_name_is_more_than_255_charac +
Unix File Name => '/u-ts3435-zfstest-test1-this-name-is-more-than-255-charac +

   There is no more info for zFS file
```

Note: Here the users can use the **ZEXPAND** command to display the complete Unix File/Unix Directory Name by typing **ZEXPAND** in the command bar, place cursor on any of these two and press ENTER.

Displaying data set overlaps

From the data set search results, use the line command O to search the Advanced Backup and Recovery for z/OS Inventory Data Set for any other aggregates or events that have backed up the selected data set.

- On the **Search for Dataset** or **Quick Search** panel, an asterisk (*) in the **O** field indicates that overlaps exist for a data set. To identify the data set overlaps for a particular data set, type + in the **C** field next to the appropriate data set and press Enter.

  Note: If the **HSM Incrementals** option was specified as Y on the **Search for Data Set Setup** panel, HSM incremental and command backups are also displayed. These HSM backups can be selected for HRECOVER without knowledge of the HSM HRECOVER syntax. However, you must ensure the data set does not exist on the system (unless you are using the REPLACE command), or is not migrated. HSM backups are not considered to be overlap data sets.

  The **Overlap Search by Dataset** displays the data set overlaps found for the selected data set (if any).

Overlap Search by Dataset field and command reference

This information describes the fields and commands available from the **Overlap Search by Dataset** panel.

Fields

Name
The program or utility that performed the backup:

- CA1
- DS7
- IVPACC2
- TLMS
- DSM

Dataset Name
The name of the data set that has overlaps.

DSN
The name of the data set overlaps.

Name
The program or utility (tape management system) that performed the backup:

- CA1
- DS7
- IVPACC2
- TLMS
- DSM

Date
The date of the backup.

Time
The time of the backup.

Recovering a data set backed up by an HSM incremental using the HRECOVER command

Directly from the data set search results list, you can recover any data set that has an HSM incremental backup.

1. Type R next to the data set that has HSM as the backup name

   An ISPF panel is displayed, from which you can select **HRECOVER** parameters. This panel is displayed when you use the line command **R** on the **Search for Dataset**, **Quick Search for Dataset**, or **Historical Dataset Usage** options.
2. Specify the **HRECOVER** options:
   a) In the **Specify Newname** field, specify the data set you want **HRECOVER** use for the restore.
   b) In the **Replace data set if already cataloged** field, specify one of these options:
      - To issue the **REPLACE** command on **HRECOVER**, specify Y.
      - If you do not want to replace the existing data set during recovery, specify N.
3. The System Administrator feature gives authorized users the ability to issue the **HSEND** command and the ability to specify additional System Administrator keywords. By specifying Y, the **HSEND** command is issued instead of **HRECOVER**. If you do not have System Administrator access DFSMShsm fails the command.
   a) In the **Force to NON-SMS** field, specify one of these options:

- To use the HSM keyword **FORCENONSMS** to force the data set to a non-SMS volume, specify Y.
- If you do not want to use the **FORCENONSMS** keyword to force the data set to a non-SMS volume, specify N.

b) In the **ToVolume** field, specify the volume on which the data set should reside:

c) In the **Unit** field, specify the unit to use.

4. Press Enter to recover the data set using **HRECOVER**. (To replace the data set if it already exists, set **Replace data set if already cataloged** to Y and press Enter.)

   The message `*PRESS END(PF3) TO RESTORE SELECTED DSNS` is displayed.
5. To cancel this request so the **HRECOVER** is not submitted, type RESET on the command line or `CANCEL`. Press PF3 to recover the selected data set and view the HSM **HRECOVER** messages.

Restoring data backed up by an unknown utility

From the results returned from a full data set search, you can use the **R MASK** and **X Mask** primary commands to restore a group of data sets without having to know which utilities backed them up. For example, if you have 1000 IDMS databases you want to restore but you do not want to restore a specific aggregate, you can review the search results list and select the IDMS databases you want to restore.

The **X Mask** command is useful when you want to filter a category of data sets. For example, if you specify `R IDMS.**` and then specify `X IDMS.O*.**`, the command excludes those data sets whose names match the mask. Only those data sets with an R in the **C** column next to their names will be recovered.

1. In the **R Mask** field, specify the mask for the data sets you want to restore. the data sets) and **X Mask** (exclude the data sets from restore) commands to select only those you data sets you want to restore.
2. In the **X Mask** field, specify the mask for the data sets you do not want restored.
3. When you have the final list of data sets, press END.

   All data sets in the list are sorted by utility.

   Note: If an ABARS data set was selected for restore, use the **Restore Warning** panel to recover the data set through the **Submit ARECOVER** panel or by using skeleton processing. If A is specified, each aggregate's ARECOVER panel is displayed for specific ARECOVER keyword support. If S is specified, the data set is restored using skeleton processing.

Showing SMF information for a search result data set

For any data set in the search result list shown on the **Dataset History Record** panel, you can display the data set's SMF information.

1. From the list of data sets shown on the **Dataset History Record** panel, type S in the **C** column adjacent to the name of the data set whose SMF information you want to view, and then press Enter.
2. When you have finished viewing the information on the **Historical Dataset Usage Detail Record** panel, press the END function key to display the **Dataset History Record** panel.

Viewing evaluation results for a search result data set

Directly from the search results list on the **Dataset History Record** panel, you can view the evaluation results for an individual data set.

1. From the list of data sets shown on the **Dataset History Record** panel, type E in the **C** column adjacent to the name of the data set whose evaluation results you want to view, and then press Enter.
2. When you have finished viewing the information on the **Historical Dataset Usage Detail Record** panel, press F3 key to display the **Dataset History Record** panel.

Viewing backup information for a search results data set

Directly from the historical data set usage search results list on the **Dataset History Record** panel, you can view information about the last backup or all backups for any selected data set.

1. From the list of data sets shown on the **Dataset History Record** panel, use the appropriate line command:
   - Type L in the **C** column adjacent to the name of the data set whose most recent backup you want to view, and then press Enter.
   - Type A in the **C** column adjacent to the name of the data set for which you want to view all backups, and then press Enter
2. When you have finished viewing the information on the **Search for Dataset** panel, press END to display the **Dataset History Record** panel.

Chapter 22. Monitoring in-progress ABARS backups and recoveries

Use the Advanced Backup and Recovery for z/OS Monitor feature to watch ABARS backup and recovery operations as they execute.

The Monitor feature associates the ABARS started task with an aggregate and the job executing it.

Note: Make sure `WTPMSG` and `MSGID` options are specified for Advanced Backup and Recovery for z/OS; by default, `NOWPTMSG` and `NOMSGID` are specified when a TSO ID is created. Otherwise, the **ABARS Monitor** panel does not display the appropriate messages.

1. Type 7 in the **Command** line on the **Main Menu** panel, and then press Enter to display the **Monitor** panel.
2. Type 1 in the **Command** line on the **Monitor** panel, and then press Enter to display the **ABARS Monitor Search Setup** panel.

   Be aware that if you specified / in the **Don't show this panel again** field during a previous visit to this panel, the **ABARS Monitor** panel appears immediately after you select the Monitor option on the **Main Menu** panel.
3. Use the **ABARS Monitor Search Setup** panel's online help to guide you as you specify the criteria that identify the aggregate, which type of backup event you want to monitor, a specific job name or mask, or a specific step name or mask, or a combination of these, and then press Enter to display the **ABARS Manager Monitor** panel.
4. Use the **ABARS Manager Monitor** panel's online help to guide you as you complete the fields to further refine the list of suitable backup and recovery events, and then press Enter to refresh the list.
5. Type ST in the **Command** line on the **ABARS Monitor** panel and then press Enter to display the **ABARS Manager Monitor Status** panel, which shows information about your DFSMShsm environment and your defined ABARS information settings.

The following figure shows an example **ABARS Manager Monitor Status** panel.

```
                ABM - ABARS Manager Monitor Status

  Command ===> _____________________________________________________

SMS Information:               ABARS Information:
DFSMS/HSM Level   => 02.20     Procname                   => BKMABARS
DFSMS/HSM Host ID => 1         Max Tasks                  => 05
Authorized Userid => DFHSM     Active tasks               => 04
ARCCMDxx Suffix   => DV        I/O Buffers                => 0000
Tasks Active:                  Activity Log Prefix        => HSMACT
- Volume Dump     => 00        Activity Log Message Level => FULL
- Volume Backup   => 00        Activity Log Type          => DASD
                               Activity Log Sysout Class  => N/A
DFP/DFSMS Level   => 02.10     ABACKUP Unit Name          => CART
DFDSS Level       => 02.10     ARECOVER Unit Name         => 3590-1
                               ARECOVER ML2 Unit Name     => 3590-1
                               ARECOVER Percent Utilized  => 080
Press END(PF3) to Return
```

.

Chapter 23. Viewing event history

You can view events based on their backup or recovery date. This option is available from the **Main Menu** and the **Full Volume Dumps** panels. Line commands available with this option enable you to view event detail.

1. Access event history using one of these options:
   - From the **Main Menu**, select the **History** option.
   - From the **Full Volume Dumps** panel, type H at the **Command** prompt.

   The **History Search Setup** panel is displayed.
2. Specify the search criteria:
   a) In the **Name or Mask** field, specify a name or mask for the events you want to view (up to eight characters in length). The default is * (all events).
   b) In the **Event Date** field, specify date criteria or a specific date. Specific dates must be specified in the format: YYYY/MM/DD or MM/DD/YYYY. Relative date masking is permitted. If you do not specify a date, all dates will be displayed.
   c) In the **Display ABARS** field, specify Y to display ABARS events tracked by ABARS Manager.
   d) In the **Display CBTI** field, specify Y to display events tracked by Critical Backup Tracking and Inventory.
   e) In the **Store values to be used as default for this USERID** field, type a forward slash (/) to store the volume name or mask as specified on this panel. If selected, the next time the **History Search Setup** panel is displayed for your user ID, the **Name or Mask** field, and all other fields will contain your most recent search criteria.

      To view or change the **History Search Setup** panel after it has been stored, type S or ? in the **C** field on the **History** panel.
   f) In the **Don't show this panel again** field, type a forward slash (/) to display the **History** panel when you select the **History** option from the Main Menu.
3. Press F3.

   The **History** panel is displayed.

History panel fields and commands

The **History** panel displays all events found in the Advanced Backup and Recovery for z/OS Inventory Data Set that match the search criteria you specified on the **History Search Setup** panel.

Note:

- For Critical Backup Tracking and Inventory events, the event and data set records are retained if the most recent BKMEXPIR job found that the output files from the backup were still cataloged.
- For ABARS Manager events, the event entry is retained if it is within the range of backup generations (or recover generations) to retain specified on the **Aggregate Management w/SMS Interface** panel.
- If the job name contains dashes, it means that either the job is still active, or that the job did not complete.

Fields

Display ABARS
In the **Display ABARS** field, specify Y to display ABARS events tracked by ABARS Manager.

Display CBTI
Specify Y to display events tracked by Critical Backup Tracking and Inventory.

C
Command input. The field next to an entry in the list enables you to specify a line command for that particular entry.

Date
The date of the event. Date searches are performed against the most current generation only. Relative date masking is permitted. You can also specify a specific date in the format of YYYY/MM/DD or MM/DD/YYYY. In this case, only events matching that date will be displayed. If the date field is left blank, all dates are displayed.

Time
The time the event occurred.

Jobname
The name of the job.

Name
Specify a name or mask for the backup or recovery you want to view (maximum of eight characters).

PGM/Type
The program or utility that performed the backup:

- ADR DLTA — ADARUN DELTA
- ADR FULL — ADARUN FULL
- AB VER — ABARS ABACKUP Verify
- AB EXEC — ABARS ABACKUP Execute
- AR VER — ABARS ARECOVER Verify
- AR EXEC — ABARS ARECOVER Execute
- AR PREP — ABARS ARECOVER Prepare
- ARCINBAK — HSM Incremental Backups
- BACKVOL — HSM BACKVOL Command
- BCS CR+ — Advanced Catalog Management BCS Backup
- DSN CR+ — Advanced Catalog Management DSN Backup
- VVDS CR+— Advanced Catalog Management VVDS Backup
- CA-FAVER — CA—FAVER
- CFC REP — CFCAMS REPRO
- CFC EXP — CFCAMS EXPORT
- DSS LD — DSS LOGICAL DUMP
- DSS PD — DSS PHYSICAL DUMP
- DSS FVD — DSS FULL VOLUME DUMP
- DSS ZFS— DSS Unix Files Backup
- FDR ABR — FDR ABR ABR DUMP
- FDR APPL — FDR APPL DUMP
- FDR DSF — FDR ABR DSF DUMP
- FDR FDR — FDR ABR FDR DUMP
- FDR FVD — FDR FULL VOLUME DUMP
- HSMAUTO — HSM AUTODUMP
- IDC EXP — IDCAMS EXPORT
- IDC REP — IDCAMS REPRO
- ICEGENER — ICEGENER
- IEBGENER — IEBGENER
- IEBCOPY — IEBCOPY

- SORT — SORT
- SYNCGENR — SYNCGENR

RC UTIL
Return code from ABARS or Critical Backup Tracking and Inventory processes.

RC BKM
Return code from the BKM process.

Line commands

Specify line commands in the **C** field next to an entry in the list.

BKM
Display the **Jobname List** panel for selected item.

N
Display the **View Dataset Name List** panel to view a list of data set names backed up or Unix files backed up using ADRDSSU utility and recovered.

S
Display the **View Backup Detail** panel to view backup and recovery detail.

VA
(ABARS Manager only) View the activity log produced by BACKUP or ARECOVER.

VE
(ABARS Manager only) View errors.

Chapter 24. Using the Fast Replication Services feature

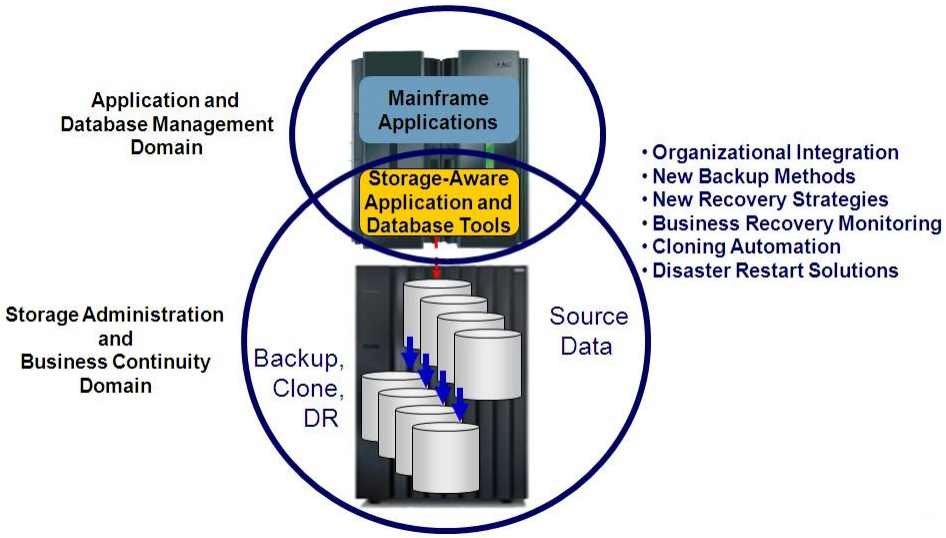
The Fast Replication Services (FRS) feature is a storage-aware backup and recovery solution that integrates storage processor fast-replication facilities with file based backup and recovery operations to allow instantaneous backups, reduce recovery time, and simplify disaster recovery procedures while using less CPU, I/O, and storage resources.

FRS provides a fast and easy-to-use implementation of a fast-replication backup and recovery methodology. It reduces backup windows by leveraging storage-based fast-replication such that backups of large amounts of data can be performed in seconds or less. It simplifies backup and recovery methodologies by allowing application level or selected data set recoveries to be performed from a point-in-time backup. Consistent backups can be created using application level point-in-time Fast Replication Services.

It provides application-level backup and recovery support even for complex applications, where all of the application's data must be backed up, restored, and recovered as a unit. In addition, when creating application level backups, FRS invokes storage-based fast-replication facilities using a Fast Replication Engine (FRE) through appropriate storage processor APIs, reducing host CPU and I/O resource utilization.

About application, database, and storage integration

This topic provides an overview of storage aware database and application utilities and describes how they are used to simplify administration tasks.



Storage aware application and data base tools provide facilities that link and coordinate application and data management organizations with storage and business continuity administrators. Using storage-based fast replication to perform traditional data management copy functions allows us to do all of the following:

- Implement new backup, restore, and recovery methods
- Simplify business continuity monitoring
- Perform automated data cloning operations
- Transform tedious disaster recovery processes into efficient disaster restart procedures

Implementing a storage-based backup methodology or data cloning solution are new concepts to many application or data base administrators because there have been no tools available that could coordinate

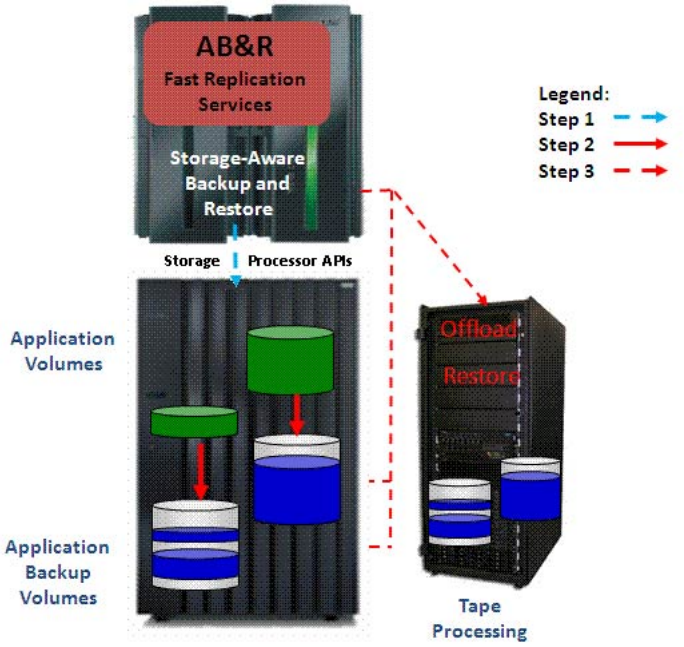
the application, data base meta data, and schema definitions with the corresponding storage processor commands to facilitate non-intrusive backup or cloning processes.

Storage-aware data base or application tools provide the necessary integration and coordination to implement fast and effective storage-based backup, restore, and data cloning methodologies.

About the storage aware backup process

This topic explains the steps involved in the volume-based fast replication backup process.

The following diagram shows the fast replication backup process:



Step 1 of the backup process is Advanced Backup and Recovery for z/OS performing application discovery to identify which data sets need to be backed up and determining on which volumes they reside. The applications identified by the discovery process must be suspended prior to performing the backup.

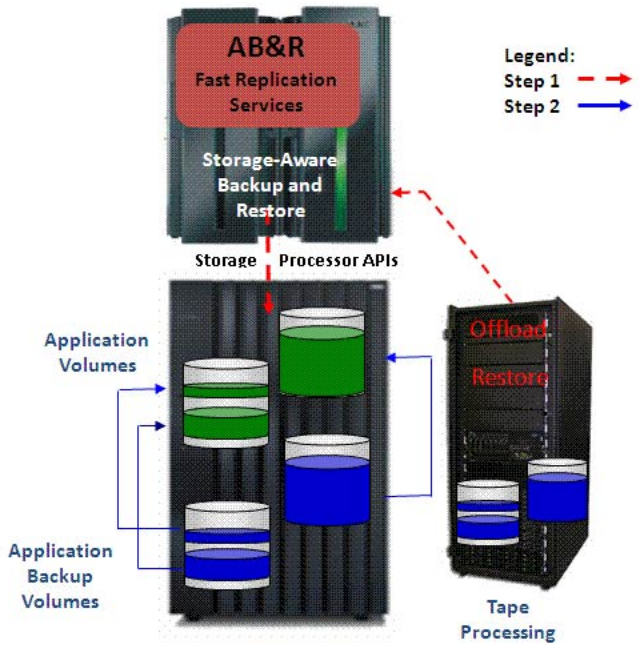
In Step 2, the appropriate fast replication backup performs the application level backup in the storage system. From the applications' perspectives, the backup is instantaneous. The backup can be used for local site application restore and for disaster recovery purposes. The application restore process uses data set level fast replication to snap data sets from the backup volumes to the source data set volumes. The volume-based fast replication facility provides an integrated meta data repository that maintains information about the application backup volumes, backup times, and so forth that can be used to maintain the system and perform application restores.

In Step 3, automated data set level offload management, fast replication backup supports tape offloading of disk-based backups. The tapes can be encrypted and can use DFDSS or FDR to perform the offload process.

About the storage aware restore process

This topic explains the steps involved in the volume-based fast replication restore process.

The following figure shows the fast replication restore process steps:



In Step 1, Advanced Backup and Recovery for z/OS automatically restores the applications from tape or disk. If a data set needs to be restored, it is restored at the data set level.

In Step 2, when Advanced Backup and Recovery for z/OS determines that data sets can be restored from disk, data set based fast replication is used to restore data instantaneously without having to use host CPU or I/O resources. Application recovery can begin as soon as the data restoration process begins, parallelizing the restore and recovery operations to minimize application down time.

Environment

Source volumes containing application data sets to be fast replicated cannot contain any User Catalogs. Fast replication target volumes and their related DFSMS Storage Groups must be dedicated to the FRE (Fast Replication Engine) function.

Hardware and software requirements

To use the Fast Replication Services feature of Advanced Backup and Recovery for z/OS, your traditional mainframe environment must meet the hardware and software requirements documented in this topic.

Software requirements

The Advanced Backup and Recovery for z/OS Fast Replication Services feature (FRS) runs under ISPF, invokes ISPF services, and requires the following software:

- z/OS V1R9 or higher
- FRZ V1.2 or higher

FlashCopy requirement

For FlashCopy backups and restores, FRS requires that the storage subsystem be FlashCopy V2 or higher.

EMC BCV and SNAP requirements

For BCV and SNAP backups, FRS requires the following:

- EMC Symmetrix 6 and higher running with Microcode 5x67 or higher

- To use Enginuity Consistency Assist (ECA), Microcode 5x67 with patch 14882 or Microcode 5x68 with patch 18954 or Microcode 5x69 or higher is required. To use ECA with SNAP VOLUME technology, the EMC SNAP library must be level 5.5 or higher.

DFSMSdss requirements

DFSMSdss must be version 1.8 or higher to use DFSMSdss backups. To perform encryption during offloads, DFSMSdss requires the following software:

- Encryption Facility DFSMSdss Encryption Feature (FMID HCF773D).
- IBM Cryptographic Services Facility (ICSF) HCR770B or higher.

About copy blades

FRS was developed to integrate with and exploit the benefits of storage-based fast-replication technologies. A *copy blade* is the term used to group storage-based fast-replication technologies that storage vendors offer. This topic describes the copy blade types.

Copy blades provide facilities to invoke various data movement and fast-replication facilities. Copy blades provide fast-replication integration and separate the backup and recovery processes from the underlying storage devices.

FRS allows you to take advantage of the specific fast-replication products and features you may have. FRS supports many of the fast-replication products natively, taking full advantage of the storage processor capabilities and fast-replication facilities that are available. Below is a list of the copy blades that FRS supports.

IBM FlashCopy
Used to invoke IBM FlashCopy operations to copy data. This blade is also used to invoke FlashCopy emulation products for EMC and HDS storage systems. For more information, refer to the IBM FlashCopy blade portion of the *Considerations for selecting a copy blade* topic.

EMC TimeFinder
Used to invoke EMC TimeFinder operations to copy data. For more information, refer to the EMC TimeFinder copy blade portion of the *Considerations for selecting a copy blade* topic.

HDS ShadowImage
Used to invoke HDS ShadowImage operations to copy data. For more information, refer to the HDS ShadowImage copy blade portion of the *Considerations for selecting a copy blade* topic.

IBM DFSMSdss
Used to invoke IBM DFDSSdss to perform storage-based fast-replication and slow copy processes. For more information, refer to the IBM DFSMSdss copy blade portion of the *Considerations for selecting a copy blade* topic.

Considerations for selecting a copy blade

Before you configure FRS, it is important that you consult with your storage administrator to determine the type of storage processors used in your environment and the preferred fast replication facilities to use.

Knowing the type of storage processors and available fast-replication facilities will allow you to determine which copy blade, which fast-replication method, and which data consistency mechanism to use for creating an application backup in your environment.

IBM FlashCopy blade

The IBM FlashCopy blade provides interface support for IBM FlashCopy. The FlashCopy blade uses the native IBM ANTRQST macro interface to invoke FlashCopy. This method of invoking FlashCopy is very fast, as it takes a very small amount of time to issue FlashCopy commands which create a point-in-time copy of application data sets at the volume level. All tracks are copied in the background for each volume

backup. FlashCopy does not require the use of SMS or HSM. This reduces the complexity of the backup configuration.

FlashCopy V1 or higher can be used to create application backup and restore operations. When using the FlashCopy blade, the application data must reside on FlashCopy capable storage subsystems and the backup profile must define target volumes that are identical to the source volume model types.

In addition, the source and target volumes must both be located in the same storage subsystem. Users can specify target volume ranges, so there is no need to update a backup profile when the application expands to new volumes.

FlashCopy commands issued through the FlashCopy blade interface make a backup of all the volumes containing application data sets. The FlashCopy blade uses volume based copy services to create the application level backup and it uses volume or data set copy services to restore the application volumes or data sets respectively. Application data set recovery is performed through DFSMSdss. DFSMSdss uses fast-replication if possible and uses host based I/O (slow copy) if the FlashCopy background copy process is not complete.

The IBM FlashCopy blade can support IBM, EMC and HDS storage processors when FlashCopy or FlashCopy emulation products are available for use. The FlashCopy blade can create application level backups for applications that are spread across heterogeneous storage systems when all storage systems are using FlashCopy compatible fast-replication products.

EMC TimeFinder copy blade

The EMC TimeFinder copy blade is used to invoke EMC TimeFinder copy services. The EMC TimeFinder copy blade supports the following EMC copy methods:

- TimeFinder/Mirror
- TimeFinder/Clone Mainframe Volume Snap
- TimeFinder/Snap Mainframe Data Set Snap
- TimeFinder/Snap Virtual Devices
- EMC Enginuity Consistency Assist

FRS users should consult with their storage administrator to determine which EMC TimeFinder copy blade method is best for their environment. Storage administrators should be consulted on the use of EMC Consistency Assist technology in their environment.

EMC TimeFinder/Mirror copy method

The TimeFinder/Mirror copy method uses EMC TimeFinder/Mirror to make a Business Continuity Volume (BCV) backup of an application. A BCV can be established as a mirror of a standard volume and synchronized with the standard volume. Once synchronization is established, then all data written to the standard volume is also written to the BCV.

TimeFinder/Mirror does not require SMS or HSM usage. This reduces the complexity of the backup configuration.

Usage Requirements:

- EMC Symmetrix 6 and higher running with Microcode 5x67 or higher.
- To use Enginuity Consistency Assist (ECA), Microcode 5x67 with patch 14882 or Microcode 5x68 with patch 18954 or Microcode 5x69 or higher is required. To use ECA with SNAP VOLUME technology, the EMC SNAP library must be level 5.5 or higher.

EMC TimeFinder/Clone copy method

TimeFinder/Clone does not require SMS or HSM usage. This reduces the complexity of the backup configuration.

EMC TimeFinder/Clone Mainframe Volume Snap can be performed in phases using a phased snap so performance implications to source volumes can be timed and mitigated. TimeFinder/Clone full volume snap operations are differential so only changed tracks are copied since the last backup was created.

EMC TimeFinder/Snap virtual device usage

TimeFinder/Snap does not require SMS or HSM usage. This reduces the complexity of the backup configuration.

HDS ShadowImage copy blade

The HDS ShadowImage copy blade supports HDS native ShadowImage volume copy processes.

ShadowImage fast-replication copies are incremental copies so only the changed tracks are copied since the last system level backup was created. The incremental process reduces the storage processor overhead associated with copying all the source tracks to the target device for each backup operation.

The HDS ShadowImage copy blade does not require the use of SMS or HSM. This reduces the complexity of the backup configuration.

The ShadowImage copy process requires that the source and target volumes be on the same storage processor. The ShadowImage copy blade supports three backup generations. The ShadowImage copy blade allows coordinated FlashCopy and ShadowImage copy methods to be used. That is, application volumes located on HDS storage processors can use ShadowImage while other source volumes can use FlashCopy fast-replication facilities to perform a backup operation.

Recovery operations use ShadowImage to restore backup volumes. Application recovery operations are performed using DFSMSdss. DFSMSdss uses data set fast replication if it is available; it uses host based I/O (slow copy) if data set fast replication is not available or if the backup background copy process is not complete.

When using the ShadowImage copy blade, the target volumes are kept offline. FRE performs all the necessary commands to bring the volumes online when needed to copy the volumes to tape or to perform application recovery through DFSMSdss.

IBM DFSMSdss copy blade

The DFSMSdss copy blade can be used to create an application level backup.

The DFSMSdss copy blade uses the ADRDSSU interface specifying fast-replication (preferred). The copy blade will drive fast-replication services if they are available and will use host based I/O copy methods if it is not. This copy blade interface can be used to drive data sets or volume based fast-replication facilities.

The FRS backup profile must define target volumes that are identical to the source volume model types.

The DFSMSdss copy blade does not require SMS or HSM usage. This reduces the complexity of the backup configuration.

The DFSMSdss copy blade supports IBM FlashCopy on IBM storage processors and compatible FlashCopy products on EMC and HDS storage processors. It also supports SnapShot on STK or IBM Ramac storage processors. Invoking FlashCopy facilities using the ADRDSSU interface is slower than using the IBM FlashCopy blade which uses the ANTRQST interface.

When using the DFSMSdss blade, the target volumes are kept online and therefore must all have unique volume labels. The z/OS data set placement rules must be set up such that the target units will not be used for new data sets or altered by other processes since they are kept online at all times.

Using Fast Replication Services to drive the Fast Replication Engine

This topic provides information about preparing to use FRS, creating and maintaining profiles, and running FRS.

Generally, using the Fast Replication Services feature involves these tasks:

- Preparing FRS for use by installing the Fast Replication Engine (FRE), defining the control files and repository files, APF authorizing the FRE load library, and setting the BKMINI initialization parameters
- Creating, updating, and managing application backup profiles

- Running Fast Replication Services
- Using the Offload Tape List

Note:

- Use the Action bar's **Diagnostic** menu to determine the availability of the Fast Replication Services feature on the system where you are currently logged on. Select the **About** item and use the scroll function keys to view the appropriate information.
- Use the Action bar's **Menu** option to view Fast Replication Services messages. Select the **Messages** item and specify the **F** option to view the messages of interest.

Preparing to use FRS

Before you can use the Fast Replication Services feature, you must perform some preparation tasks.

The following is an overview of preparation tasks, in the recommended order:

1. Install the FRE product and configure the Fast Replication target volume environment.
2. APF authorize the FRE LOADLIB.
3. Define and initialize the Fast Replication Repository files.
4. Define and initialize the Fast Replication Control File.
5. Specify the Advanced Backup and Recovery for z/OS Fast Replication Services initialization parameters.
6. (Optional) Make any necessary changes to the product sample library FRZ#PARM member.
7. (Optional) Create and schedule a job to back up the Fast Replication Repository and Control Files.

Refer to the following topics for more detailed information about performing each task.

Step 1: Install FRE and configure target volumes

This topic tells you how to install the FRE product and find documentation that provides instructions for configuring the target volume environment.

1. Use SMP/E to apply FMID HFRZ120.
2. Ensure that the application source and fast target volumes reside in the same storage subsystem.
3. Dedicate the target volumes to the Advanced Backup and Recovery for z/OS Fast Replication Services function.

   It is recommended that you use a dedicated SMS Storage Group for the fast replication target volumes.

Step 2: APF authorize the FRE LOADLIB

Before you can use the Fast Replication Services feature, you must APF authorize the FRE LOADLIB.

Consult your systems programmer to have this library added to the APF list and ensure appropriate access controls have been established.

Step 3: Define and initialize the Fast Replication Repository files

Use the Advanced Backup and Recovery for z/OS product JCL library member BKM#REPO to define the Fast Replication Repository Files.

1. In an ISPF EDIT session, open the product JCL library's BKM#REPO member.
2. Use the instructions in the comments section to customize the member appropriately for your environment.
3. Submit the job for execution.

   When the job has executed successfully, the Fast Replication Repository Files have been defined and initialized.

Step 4: Define and initialize the Fast Replication Control File

Use the Advanced Backup and Recovery for z/OS product JCL library member BKM#CNTL to define and initialize the Fast Replication Control File.

The FRS repository files must be created before the control file is initialized.

1. In an ISPF EDIT session, open member BKM#CNTL from the product's JCL library.
2. Use the instructions in the comments section of BKM#CNTL to customize it appropriately for your environment.
3. Submit the job for execution.

   When execution completes successfully, the control file has been created and initialized.

Step 5: Specify the Fast Replication initialization parameters

Update the BKMINI member in the product parameter library by supplying the appropriate values for the Fast Replication Services tokens.

1. Use and ISPF EDIT session to open the BKMINI member in the product parameter library.
2. In the :PRODUCT_SELECTABLE_UNITS section, specify FRS_ACTIVE=YES.
3. In the :PRODUCT_DATASETS section, make these specifications:
   a) FRZ_CNTL_FILE_DSN=*filename*, where *filename* is the name of the Fast Replication Engine Control File.
   b) FRZ_LOADLIB=*filename*, where *filename*, where *filename* is the name of the Fast Replication Engine load library.

      Note: The Fast Replication Engine load library must be APF authorized.
   c) FRZ_ISPPLIB=*libname*, where *libname* is the name of the Fast Replication Engine ISPPLIB. Refer to the FRE installation documentation for detailed information.
   d) FRZ_ISPMLIB=*libname*, where *libname* is the name of the Fast Replication Engine ISPMLIB. Refer to the FRE installation documentation for detailed information.
   e) FRZ_MSGLIB=*libname*, where *libname* is the name of the Fast Replication Engine messages library. Refer to the FRE installation documentation for detailed information.

Step 6: (Optional) Make any necessary changes to FRE sample library member FRZ#PARM

Use the information in this topic to specify nondefault values for the parameters in the Fast Replication Engine (FRE) sample library member FRZ#PARM.

The default values provided in FRZ#PARM are appropriate for most sites. However, your environment might make it necessary to override the default values for some of these parameters.

1. In an ISPF EDIT session, open *hlq*.SFRZSAMP(FRZ#PARM). Refer to the FRE product installation results to determine the data set name.
2. Use the parameter descriptions to locate the appropriate parameters and to specify valid values for them.

   General backup and recovery parameters

   PARMLIB_VERSION
   Identifies the product parameter library member version for Advanced Backup and Recovery for z/OS. By default, the current version number is assigned to this parameter and should not be changed.

   Note: Do not change the value assigned to this parameter.

   GENERATED_JOB_REGION

   The job card REGION, in terms of megabytes. Valid values are integers in the range 0 - 256.

Default: 0

ROUTE_ALL_ON_CONSOLE_CMDS

Specifies whether console commands are prefixed with RO *ALL. Valid values are:

- Y indicates the console commands are prefixed.
- N indicates the console commands are not prefixed.

Default: Y

Note: If your z/OS system is not part of a sysplex, you may change this parameter value to N; otherwise, do not change the default value.

DASD_ALLOCATION_UNIT

Specifies the allocation unit to be used for dynamic allocations.

Default:SYSALLDA

TEMP_DSN_ALIAS

Specifies the data set high level alias to be used for creating temporary data sets. The default is the TSO user ID of the job submitter. You may also specify a literal for this parameter.

Default: FRE

FCTOPPRCP

If the target specified is a primary device in a PPRC relationship, valid values are:

- N—Do not allow the PPRC primary to become a FlashCopy target.
- Y—The pair can go into a duplex pending state.
- P—It is preferable that the pair not go into a duplex pending state.
- R—It is required that the pair not go into a duplex pending state.

Default: N

BACKUP_WORK_DATASETS

Specifies whether work data sets are considered to be critical and therefore are backed up. Valid values are:

- Y—Work data sets are critical. If the work data sets are *not* included in the system backup, an error message is issued.
- N—Work data sets are not critical and no error message is issued if the work data sets are not included in the system backup.

Default: Y

Profile setup utility parameters

ABEND_ON_ERRORS

Specifies whether the backup profile setup utility ABENDs when it encounters an error. Valid values are:

- Y—Produce a z/OS ABEND code when errors are encountered.
- N—The utility does not ABEND when it encounters errors.

Default: N

USER_ABEND_RETURN_CODE

Specifies the error code returned when the backup profile setup utility encounters an error and you specified Y for the **ABEND_ON_ERRORS** parameter.

Valid values are integers in the range 01-99, inclusive.

Default: 08

RELEASE_HELD_VOLUMES

Specifies what happens to held volumes encountered during the backup profile setup process. Valid values are:

- Y—Release the held volumes.
- N—Held volumes are marked unusable and an error is produced.

Default: Y

Note: Do not use RELEASE_HELD_VOLUMES=N in conjunction with PLACE_BKUP_VOLS_ON_HOLD=Y. Doing so causes Advanced Backup and Recovery for z/OS to produce an error when it encounters volumes placed on hold by the backup profile setup process.

PLACE_BKUP_VOLS_ON_HOLD

Specifies whether the backup profile setup utility places future target volumes on hold. Valid values are:

- Y—Place the future target volumes on hold.
- N—Do not place the future target volumes on hold.

Default: N

CLEAN_OLD_CONSIST_WINDOWS

Specifies whether non-active consistency windows are cleaned by the backup profile setup utility. Valid values are:

- Y—The backup profile setup utility cleans the non-active consistency windows.
- N—The backup profile setup utility produces an error and terminates the setup.

Default: Y

CLEAN_OLD_SNAP_SESSIONS

Specifies whether non-active SNAP sessions are cleaned from the volume before the profile setup process begins. Valid values are:

- Y—The backup profile setup utility cleans the non-active SNAP sessions.
- N—The backup profile setup utility produces an error and terminates the setup.

Default: Y

Note: Specify CLEAN_OLD_SNAP_SESSIONS=Y.

SYNC_ALL_BCV_GENERATIONS

Specifies whether Advanced Backup and Recovery for z/OS establishes all generations of target BCVs to their source volumes on the very first setup run of a backup profile. Valid values are:

- Y—Advanced Backup and Recovery for z/OS establishes all generations of target BCVs to their source volumes on the very first setup run of a backup profile.
- N—Advanced Backup and Recovery for z/OS does not establish all generations of target BCVs to their source volumes on the very first setup run of a backup profile.

Default: N

Note: The value of this parameter is used for only the initial setup run of a backup profile.

VALIDATE_VOLUME_SUBTASKS

Specifies the maximum number of subtasks used by Advanced Backup and Recovery for z/OS. Valid values are integers in the range 1–99, inclusive.

Default: 1

VALIDATE_VOLUMES

Specifies how often the backup profile setup utility validates backup profile source volumes. Valid values are:

- A—(Always) Validate the backup profile source volumes for every setup run for a profile.
- F—(First) Validate the backup profile source volumes for only the initial setup run for a profile.

Default: A

MAKE_READY_NOTREADY_DEVICES

Specifies what the backup profile setup utility does when it encounters a held device. Valid values are:

- Y—Make held devices ready.
- N—Produce an error.

Default: Y

MAKE_BKUP_VOLS_NOTREADY

Specifies whether the backup profile setup utility puts future generations of target volumes in the not-ready state. Valid values are:

- Y—Put future generations of target volumes in the not-ready state.
- N—Do not put future generations of target volumes in the not-ready state.

Default: N

Backup utility parameters

ABEND_ON_ERRORS

Specifies whether the backup utility ABENDS when it encounters an error. Valid values are:

- Y—Produce a z/OS ABEND error code when an error is encountered.
- N—When errors are encountered, produce an error code having the value specified by the **USER_ABEND_RETURN_CODE** parameter.

Default: N

USER_ABEND_RETURN_CODE

Specifies the error return code produced if ABEND_ON_ERRORS=Y. Valid values are integers in the range 01-99, inclusive.

Default: 08

Note: If ABEND_ON_ERRORS=N, the program error return code is provided.

WAIT_FOR_VOLUME_SYNC

Specifies what the backup utility does in situations where a target BCV is not synchronized to its source volumes at the time of the backup. Valid values are:

- P—The backup utility issues a WTOR.
- Y—The backup utility waits automatically for the BCV to synchronize.
- N—The backup utility issues an error return code if the previous generation of BCVs has not yet fully synchronized.

Default: Y

PLACE_BKUP_VOLS_ON_HOLD

Specifies whether the backup utility to place backup volumes on hold. Valid values are:

- Y—Place the backup volumes on hold.
- N—Do not place the backup volumes on hold.

Default: N

RELEASE_HELD_VOLUMES

Specifies what the backup utility does in situations where a target BCV is not synchronized to its source volumes at the time of the backup. Valid values are:

- P—The backup utility issues a WTOR.
- Y—The backup utility waits automatically for the BCV to synchronize.
- N—The backup utility issues an error return code if the previous generation of BCVs has not yet fully synchronized.

Default: Y

CLEAN_OLD_CONSIST_WINDOWS

Specifies whether non-active consistency windows are cleaned. Valid values are:

- Y—The backup utility cleans the non-active consistency windows.
- N—The backup utility produces an error and terminates the backup.

Default: Y

Note: Specify `CLEAN_OLD_CONSIST_WINDOWS=Y`.

CLEAN_OLD_SNAP_SESSIONS

Specifies whether non-active SNAP sessions are cleaned from the volume prior to the backup. Valid values are:

- Y—The backup utility cleans the non-active SNAP sessions.
- N—The backup utility produces an error and terminates the backup.

Default: Y

Note: Specify `CLEAN_OLD_SNAP_SESSIONS=Y`.

CONSIST_TIME_OUT_SECONDS

Specifies the maximum number of seconds to suspend I/O on standard volumes during a backup. Valid values are integers in the range 01-256, inclusive.

Default: 256

BKUP_VALID_ON_CONSIST_FAIL

Specifies whether the backup is still registered in situations where a consistency window cannot be obtained or the window closes before the split or SNAP completes. Valid values are:

- Y—The backup remains registered in the situation described.
- N—The backup does not remain registered in the situation described.

Default: N

VALIDATE_VOLUMES

Specifies how often the backup utility determines which volumes to include in its list of volumes in use by applications. Valid values are:

- P—Determine the list according to the profile settings.
- A—Always determine the list.
- D—Determine the list daily.
- W—Determine the list weekly.

Default: P

VALIDATE_VOLUMES_TIME

Specifies when in the backup process source volume validation is performed. Valid values are:

- B—Source volume validation is performed before the backup is taken.
- A—Source volume validation is performed after the backup is taken.

Default: B

VALIDATE_VOLUME_SUBTASKS

Specifies the maximum number of subtasks used by Advanced Backup and Recovery for z/OS. Valid values are integers in the range 1–99, inclusive.

Default: 1

WAIT_FOR_VOL_OFFLINE_SECONDS

Specifies how many seconds to wait for a volume to go offline. Valid values are integers in the range 1-99.

Default: 05

WAIT_FOR_VOL_OFFLINE_RETRIES

Specifies the maximum number of retries to attempt while waiting for a volume to go offline. Valid values are integers in the range 1-99.

Default: 99

MAKE_READY_NOTREADY_DEVICES

Specifies what the backup utility does when it encounters a held volume. Valid values are:

- Y—Make held volumes ready.
- N—Produce an error.

Default: Y

MAKE_BKUP_VOLS_NOTREADY

Specifies whether the backup utility makes backup volumes not-ready. Valid values are:

- Y—Make backup volumes not-ready.
- N—Do not make backup volumes not-ready.

Default: N

RESET_COPY_PENDING_TS

Specifies whether to reset the COPY PENDING status on table spaces after a system backup. Valid values are:

- Y—Reset the COPY PENDING status.
- N—Leave the table space in COPY PENDING status.

Default: Y

RESET_COPY_PENDING_IX

Specifies whether to reset the COPY PENDING status on index spaces after a system backup. Valid values are:

- Y—Reset the COPY PENDING status.
- N—Leave the index space in COPY PENDING status.

Default: Y

ALLOW_SHARED_TARGET_VOLUMES

Specifies whether to allow SNAP or Flash target volumes to be shared amongst different backup profiles. Valid values are:

- Y—Allow target volume sharing.
- N—Do not allow target volume sharing

Default: Y

Note: Specify `ALLOW_SHARED_TARGET_VOLUMES=Y`.

RECALL_MIGRATED_WAIT

Specifies the maximum number of minutes Advanced Backup and Recovery for z/OS waits for recall of migrated data sets.

Default: 15

RECALL_MIGRATED_DATA

Specifies how migrated data sets are handled when they are encountered during a backup. Valid values are:

- E—Advanced Backup and Recovery for z/OS issues an error message and halts the backup.
- I—Advanced Backup and Recovery for z/OS ignores migrated data sets and continues performing a partial backup.
- R—Advanced Backup and Recovery for z/OS waits until all migrated data sets have been recalled, then continues with the backup.

Default: E

SHADOW_IMAGE

Specifies whether you want FlashCopy to use ShadowImage commands on Hitachi Data Systems hardware. Valid values are:

- Y—Use ShadowImage commands on HDS hardware.
- N—Do not use ShadowImage commands on HDS hardware.

Default: Y

Note: You may not specify `SHADOW_IMAGE=Y` if your site does not have a FlashCopy emulation license for HDS.

SNAP global parameters

MAX_RETURN_CODE

Specifies the maximum EMC SNAP command return code value that can be returned without stopping processing. If a return code greater than this value is produced, processing stops. Valid values are 4, 8, 12, or other integers.

Default: 12

SNAP_WAIT

Specifies whether the backup utility waits for the SNAP operation to complete before ending the job. Valid values are:

- Y—The backup utility waits for the SNAP operation to complete before ending the job.
- N—The backup utility can end while the SNAP operation completes the Symmetrix caches updates to the source volume.

Default: Y

SNAP_WAIT_HOURS

Specifies the number of hours the backup utility waits for the SNAP operation to complete before continuing. Valid values are integers.

Default: 00

Note: This setting is ignored if `SNAP_WAIT=N`.

SNAP_WAIT_MINUTES

Specifies the number of minutes the backup utility waits for the SNAP operation to complete before continuing. Valid values are integers.

Default: 02

Note: This setting is ignored if `SNAP_WAIT=N`.

SNAP_WAIT_SECONDS

Specifies the number of seconds the backup utility waits for the SNAP operation to complete before continuing. Valid values are integers.

Default: 30

Note: This setting is ignored if `SNAP_WAIT=N`.

MAX_ADRDSSU

Specifies the maximum number of address spaces that can be spawned.

Default: EMC default

MAX_TASKS1

Specifies the maximum number of individual requests that can be attached and used.

Default: EMC default

MAX_TASKS2

Specifies the maximum number of individual activities that can be performed within an individual request.

Default: EMC default

DEBUG_MODE

Specifies the amount of debugging information written to the BKMSNAPO DD. Valid values are:

- A—All
- T—Trace
- D—Dump
- E—Error
- X—Extra

Default: A

DEBUG_EXTENTS

Specifies whether to include in the debug information data about extents. Valid values are:

- Y—Yes, include extent information in the debug information.
- N—Do not include extent information in the debug information.

Default: Y

TOLERATE_ENQ_FAILURES

Specifies whether a volume can be SNAPped when exclusive specialization cannot be obtained. Valid values are:

- Y—Yes, allow the volume to be SNAPped.
- N—Do not allow the volume to be SNAPped.

Default: EMC default

Note: This parameter should be set to Y.

COPY_VOLUME_ID

Specifies whether the source volume VOLSER is copied to the target volume. Valid values are:

- Y—Yes, copy the source volume's VOLSER to the target volume.
- N—Do not copy the source volume's VOLSER to the target volume.

Default: Y

PHASED_SNAP

Specifies whether phased SNAP or SNAP group processing is enabled, breaking the EMC SNAP VOLUME command into separate phases. Valid values are:

- Y—Enable phased SNAP.
- N—Do not enable phased SNAP.

Default: Y

Note: Setting PHASED_SNAP=Y improves performance and lessens the impact of SNAP VOLUME commands on the storage array.

SNAP_GROUP_PDS

Specifies the name of the partitioned data set that is used internally to store the phased SNAP commands that need to be executed when PHASED_SNAP=Y.

Note: It is recommended that you use an extended PDS (PDSE) for this purpose.

WAIT_FOR_BACKGROUND_COPY

Specifies whether Advanced Backup and Recovery for z/OS waits for background SNAPs to complete before letting the backup complete. Valid values are:

- Y—Wait for background SNAPs to complete.
- N—Do not wait for background SNAPs to complete.

Default: N

BCV split utility parameters

BCV_WAIT_SECONDS

Specifies how many seconds Advanced Backup and Recovery for z/OS waits between each check for BCV split completion. Valid values are integers in the range 01-99, inclusive.

Default: 06

WAIT_RETRIES

Specifies how many times Advanced Backup and Recovery for z/OS checks for BCV split completion. Valid values are integers in the range 01-99, inclusive.

Default: 99

Restore utility parameters

ABEND_ON_ERRORS

Specifies whether the restore utility issues a z/OS ABEND code when errors are encountered. Valid values are:

- Y—Return a z/OS ABEND code when errors are encountered by the restore utility.

- N—When the restore utility encounters errors, it provides the program error return code.

Default: N

Note: If you specify ABEND_ON_ERRORS=Y, the error code returned by the restore utility will have whatever value you assign to the USER_ABEND_RETURN_CODE parameter.

USER_ABEND_RETURN_CODE

Specifies the z/OS ABEND code you want returned by the restore utility when it encounters errors. Valid values are integers in the range 01-99, inclusive.

Default: 08

Note: If ABEND_ON_ERRORS=N, the value you assign to the USER_ABEND_RETURN_CODE is ignored.

WAIT_FOR_VOL_OFFLINE_SECONDS

Specifies how many seconds the restore utility waits for a volume to go offline. Valid values are integers in the range 01-99, inclusive.

Default: 06

WAIT_FOR_VOL_OFFLINE_RETRIES

Specifies how many retries the restore utility attempts while waiting for a volume to go offline. Valid values are integers in the range 01-99, inclusive.

Default: 99

WAIT_FOR_VOL_ONLINE_SECONDS

Specifies how many seconds the restore utility waits for a volume to go online. Valid values are integers in the range 01-99, inclusive.

Default: 06

WAIT_FOR_VOL_ONLINE_RETRIES

Specifies how many retries the restore utility attempts while waiting for a volume to go online. Valid values are integers in the range 01-99, inclusive.

Default: 99

CLEAN_OLD_SNAP_SESSIONS

Specifies whether to clean (remove) inactive SNAP sessions (from other source volumes) that reside on a volume. Valid values are:

- Y—Clean the inactive SNAP sessions.
- N—Do not clean the inactive SNAP sessions and end the restore process with an error.

Default: Y

Note: Specify CLEAN_OLD_SNAP_SESSIONS=Y.

FORCE_SPLIT

For BCV profiles, this parameter specifies whether the current BCV generation is split before the restore. Valid values are:

- Y—Split the current BCV generation prior to starting the restore.
- N—Do not split the current BCV generation prior to starting the restore.

Default: Y

Note: Specifying FORCE_SPLIT=Y causes the FORCE parameter to be added to the split call.

PERFORM_CHECKSUM

Specifies whether to ensure that a restored volume has not been altered since it was backed up by performing a checksum operation is performed on that volume. Valid values are:

- Y—Perform the checksum operation on the restored volume.
- N—Do not perform the checksum operation on the restored volume.

Default: Y

CLEAR_CF_STRUCTURES

Specifies whether to clear the coupling facility structure for data sharing systems. Valid values are:

- Y—Clear the coupling facility structure.
- N—Do not clear the coupling facility structure.

Default: Y

DB2_UTILITY_SUITE_INSTALLED

Specifies whether the Db2 Version 8 System Restore utility is available for the recovery process. Valid values are:

- Y—The Db2 Version 8 System Restore utility is available for the recovery process.
- N—Use the Recovery utility for the recovery process.

Default: Y

Offload utility parameters

ABEND_ON_ERRORS

Specifies whether the offload utility issues a z/OS ABEND code when errors are encountered. Valid values are:

- Y—Return a z/OS ABEND code when errors are encountered by the offload utility.
- N—When the offload utility encounters errors, it provides the program error return code.

Default: N

Note: If you specify ABEND_ON_ERRORS=Y, the error code returned by the offload utility will have whatever value you assign to the USER_ABEND_RETURN_CODE parameter.

USER_ABEND_RETURN_CODE

Specifies the z/OS ABEND code you want returned by the offload utility when it encounters errors. Valid values are integers in the range 01-99, inclusive.

Default: 08

Note: If ABEND_ON_ERRORS=N, the value you assign to the USER_ABEND_RETURN_CODE is ignored.

CLIP_PREFIX

Specifies the prefix to used when clipping VOLSERs during offload processing.

Default: FF

Note: Advanced Backup and Recovery for z/OS might need to clip (or change the sequence of) a target volume VOLSER to bring the target online.

Step 7: (Optional) Create and schedule a job to back up the Fast Replication Repository and Control Files

Follow the procedure provided in this topic to build a job that backs up the Fast Replication Services database and control files.

The Fast Replication Services database is required at a remote location if the application data set backups are required for remote application recovery.

1. Use the sample job in member BKM#BKUP of the product's JCL library to create your backup job.
2. Schedule execution of the backup job at the appropriate time for your environment.

Creating application backup profiles

Use the information in this topic to access the backup profile list and to create and manage backup profiles. This topic also provides information about offload options and reports.

This section explains how to create and maintain application backup profiles using the FRS ISPF interface. Application backup profiles contain information that is used by the backup jobs. Using the FRS ISPF interface, you can create an application backup profile and specify the fast replication target volume units or DFSMS Storage Groups to be used.

In addition, you can set other backup options such as the number of generations to keep, whether to require backup offloads and backup offload attributes. Backup profiles are reusable and editable, and are created on a per storage subsystem basis. You can easily rename and delete backup profiles using line commands.

Accessing the backup profile list

Use the information in this topic to access a list of the existing application backup profiles.

1. From the **Main Menu**, select the **Setup and Configuration** option, then press Enter to display the **Setup and Configuration** menu panel.
2. From the **Setup and Configuration** menu, select the **Fast Replication Services** option, and then press Enter to display the **Fast Replication Engine Profile Service** panel.

```
---------- BKM - Fast Replication Engine Profile Service ----------ssN
e                                                                    e
e Command ===>                                                       e
e                                                                    e
e Specify specific search criteria or masks.                         e
e                                                                    e
e Profile Creator  ===> *         ( *  for all profiles)             e
e                                                                    e
e Profile Shr Opt  ===> *         (Share Option: N or U or V or * )  e
e                                                                    e
e Profile Name     ===> *                              (Name or Filt)  e
e                                                                    e
e Profile Desc     ===> *                              (Desc or Filt)  e
e                                                                    e
e Sort By          ===> T  ( C Creator D Description  N Name         e
e                           S Share Option T Type)                   e
e Sort Direction   ===> A  ( A Ascending D Descending )              e
e                                                                    e
e Enter "/" to select option                                         e
e   Do not show this panel at start-up                              e
e                                                                    e
e Press ENTER to Continue or Press END(PF3) to Return                e
e                                                                    e
DsssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssM
```

You can specify names or masks for the various profile attributes to restrict the subsequent list to only those profiles whose attributes match your names and masks.

3. Press Enter to display the **Profile Display List** panel.

   If you specified names or masks on the **Fast Replication Engine Profile Service** panel, the **Profile Display List** panel shows only those profiles whose attributes match your filters.

Profile Display List field and command reference

This topic explains the primary and line commands available from the Profile Display List panel and describes the panel's fields.

The Profile Display List panel is shown in the following figure.

```
          IBM Tivoli Advanced Backup and Recovery for z/OS, V Row 1 to 1 of 19
                    Application Fast Replication Backup
 User ==> TWKALE          Profile Display List
 SysID==> TST
 Command ===>                                                  Scroll ===> PAGE

 Primary Command => CRE (Create Profile)  OTL (Offload Tape List)
                    REF (Refresh)
    Line Command => C   (Create)          D   (Delete)
                    R   (Rename)          S/V (Select/View)
                    U   (Update)

         Sort By => T    C (Creator) D (Description) N (Name) S (Share Option)
                         T (Type)
  Sort Direction => A    A (Ascending) D (Descending)

 I T S Profile Name                  Description                    Creator
   B * *                             *                              *
-------------------------------------------------------------------------------
   B U APFRRR-BACKUP-ONLY            OFFLOAD NOT PERMITTED          RR02
  F1=HELP      F2=SPLIT     F3=END       F4=RETURN    F5=IFIND     F6=RCHANGE
  F7=UP        F8=DOWN      F9=SWAP     F10=LEFT     F11=RIGHT    F12=RETRIEVE
```

Commands

You can use the primary commands to create profiles, refresh the Profile Display List, or to produce a Offload Tape List report. Use the line commands to create, rename, update, delete, and view individual application backup profiles.

Field descriptions

User
TSO ID of the user currently logged on.

SysID
The system ID of the current TSO/ISPF session.

I
The value specified in this column determines whether to display the initial panel. If you want to skip the initial panel, type / in this column.

T
The value shown in this column indicates the profile's type. Currently, only backup profiles appear in the list.

S
The value shown in this column indicates the type of sharing allowed for a profile:

- V indicates that other users can view the profile, but cannot modify it.
- N indicates that other users cannot view or update the profile.
- U indicates that other users can update the profile.

Profile Name
The profile's name.

Description
The description that was given to the profile when it was created or updated.

Creator
User ID under which the profile was initially created.

Creating an application backup profile

Use the information in this topic to create application backup and backup/offload profiles for use with the Fast Replication Services feature.

FRS uses the Fast Replication Engine (FRE) to make IBM FlashCopy backups using the FlashCopy function, a copy service available for IBM Enterprise Storage Systems (ESS) devices. FRS uses FRE to make EMC TimeFinder/Clone Volume Snap (SNAP) backups using the EMC SNAP VOLUME utility. FRS can maintain up to 99 generations of SNAP or FlashCopy backups. These target units will be exact copies of the source units with which they are paired.

Note: When using the SNAP backup method, FRE automatically varies all target units offline when the backup is taken. It is not necessary or recommended to vary the target volumes back online in preparation for the next backup.

FRS can make backups using DFSMSdss. You might want to use DFSMSdss profiles if your site does not have fast replication hardware, or if your site's hardware does not support Snap or FlashCopy but supports other fast replication services through DFSMSdss. For DFSMSdss backups, the target volumes must be online when the backup is taken. In addition, all target volumes must have their own unique volume serial.

Note: If fast replication is not used, the backup process may be lengthy depending on the number of data sets and tracks copied. If the storage array offers a fast replication utility, FRS can use the utility to create Fast Replication backups.

About offload options

FRS allows you to offload backups to tape or to another disk location. You can define various options for offloading, depending on the backup offload method selected. When you build a backup profile and set the **Perform Offload** field to Y, FRS offloads the backup according to your specifications when the offload job is run. You specify the retention period, expiration date, and how many generations of backup offloads you want to keep. FRS rolls off the oldest copy if necessary.

Creating an application backup profile without offload options

Use the information in this topic to create a new backup application profile without specifying any offload options.

Be aware that if you use this type of backup profile, an offload cannot be performed.

1. From the **Main Menu** panel, select the **Setup and Configuration** option and press Enter to display the **Setup and Configuration** menu panel.
2. From the **Setup and Configuration** menu, select the **Fast Replication Services** option and press Enter to display the **Fast Replication Engine Profile Service** panel.
3. On the **Fast Replication Engine Profile Service** panel, press Enter to display an unfiltered list of existing profiles on the **Profile Display List** panel.
4. On the **Profile Display List** panel, type CRE on the **Command** line and press Enter to display the **Enter New Profile Backup Options** panel.
5. On the **Enter New Profile Backup Options** panel, make the appropriate specifications:
   a) In the **Profile Name** field, specify a meaningful name for the new profile. Use as many as 30 characters.
   b) (Optional) In the **Description** field, specify enough information to help users differentiate this profile from others.
   c) Use the **Backup Method** field to specify which backup method you want used with this profile.
   d) Use the **Update Option** field to specify whether you want other users to be able to update this profile, have view-only access to it, or have no access to it.
   e) Press Enter to create the profile and display the **Update Backup Profile** panel.

   Note: When the Update Backup Profile panel appears, the **Creator**, **Name**, **Backup Method**, and **Current Generation** fields are read-only and cannot be changed from this panel.

6. Press the HELP function key and use the online help as a guide as you specify appropriate information for these fields:
   a) In the **Backup Generations** field, specify how many generations you want retained.

      Note: For each generation retained, there is one target volume in use for each source volume backed up.

   b) In the **Offload Options** field, type N.
   c) In the **Target Pool** field, specify whether you want to open the Target Pool Selection panel, where you can specify unit addresses for ranges of target volumes or a list of storage groups to use for target volumes.
   d) Use the HELP function key to display field description information for this panel.

Creating an application backup profile with offload options

If you need or want to offload backups to tape or another disk location, you specify backup destination, data set naming conventions, and other offload options as part of the backup profile.

Be aware that when you use this type of profile, an offload must be performed before a backup version is rolled off by another backup.

1. From the **Main Menu** panel, select the **Setup and Configuration** option and press Enter to display the **Setup and Configuration** menu panel.
2. From the **Setup and Configuration** menu, select the **Fast Replication Services** option and press Enter to display the **Fast Replication Engine Profile Service** panel.
3. On the **Fast Replication Engine Profile Service** panel, press Enter to display an unfiltered list of existing profiles on the **Profile Display List** panel.
4. On the **Profile Display List** panel, type CRE on the **Command** line and press Enter to display the **Enter New Profile Backup Options** panel.
5. On the **Enter New Profile Backup Options** panel, make the appropriate specifications:
   a) In the **Profile Name** field, specify a meaningful name for the new profile. Use as many as 30 characters.
   b) (Optional) In the **Description** field, specify enough information to help users differentiate this profile from others.
   c) Use the **Backup Method** field to specify which backup method you want used with this profile.
   d) Use the **Update Option** field to specify whether you want other users to be able to update this profile, access the profile in view-only mode, or not have access to the profile at all.
   e) Press Enter to create the profile and display the **Update Backup Profile** panel.

      Note: When the Update Backup Profile panel appears, the **Creator**, **Name**, **Backup Method**, and **Current Generation** fields are read-only and cannot be changed from this panel.

   f) Type Y in the **Offload Options** field and then press Enter to display the Offload Options panel.
6. Type Y in the **Local Primary**, **Local Backup**, **Recovery Site Primary**, or **Recover Site Backup** field to specify the backup types, then press Enter to refresh the panel to show the appropriate fields for the selected backup types.
7. Press the HELP function key and use the online help as a guide as you specify appropriate information for these fields:
   a) In the **Offload Generations** field, specify how many offload backups you want retained.
   b) In the **Delete Aged Backup Files** field, indicate whether you want the offload data sets for the rolled-off generation physically deleted.
   c) In the **Compress Data** field, specify whether you want the data mover utility to compress the data when the data is offloaded.
   d) Use the **Data Mover** field to specify which program you want FRS to use for creating the offload copy.
   e) In the **Encrypt Data** field, specify whether you want the data encrypted as it is offloaded.

f) Use the **Number of Tasks** field to specify the number of subtasks that FRS uses when it offloads the backups. Each subtask offloads a different target volume backup and creates all requested offload types (LP, LB, RP, RB) simultaneously. So if the profile indicates that four offloads (LP, LB, RP, RB) be created and one task is specified, four tape drives are required.

   Set this value to 1 for an FDR-type offload with encryption.

8. Press Enter to display the offload options pertaining to the data sets for the offloaded data:
   a) Type Y in the **Update DSN Specification** field if you want to set or change the data set specifications.
   b) In the **Unit Type** field, specify a valid UNIT where the data sets are to be written.
   c) If you want the offload data sets cataloged, type Y in the **Catalog** field.
   d) If your site uses SMS to manage data sets, type the SMS data class in the **Data Class** field.
   e) If your site uses SMS to manage data sets, type the SMS storage class in the **Storage Class** field.
   f) If your site uses SMS to manage data sets, type the SMS management class in the **Management Class** field.
   g) In the **Stack Backups on Tape** field, indicate whether you want backups stacked on a single tape whenever possible.
   h) If you specified Y in the **Stack Backups on Tape** field, use the **Tape Stack Limit** field to specify the maximum number of volumes you want stacked on one tape before dismounting the tape.
   i) If the unit type you specified in the **Unit Type** field is a tape device, use *either* the **Expiration Date** field to specify an expiration date (in YYYYDDD format) *or* the **Retention Period** field to specify the number of days to retain the tape.
   j) Use the **Maximum Tapes** field to specify the maximum number of tapes that can be used for offloaded data.
9. Press Enter.

   If you specified Y in the **Update DSN Specification** field, the **Offload DSN Specification** panel appears, providing options that determine how the offloaded data sets are to be named.
10. Use the HELP function key and the examples in this topic to specify a pattern for the data set names, using the variables you select. The variables are resolved at run time.
11. Press PF3 to save the profile.

Executing Fast Replication Services

Use the information in this topic to understand the syntax of the BKMFPROC control statements, build the Fast Replication Services backup, offload, and restore jobs using the profiles you have defined, and display a list of tapes onto which FRS backups have been offloaded, and to generate an Offload Tape List.

About the BKMFPROC statement

This topic provides syntax information for coding the BKMFPROC procedure's control statements: FRE BACKUP, FRE OFFLOAD, and FRE RESTORE.

The Fast Replication Services BKMFPROC member contains the JCL necessary to execute the BKMFRE50 program that interfaces Fast Replication Services with the Fast Replication Engine though an API to perform backups and offloads. The Fast Replication Engine performs the Fast Replication backup or the offload and provides a return code. The BKMFRE50 program then creates the backup or offload information to be written to the Critical Backup Tracking and Inventory database. The BKMSTSKA started task uses this information to record the backup activity in the Advanced Backup and Recovery for z/OS Inventory Data Set.

FRE BACKUP control statement

```
FRE BACKUP                                    -
CREATOR|OWNER|APPL(name)                      -
```

```
  PROFILE|PROF('profilename')                  -
  SELDSN('selection-dataset-name')             -
  BYPASS-OFFLOAD|NO-OFFLOAD                    -
  DEBUG                                        -
  SIMULATE                                     -
  TRACE                                        -
  WAIT-FOR-BACKGROUND-COPY
```

Required parameters

[CREATOR|OWNER|APPL](*name*)
Use this keyword to specify the TSO user ID under which the FRS backup profile was created.

[PROFILE|PROF]('*profilename*')
Use this keyword to specify the name of the FRS application backup profile you want used. Enclose your profile name in single quotes.

SELDSN(*selection-dataset-name*)
Use this keyword to specify the fully qualified name of the Selection Data Set that contains the names of the data sets to be backed up. The Selection Data Set can be in any format that is produced by Automated Critical Data Identification's APPLEND or that is acceptable as input to the Aggregate LoadBalancer.

Optional parameters

BYPASS-OFFLOAD|NO-OFFLOAD
When you are using a profile that includes the offload attribute, you can specify either form of this keyword to allow a roll off of the oldest backup without an offload, which permits numerous backups during an application cycle, where some, but not all, are to be offloaded.

DEBUG
Use this keyword to produce diagnostic data necessary to diagnose a problem. Do not use this keyword unless Technical Support directs you to do so.

SIMULATE
Specify this keyword to exercise the FRS backup function without creating a fast replication backup on the target volumes. The backup is recorded in the Inventory Data Set (IDS) as though a backup had been performed. The BKMEXPIRE process automatically removes all simulated backups from the IDS. When a simulated backup is referenced, the OFFLOAD and RESTORE commands automatically use the SIMULATE option.

TRACE
Use this keyword to produce diagnostic data necessary to identify a problem. Do not use this keyword unless Technical Support directs you to do so.

WAIT-FOR-BACKGROUND-COPY
Use this keyword to specify that you want the backup command to wait until the background copy is complete before returning control to the calling process, which permits delaying the application restart until the backup is complete.

FRE OFFLOAD control statement

```
FRE OFFLOAD                                    -
 CREATOR|OWNER|APPL(name)                      -
  PROFILE|PROF('profilename')                  -
  GEN(backup-relative-generation)              -
  DEBUG                                        -
  OFFLOAD-ALL                                  -
  RE-OFFLOAD                                   -
  SIMULATE                                     -
  TRACE
```

Required parameters

[CREATOR|OWNER|APPL](*name*)
Use this keyword to specify the TSO user ID under which the FRS backup-with-offload-options profile was created.

[PROFILE|PROF]('*profilename*')
Use this keyword to specify the name of the FRS application backup-with-offload-options profile you want used. Enclose your profile name in single quotes.

GEN(*backup-relative-generation*)
Use this keyword to specify which relative active generation of the backup you want offloaded. The backup profile indicates the maximum number of active generations; therefore, a profile with up to 2 generations would have a *backup-relative-generation* of either 0 or –1.

Optional parameters

DEBUG
Use this keyword to produce the messages necessary to diagnose a problem. Do not use this keyword unless Technical Support directs you to do so.

OFFLOAD-ALL|RE-OFFLOAD
Specify this keyword to restart a failed FRS offload. Any existing offload copies for the failed generation are deleted from the FRE repository and the offload is restarted from the beginning. To resume the offload without deleting the existing offloads, resubmit the OFFLOAD job without the OFFLOAD-ALL or RE-OFFLOAD parameter.

Note: The offload tapes for the failed offload are not deleted and expire according to the profile attributes.

SIMULATE
Specify this keyword to exercise the FRS offload function without creating a fast replication offload copy. The offload is recorded in the Inventory Data Set (IDS) as though an offload had been performed. The BKMEXPIRE process automatically removes all simulated offloads from the IDS.

TRACE
Use this keyword to produce the messages and captured data necessary to identify a problem. Do not use this keyword unless Technical Support directs you to do so.

FRE RESTORE control statement

```
FRE RESTORE                                -
 TOKEN(n)                                  -
  INCLDSN(                                 -
          dsn1                             -
          .                                -
          .                                -
          .                                -
          dsnn                             -
         )                                 -
 DEBUG                                     -
 FROM-OFFLOAD                              -
 LOCAL-SITE                                -
 RECOVERY-SITE                             -
 SIMULATE                                  -
 TRACE
```

Required parameters

TOKEN(*value*)
The value for this parameter is supplied automatically when you use the Critical Backup Tracking and Inventory process to construct a restore job. The value uniquely identifies the FRS backup version to be used for the restore operation.

INCLDSN(*dsnames*)
Use this parameter to identify the data sets you want restored from the specified FRS backup generation.

Optional parameters

DEBUG
Use this keyword to produce the messages necessary to identify a problem. Do not use this keyword unless Technical Support directs you to do so.

FROM-OFFLOAD
Use this keyword to specify that you want to restore using the offload copy, rather than the fast replication volume backup.

LOCAL-SITE
Use this keyword to specify that you are performing the restore at the local site where fast replication backup or local primary or local backup offload copies are used for the restores.

Note: If you specify the LOCAL-SITE keyword, you may not also specify the RECOVERY-SITE keyword; the two keywords are mutually exclusive.

RECOVERY-SITE
Use this keyword to specify that you are performing the restore at the recovery site where remote primary or remote backup offload copies are used for the restores.

Note: If you specify the RECOVERY-SITE keyword, you may not also specify the LOCAL-SITE keyword; the two keywords are mutually exclusive.

SIMULATE
Specify this keyword to exercise the FRS RESTORE function without restoring any data.

TRACE
Use this keyword to produce the messages and captured data necessary to diagnose a problem. Do not use this keyword unless Technical Support directs you to do so.

Building and submitting FRS backup jobs

After your FRS application backup profiles are defined, you can create backup jobs that use the BKMFPROC procedure in the product's JCL library. The process for building and submitting FRS restore jobs is different, so refer to that topic for more information about that process.

After the backup job JCL has been built, you should put the job into a scheduler to create regular backups.

The following figure shows an example of the JCL and control statements used to perform a Fast Replication backup and offload copy.

```
******************************** Top of Data ********************************
//APPLBKOF JOB (ACCT),'APPL BKUP/OFFLOAD',CLASS=A,MSGCLASS=X,
//         NOTIFY=&SYSUID,REGION=8M
//PROCLIB  JCLLIB ORDER=prefix.value.SBKMJCL
//*
//FRSBKUP  EXEC BKMFPROC,REGION=0M
//SYSIN    DD   *
FRE BACKUP  OWNER(RR02)                          -
            PROFILE('MYAPPL BACKUP WITH OFFLOAD')
            SELDSN(RR02.APFRRR.SELECT)
FRE OFFLOAD OWNER(RR02)                          -
            PROFILE('MYAPPL BACKUP WITH OFFLOAD')  -
            GEN(+0)
/*
******************************** Bottom of Data ********************************
```

Building and submitting FRS restore jobs

After using FRS to create a backup or offload, you can use the Critical Backup Tracking and Inventory Jobname List panel to begin building a restore job.

Note that each backup version has a unique token number that is required for performing a data set or application restore; therefore, the restore job must be constructed using the Critical Backup Tracking and Inventory process described in this topic.

1. From the **Main Menu** panel, select the **Backup and Recovery Management** option, and then press Enter to display the **Backup and Recovery Management** menu panel.
2. From the **Backup and Recovery Management** menu, select the **Jobs** option and press Enter to display the **Jobname List Search Setup** panel.

3. On the **Jobname List Search Setup** panel, you can press Enter to display an unfiltered list of job names, or you can specify names and masks and press Enter to display a list of only those jobnames whose attributes match your filters.
4. When the Jobname List panel appears, use the UP and DOWN function keys to scroll the list until you locate the jobname of the backup you want to use for the restore.
5. Specify which data sets you want restored:
   - To restore all data sets backed up by the job, type R in the text entry area to the left of the backup job name and press Enter, then press PF3 to pop up the Select Restore Option panel.
   - To restore specific data sets backed up by the job, type N in the text entry area to the left of the backup job name and press Enter. Type R in the text entry area to the left of the names of the data sets you want restored, and then press PF3 to pop up the Select Restore Option panel.
6. Determine whether you want to use Quick Restore or Setup Restore to generate the JCL, then make the appropriate specifications on the Select Restore Option panel. Use the HELP function key to display additional information about the available options on this panel. When you have finished your specifications, press Enter.
7. When the Select Restore Viewing Option panel appears, type S in the **Restore JCL** field and press Enter to display the built restore JCL.

   If the restore could cause data set conflicts, a confirmation panel lets you specify how you want to proceed. When you press Enter, the generated JCL appears.
8. Make the necessary modifications to the JCL and submit the job for execution.

   Note:

   - If the selected backup currently resides on a Fast Replication target volume, the data is restored using the Fast Replication restore process.
   - If the selected backup does not currently reside on a Fast Replication target volume and an offload was created, the data is restored using the offload restore process.
   - If the selected backup is a simulated backup, the SIMULATE option is added automatically to the FRE RESTORE command options.
   - The BKMEXPIRE process is used to maintain backup inventory currency.

The following figure shows an example of the JCL and control statements generated by Advanced Backup and Recovery for z/OS to restore application data sets from a Fast Replication backup or an offload copy.

```
******************************** Top of Data ********************************
//RR02RSTA JOB (ACCT),'FRS RESTORE',CLASS=A,MSGCLASS=X,NOTIFY=&SYSUID
//*
//* ******************************************
//* *   DFDSS RESTORE FROM LOGICAL COPY  TAKEN
//* *   ON 20110708 AT 08553514
//* *   BY JOB=APFRRRBS
//* ******************************************
//*
//FRSREST  EXEC PGM=BKMFRE50,REGION=0M,TIME=1440
//STEPLIB  DD   DISP=SHR,DSN=prefix.value.SBKMLOAD
//         DD   DISP=SHR,DSN=prefix.value.FRZ.SBKMLOAD
//FRZCNTFL DD   DISP=SHR,DSN=prefix.value.FRZ.CONTROL
//INI      DD   DISP=SHR,DSN=prefix.value.SBKMPARM(BKMINI)
//BRMIDS   DD   DISP=SHR,DSN=BKM.PROD.INVDB
//SYSPRINT DD   SYSOUT=*
//FREOUT   DD   SYSOUT=*
//SYSABEND DD   SYSOUT=*
//SYSIN DD *
 FRE RESTORE                                                          -
  TOKEN(26)                                                           -
  INCLDSN(                                                            -
          FRETST.APFRRR.APPLJ1.DC.G0001V00                            -
          FRETST.APFRRR.APPLJ1.DC.G0002V00                            -
          FRETST.APFRRR.APPLJ2.LISTC.G0002V00                         -
          FRETST.APFRRR.APPLJ2.LISTC.G0003V00                         -
          FRETST.APFRRR.APPLJ3.SORT.G0002V00                          -
          FRETST.APFRRR.APPLJ4.REPT.G0002V00                          -
         )
******************************** Bottom of Data ********************************
```

Creating an Offload Tape List

The information in this topic tells you how to use the Profile Display List panel to display a list of tapes onto which backups have been offloaded.

1. From the **Main Menu** panel, select the **Setup and Configuration** option and then press Enter to display the **Setup and Configuration** menu panel.
2. From the **Setup and Configuration** menu, select the **Perform Fast Replication Services** option, and then press Enter to display the **Fast Replication Engine Profile Services** panel.
3. On the **Fast Replication Engine Profile Services** panel, press Enter to display an unfiltered list of FRS profiles on the **Profile Display List** panel.
4. On the **Profile Display List** panel, type 0TL in the **Command** line and press Enter to display the **FRS Offload Tape List** panel.
5. On the **FRS Offload Tape List** panel, you can specify names or masks for the various fields and press Enter to display a filtered list of tapes whose attributes match your filters, or you can press Enter to display an unfiltered list of tapes that contain offloaded backups.
6. To sort the list, type the appropriate sort value in the **Sort By** and **Sort Direction** fields, then press Enter to refresh the display.

Chapter 25. Generating backup reports

Use the Main Menu **Reports** option to submit batch report requests or to create and view reports online. Critical Backup Tracking and Inventory, Automated Critical Data Identification, and ABARS Manager reports are generated using backup data only.

Generating Automated Critical Data Identification reports

This information describes the reports that are available through Automated Critical Data Identification.

These reports are generated through the Automated Critical Data Identification batch facility. The JCL for these reports is located in the product JCL library.

Generating a data set name report

The data set name report (BKMDSRS) provides data set name and inclusion and exclusion information from the last successful cycle.

When Automated Critical Data Identification collects JCL and RSP SMF data, it stores the data in Data Set Detail Records (DSD). When you run BKMAPLEN, these records are analyzed and Data Set Records (DSR) are created.

Note: New DSR records are created each time you run BKMAPLEN.

You can locate DSR records by running the BKMDSRS report or by selecting the **Display Current JCL/SMF DSN Entries** option from the **APPLICATION Display Selections** panel.

1. Copy member BKMDSRS from the product JCL library.
2. Modify the symbolic parameter, BKMDSRS, to specify an output data set name of your choice where the report will be written.

   BKMDSRS is a data set that can be browsed using ISPF.
3. Specify PARM=*applname* where *applname* is an application name.

   Note: If you do not specify the application name, the report contains all data set records for all applications.

Data set name report description

The data set name report provides data set name and inclusion and exclusion information.

Fields

DSNAME
The data set name.

APPLNAME
The application name.

ACDI STATUS
The Automated Critical Data Identification status.

EVALUATION STATUS
The reason for the ACDI status.

FLTR SCOPE
The scope of the filter: Local, Global, or Universal.

FLTR TYPE
The type of filter.

FILTER
Filter specification.

LAST UPDATE
The date and time stamp of the last update.

JCL
The disposition from the JCL record.

SMF
The disposition from the SMF record.

SMF ACCS
The type of data set access from SMF records.

JOB NAME
The name of the job.

SMF JOBN
The job name from the SMF records.

SMF DATE
The SMF date.

SYSID
System ID.

UNIT
Unit name.

VOLSER
Volume serial number.

Generating an overlap report

The BKMOVLP report shows data set overlaps within the Automated Critical Data Identification Application Data Inventory database.

The overlap report uses data set records (DSRs) for input. Data set records are records that have already been analyzed and created by BKMAPLEN or Verification. The currently running cycle creates data set detail (DSD) records. Running cycles are not contained in the overlap report.

Note: Run BKMJRFSH any time you change job names or define new applications.

1. Copy member BKMOVLP from the product JCL library.
2. Modify the symbolic parameter BKMOVLP to specify the output data set where the report will be written.

BKMOVLP creates a data set that can be browsed through ISPF.

Overlap report description

The overlap report shows data set overlaps within the Automated Critical Data Identification Application Data Inventory database.

Fields

DSNAME
The data set name.

APPLNAME
The application name.

ACDI STATUS
The Automated Critical Data Identification status.

EVALUATION STATUS
The reason for the ACDI status.

FLTR SCOPE
The scope of the filter: Local, Global, or Universal.

FLTR TYPE
The type of filter.

FILTER
The filter specification.

LAST UPDATE
The date and time stamp of the last update.

JCL
The disposition from the JCL record.

SMF
The disposition from the SMF record.

SMF ACCS
The type of data set access from the SMF records.

JOB NAME
The name of the job.

SMF JOBN
The job name from SMF records.

SMF DATE
The SMF date.

SYSID
System ID.

UNIT
Unit name.

VOLSER
Volume serial number.

Generating a filter report

The Automated Critical Data Identification filter report lists the filters for all applications in the Automated Critical Data Identification Application Data Inventory database.

1. Copy member BKMFLTRP from the product JCL library.
2. Specify an output data set in the SYSPRINT where the report will be written.

BKMFLTRP creates a data set that can be browsed using ISPF.

Filter report description

The filter report lists the filters for all applications in the Automated Critical Data Identification Application Data Inventory database.

Fields

RETENTION FILTERS
This section of the report lists the Retention filters found in the Automated Critical Data Identification Application Data Inventory database for the application.

EVALUATION FILTERS
This section of the report lists the Evaluation filters found in the Automated Critical Data Identification Application Data Inventory database for the application.

EXTERNAL FILTERS
This section of the report lists the External filters found in the Automated Critical Data Identification Application Data Inventory database for the application.

Generating a DSD record count report

The BKMCHKDB utility reports the number of Data Set Detail (DSD) records in each application. To generate a DSD record count report, complete these steps.

Note: BKMCHKDB does not report on records captured using SMF Scan.

1. Copy member BKMCHKDB from the product JCL library.
2. Modify the SYSIN statement for the appropriate applications. If you do not specify an application, BKMCHKDB generates a list of all applications found in the Automated Critical Data Identification Application Data Inventory database.

DSD record count report description

The DSD record count report shows the number of Data Set Detail (DSD) records in each application.

Fields

CURRENT CYCLE
The number of DSD records reported for the current cycle for the application.

EARLIEST
The earliest date reported.

LATEST
The latest date reported.

PRIOR CYCLES
The number of DSD records reported for prior cycles for the application.

Status descriptions

This information describes status (including evaluation status fields) for the data set name and overlap reports.

ADD NEW
Data set added by user (can be matched with any status).

INC
Include (critical). Matched with the status of INC CRITICAL or GDG BASE NAME.

INC ALWAYS
Data set included (critical) regardless of normal evaluation process. GDG data sets and filtered data sets can be included in this category.

INC ACCOM
Critical tape data set (determined by the user whether the tape data sets are categorized as INCLUDES or ACCOMPANY).

INC-ADD NEW
Data set added by user (can be matched with any include, critical, status).

INC ALLOC
Critical data sets that must be present for recovery, but are not backed up.

INC CRITICAL
Critical data set to be backed up.

INC CRITICAL INC - Add New
Setup for rerun recovery (only if **FR if new GDS** was set to Y).

INC - INC-FORCE
Included regardless of the normal evaluation process (GDG data set processing or match on INCLUDE filter). Matched with INC ALWAYS.

EXC
Excluded. Matched with EXC NON-CRIT.

EXC ALWAYS
Data set excluded by other than normal evaluation process. This status could result from:

- A filter match
- The data set is uncataloged
- Relative GDGs when absolute are requested (or the other way around)

EXC-FILTERED
Match on EXCLUDE filter. Matched with status of EXC ALWAYS.

EXC-FORCE
Excluded regardless of normal evaluation process (GDG data set processing). Matched with EXC ALWAYS.

EXC NON-CRIT
Non-critical data set. It is not critical from normal evaluation. For rerun recovery, these would be output files.

EXC-UNCATLG
Excluded (not cataloged). Matched with EXC NON-CRIT.

GDG BASE NAME
GDG BASE data set must be present for recovery.

UNKNOWN - NOT EVALUATED
This status could result from records that have been built but not yet evaluated.

Generating Critical Backup Tracking and Inventory backup reports

The **Main Menu Reports** option enables you to submit batch report requests, or create and view reports online. Reports are generated using backup data only.

1. From the **Main Menu**, select the **Reports** option.
   The **Reports** panel is displayed.
2. In the **Critical Backup Tracking Inventory (CBTI)** area, select the **Backup Reports** option.
   The **Critical Backup Tracking and Inventory Backup Reports Menu** appears.
3. Select the number corresponding to the report you want to generate and press Enter.
4. Specify the appropriate report options.
5. Type s to select one of these options:

Option	Description
Build/Submit JCL	Build the JCL from the options supplied on the panel and submit.
Build/View JCL	Build the JCL from the options supplied on the panel, and then view the JCL.
Exec/View Online	Generate the report with the options specified and view the report online.

Overlap by Data Set report

The Overlap by Data Set report displays data set/Unix files overlaps by event, date, time, and type.

The following figure shows an example of an Overlap by Data Set report.

```
 SEARCH CRITERIA:
         DATASET => /u/ts3435/~_for_test|u-ts3435*.**
         DATASET => TS3435.TEMP.ARCINBAK.TEST5.**

PAGE 1                                    CBTI - OVERLAP BY DATASET             2023/07/14 00:00:36

DATASET NAME                                        EVENT    DATE       TIME     TYPE
--------------------------------------------------- -------- ---------- -------- --------
/u/ts3435/~_for_test|u-ts3435-~aracters|001         TS3435ZL 2023/03/02 04:41:54 DSS ZFS
                                                    TS3435ZF 2023/03/02 02:44:42 DSS ZFS
                                                    TS3435ZI 2023/03/01 22:50:32 DSS ZFS
                                                    TS3435ZJ 2023/03/01 05:42:45 DSS ZFS
                                                    TS3435ZI 2023/03/01 05:28:53 DSS ZFS

TS3435.TEMP.ARCINBAK.TEST5                          TS3435II 2023/05/31 04:02:06 IEBGENER
                                                    TS3435BR 2022/08/28 21:28:44 IEBGENER

----------------------------
TOTAL DATASETS FOUND:       2
```

Report options

Dataset Name(s) or Mask(s)
(Required) Specify the data set names/Unix file indexes or masks to use when searching for overlaps. You can specify up to five data set names/Unix file indexes or masks with a maximum of 44 characters each.

Include HSM Incrementals
Specify Y to include the HSM incrementals in the search for data set overlaps, otherwise specify N.

Overlap by Backup Event report

The Overlap by Backup Event report displays overlaps by event, date, time, and type.

The following figure shows an example of an Overlap by Backup Event report.

```
 SEARCH CRITERIA:
           EVENT => DS*

 PAGE 1                 CBTI -  OVERLAP BY BACKUP EVENT
 DATASET NAME                        EVENT    DATE       TIME    TYPE
 ----------------------------------- -------- ---------- ------  ------
 BACKUP.BKM.JUL0805.DTS001           DSM      2006/03/30 10:20:05 ABARS
                                               DS7      2005/10/31 07:37:26 ABARS

 BACKUP.BKM.NOBKP.JUL1105.DTS001               P390BKPE 2007/07/08 19:59:33 DSS FVD
                                               DSM      2006/03/30 10:20:05 ABARS
                                               DS7      2005/10/31 07:37:26
ABARS
 BACKUP.CART0A.FILE1                           DSM      2006/03/30 10:20:05 ABARS
                                               DS7      2005/10/31 07:37:26
ABARS
```

Report options

Event Name(s)
(Required) Specify a maximum of five event names or masks. An event name can be a jobname or ABARS aggregate name.

Overlap by Job Groups report

The Overlap by Job Groups report display job groups that are tracking specific jobs.

The following figure shows an example of an Overlap by Job Groups report.

```
SEARCH CRITERIA:
         JOBNAMES => *

PAGE 1                                    CBTI     - OVERLAP BY JOB GROUPS

JOBNAME    RC  RCB DATE       TIME     JOB GROUPS
-------- ---- ---- ---------- -------- ------------------------------------
ADDB2001    0    0 2006/02/01 09:30:01 EDSGRP   PAYDLY

ADSCD001    0    0 2007/01/10 09:30:01 EDSGRP   PAYDLY

BKMBJ110    0    0 2006/12/14 07:51:47 EDSGRP   PAYDLY

         ------------------------------------------------------
         JOBS FOUND:      3
```

Report options

Job Name(s)
(Required) Specify up to five job names (a maximum of eight characters each). The job names you specify will be compared to each of the defined job groups to determine if multiple job groups are tracking the specified jobs.

Data Set List report

This information describes the Data Set List report.

The following figure shows an example of a Data Set List report.

```
SEARCH CRITERIA:
         DATASET => /u/ts3435/~_for_test|u-ts3435-~aracters*
         DATASET => TS3435.TEMP.ARCINBAK.TEST5
            TYPE => ACDFGHIMKORSTUVWZ

PAGE 1                                  CBTI  -  DATASET LIST                                  2023/07/14 00:20:04

DATASET NAME                                        EVENT    DATE       TIME     TYPE
--------------------------------------------------- -------- ---------- -------- ---------
/u/ts3435/~_for_test|u-ts3435-~aracters|001         TS3435ZL 2023/03/02 04:41:54 DSS ZFS
                                                    TS3435ZL 2023/03/02 04:38:36 DSS ZFS
                                                    TS3435ZL 2023/03/02 04:37:05 DSS ZFS
                                                    TS3435ZL 2023/03/02 04:24:34 DSS ZFS

TS3435.TEMP.ARCINBAK.TEST5                          TS3435II 2023/05/31 04:02:06 IEBGENER
                                                    TS3435II 2023/05/26 01:26:34 IEBGENER
                                                    TS3435II 2023/04/11 22:20:46 IEBGENER
                                                    TS3435II 2023/04/11 22:18:43 IEBGENER
```

Report options

Dataset Name(s) or Mask(s)
(Required) Specify data set names/Unix file indexes or masks to include in the report. You can specify up to five data set names/Unix file indexes or masks with a maximum of 44 characters each.

The following is the format for providing Unix file index:

```
<first 10 characters of Directory Name>~<Last 10 Characters of Directory
Name>|<first 10 characters of Unix File Name>~<Last 10 Characters of Unix
File Name>
```

Advanced options

Job Name(s)
Filter the report output by job name or mask. You can specify up to five job names or masks with a maximum of eight characters each.

Exclude RC
Specify RC to exclude event Return code >= Entered Value

For example,

- 4 - show only events with RC < 4
- 8 - show only events with RC < 8

Note: The HSM backup events will not be displayed if you opt for this field.

From Date

The starting date or mask of the data sets or events to search for. The date must be in format YYYY/MM/DD or MM/DD/YYYY. Relative date masking is permitted. For example, you can specify an asterisk (*) to indicate the current date, you could specify *-5 to indicate five days before the current date (up to *-999). Valid values are:

- * — today only.
- *+1 — today and yesterday.
- *-1 — yesterday only.
- *+*nnn* — today and up to 999 days ago.
- *-*nnn* — *nnn* days ago only.

In the following examples *n* is an integer 1-999. Specifying date criteria in the format *-*n* only displays events equal to *n* relative days from the current date:

Type *-4 to display events that occurred four days prior to today.

Type *+4 to display all events, from today through four days prior to today.

You can also specify a specific date in the format of YYYY/MM/DD or MM/DD/YYYY. In this case, only events matching the specific date are displayed.

To Date

The ending date or mask of the data sets or events to search for. The date must be in format YYYY/MM/DD or MM/DD/YYYY. Relative date masking is permitted. For example, you can specify an asterisk (*) to indicate the current date, or you could specify *-5 to indicate five days before the current date (up to *-999). Valid values are:

- * — today only.
- *+1 — today and yesterday.
- *-1 — yesterday only.
- *+*nnn* — today and up to 999 days ago.
- *-*nnn* — *nnn* days ago only.

In the following examples *n* is an integer 1-999. Specifying date criteria in the format *-*n* only displays events equal to *n* relative days from the current date:

Type *-4 to display events that occurred four days prior to today.

Type *+4 to display all events, from today through four days prior to today.

You can also specify a specific date in the format of YYYY/MM/DD or MM/DD/YYYY. In this case, only events matching the specific date are displayed.

Type(s)

Specify one, or any combination of the following types:

Type	Description
A	Data sets backed up by ABARS.
C	Data sets backed up by ACM or CR+.
D	Data sets backed up by DSS.
F	Data sets backed up by FDR.
G	Data sets backed up by IEBCOPY, IEBGENER, ICEGENER, or SYNCGENR.
H	Data sets backed up by HSM Incremental, ARCINBAK, or AUTODUMP.
M	Data sets backed up by CFCAMS.

Type	Description
K	Data sets backed up by ADARUN.
O	Data sets backed up by ICETOOL.
R	Data sets backed up by Fast Replication Services.
S	Data sets backed up by ICEMAN, SORT, or SYNCSORT.
T	Data sets backed up by TAPECOPY.
U	Data sets backed up by User Defined Backup programs.
V	Data sets backed up by CA-FAVER.
W	Data sets backed up by CA-Disk.
Z	Data sets backed up by VSAM Assist.

If a user has chosen option **H** to view HSM Backup Report and if the DSNLIST_INCLUDE_HSM_FULL = N in SBKMPARM member BKMINI, the report will show maximum two HSM backups per day. The following is a sample report:

```
SEARCH CRITERIA:
          DATASET => TS3435.TEMP.ARCINBAK.TEST5
             TYPE => H

PAGE 1                                    CBTI  -  DATASET LIST                                  2023/07/18 00:29:58

DATASET NAME                                       EVENT    DATE       TIME     TYPE
-------------------------------------------------- -------- ---------- -------- --------
TS3435.TEMP.ARCINBAK.TEST5                         HSM      2023/07/18 00:47:42 HSM INCR
                                                   HSM      2023/07/18 00:47:37 HSM INCR
                                                   HSM      2023/01/17 21:40:58 HSM INCR


------------------------------
TOTAL DATASETS FOUND:       1
```

If DSNLIST_INCLUDE_HSM_FULL = Y in SBKMPARM member BKMINI, the report will show all HSM backups for the dataset. The following is a sample report:

```
SEARCH CRITERIA:
          DATASET => TS3435.TEMP.ARCINBAK.TEST5
             TYPE => H

PAGE 1                                    CBTI  -  DATASET LIST                                  2023/07/18 00:29:58

DATASET NAME                                       EVENT    DATE       TIME     TYPE
-------------------------------------------------- -------- ---------- -------- --------
TS3435.TEMP.ARCINBAK.TEST5                         HSM      2023/07/18 00:47:42 HSM INCR
                                                   HSM      2023/07/18 00:47:37 HSM INCR
                                                   HSM      2023/07/18 00:47:32 HSM INCR
                                                   HSM      2023/01/17 21:40:58 HSM INCR

------------------------------
TOTAL DATASETS FOUND:       1
```

Data Set List by Jobname with Outputs report

The Data Set List by Jobname with Outputs report displays data sets listed by job name.

The following figure shows an example of a Data Set List by Jobname with Outputs report.

```
SEARCH CRITERIA:
         JOBNAMES => AD*

PAGE 1                   CBTI - DATASET LIST BY JOBNAME WITH OUTPUTS

JOBNAME: ADDB2001 PROC:          STEP: STEP0001 TASK: 0001
                  DATE: 2006/02/01 TIME: 09:30:01.54
DATASET NAME LIST
-------------------------
DB2.DB8G.ARCHLOG1.A0000241
DB2.DB8G.ARCHLOG1.A0000242
DB2.DB8G.ARCHLOG1.A0000243
DB2.DB8G.ARCHLOG1.A0000244
DB2.DB8G.ARCHLOG1.A0000245
DB2.DB8G.ARCHLOG1.A0000246
DB2.DB8G.ARCHLOG1.A0000357
```

Report options

Job Name(s)
Specify a maximum of five job names or masks (maximum of eight characters each).

Full Volume Dump List report

The Full Volume Dump List report displays information on specific volumes associated with full volume dumps.

The following figure shows an example of a Full Volume Dump List report.

```
SEARCH CRITERIA:
          VOLUME => BK*

PAGE 1                                        CBTI     - FULL VOLUME DUMP LISTV
VOLUME   PGM/TYPE RC  RCB DATE       TIME     PROCSTEP STEPNAME TASK# # DSNS
-------- -------- --- --- ---------- -------- -------- -------- ----- ------
BKP001   DSS FVD    0   0 2007/07/07 22:00:03          S3        0001    190
         DSS FVD    0   0 2007/06/23 22:00:06          S3        0001    205
         DSS FVD    0   0 2007/06/09 22:00:04          S3        0001    193
         DSS FVD    0   0 2007/06/02 22:00:04          S3        0001    191

BKP002   DSS FVD    0   0 2007/07/07 22:00:03          S3        0002    188
         DSS FVD    0   0 2007/06/23 22:00:06          S3        0002    184
         DSS FVD    0   0 2007/06/09 22:00:04          S3        0002    190
         DSS FVD    0   0 2007/06/02 22:00:04          S3        0002    194
```

Report options

Volume Name(s)
(Required.) Specify a maximum of five volume names or masks to search for (maximum of 6 characters each).

Advanced options

From Date
The starting date or mask of the volumes or events to search for. The date must be in format YYYY/MM/DD or MM/DD/YYYY. Relative date masking is permitted. For example, you can specify an asterisk (*) to indicate the current date, you can specify `*-5` to indicate five days before the current date (up to *-999). Valid values are:

- * — today only.
- *+1 — today and yesterday.
- *-1 — yesterday only.
- *+*nnn* — today and up to 999 days ago.
- *-*nnn* — *nnn* days ago only.

In the following examples *n* is an integer 1-999. Specifying date criteria in the format *-*n* only displays events equal to *n* relative days from the current date:

Type `*-4` to display events that occurred four days prior to today.

Type `*+4` to display all events, from today through four days prior to today.

To Date
The ending date or mask of the volumes or events to search for. The date must be in format YYYY/MM/DD or MM/DD/YYYY. Relative date masking is permitted. For example, you can specify an asterisk (*) to indicate the current date, or you can specify `*-5` to indicate five days before the current date (up to *-999). Valid values are:

- * — today only.
- *+1 — today and yesterday.

- *-1 — yesterday only.
- *+*nnn* — today and up to 999 days ago.
- *-*nnn* — *nnn* days ago only.

In the following examples *n* is an integer 1-999. Specifying date criteria in the format *-*n* only displays events equal to *n* relative days from the current date:

Type `*-4` to display events that occurred four days prior to today.

Type `*+4` to display all events, from today through four days prior to today.

Jobname List report

The Jobname List report displays a job list according to the report criteria you specify.

The following figures show examples of a Jobname List reports.

If `JOBNAME_LIST_INCLUDE_DSNE = N` in SBKMPARM member BKMINI, the report looks like this:

```
SEARCH CRITERIA:
          JOBNAME => BKPDSS*
          JOBNAME => BKPCKM*
          JOBNAME => CFVBKP*

PAGE 1                                        CBTI     - JOBNAME LIST                           2022/11/03 07:30:12

JOBNAME  PGM/TYPE RC  RCB   GEN DATE       TIME     END DATE   END TIME PROCSTEP STEPNAME TASK# # DSNS
-------- -------- --- --- ----- ---------- -------- ---------- -------- -------- -------- ----- ------
BKPCKM01 BCS CR+  000 000     0 2022/09/05 10:56:08 2022/09/05 10:56:09          CATBKP00  0001      1

BKPCKM02 BCS CR+  008 000     0 2022/09/05 10:56:08 2022/09/05 10:56:09          CATBKP00  0001      0

BKPCKM03 VVDS CR+ 004 000     0 2021/07/19 02:37:54 2021/07/19 02:37:55          CATBKP00  0002      1
         DSN CR+  004 000     0 2021/07/19 02:37:54 2021/07/19 02:37:55          CATBKP00  0001      1
         DSN CR+  000 000    -1 2021/06/14 01:12:25 2021/06/14 01:12:25          CATBKP00  0001      1

BKPCKM04 VVDS CR+ 004 000     0 2022/09/05 11:07:34 2022/09/05 11:07:34          CATBKP00  0001      1

BKPCKM05 DSN CR+  008 000     0 2022/09/05 10:56:10 2022/09/05 10:56:11          CATBKP00  0001      0

BKPCKM06 DSN CR+  008 000     0 2022/09/05 10:56:10 2022/09/05 10:56:11          CATBKP00  0001      0

BKPCKM07 DSN CR+  008 000     0 2022/09/05 10:56:12 2022/09/05 10:56:12          CATBKP00  0001      0

BKPCKM33 VVDS CR+ 004 000     0 2021/07/19 04:05:22 2021/07/19 04:05:22          CATBKP00  0002      1
         DSN CR+  004 000     0 2021/07/19 04:05:22 2021/07/19 04:05:22          CATBKP00  0001      1

BKPDSS01 DSS LD   000 000     0 2022/08/25 10:18:54 2022/08/25 10:18:54          BKPDSS01  0001      2
         DSS LD   000 000    -1 2021/11/19 01:28:55 2021/11/19 01:28:56          BKPDSS01  0001      1
```

If `JOBNAME_LIST_INCLUDE_DSNE = Y` in SBKMPARM member BKMINI, the report looks like this:

```
SEARCH CRITERIA:
          JOBNAME => BKPDSS*
          JOBNAME => BKPCKM*
          JOBNAME => CFVBKP*

PAGE 1                                        CBTI     - JOBNAME LIST                           2022/11/03 07:27:36

JOBNAME  PGM/TYPE RC  RCB   GEN DATE       TIME     END DATE   END TIME PROCSTEP STEPNAME TASK# # DSNS #DSNE
-------- -------- --- --- ----- ---------- -------- ---------- -------- -------- -------- ----- ------ -----
BKPCKM01 BCS CR+  000 000     0 2022/09/05 10:56:08 2022/09/05 10:56:09          CATBKP00  0001      1     0

BKPCKM02 BCS CR+  008 000     0 2022/09/05 10:56:08 2022/09/05 10:56:09          CATBKP00  0001      0     1

BKPCKM03 VVDS CR+ 004 000     0 2021/07/19 02:37:54 2021/07/19 02:37:55          CATBKP00  0002      1     0
         DSN CR+  004 000     0 2021/07/19 02:37:54 2021/07/19 02:37:55          CATBKP00  0001      1     0
         DSN CR+  000 000    -1 2021/06/14 01:12:25 2021/06/14 01:12:25          CATBKP00  0001      1     0

BKPCKM04 VVDS CR+ 004 000     0 2022/09/05 11:07:34 2022/09/05 11:07:34          CATBKP00  0001      1     1

BKPCKM05 DSN CR+  008 000     0 2022/09/05 10:56:10 2022/09/05 10:56:11          CATBKP00  0001      0     1

BKPCKM06 DSN CR+  008 000     0 2022/09/05 10:56:10 2022/09/05 10:56:11          CATBKP00  0001      0     1

BKPCKM07 DSN CR+  008 000     0 2022/09/05 10:56:12 2022/09/05 10:56:12          CATBKP00  0001      0     1

BKPCKM33 VVDS CR+ 004 000     0 2021/07/19 04:05:22 2021/07/19 04:05:22          CATBKP00  0002      1     0
         DSN CR+  004 000     0 2021/07/19 04:05:22 2021/07/19 04:05:22          CATBKP00  0001      1     0

BKPDSS01 DSS LD   000 000     0 2022/08/25 10:18:54 2022/08/25 10:18:54          BKPDSS01  0001      2     0
         DSS LD   000 000    -1 2021/11/19 01:28:55 2021/11/19 01:28:56          BKPDSS01  0001      1     0
```

Report options

Choose one of these options:

Job Name(s)
Specify up to five job names (a maximum of 8 characters each). You cannot specify a job name in conjunction with a job group.

Job Group
Specify a job group. Job group masks are not permitted. You cannot specify a job group in conjunction with a job name or mask.

Note: You must specify a job name or group.

Advanced options

Use Date Type
Specify whether to use Jobname Start Date/Time or End Date/Time when a **From Date** or **To Date** is specified. Specify the date to use: starting date (S) or ending date (E).

From Date
From Date specifies the starting date or mask of the job names to search for. The date must be in format YYYY/MM/DD or MM/DD/YYYY. Relative date masking is permitted. For example, you can specify an asterisk (*) to indicate the current date, you can specify `*-5` to indicate five days before the current date (up to *-999).

From Time
Specify a time (0000 - 2359) to narrow the search of the Jobname List report. If no **To Date** is specified then only job names matching the date and time specified will be displayed.

To Date
To Date specifies the ending date or mask of the job names to search for. The date must be in format YYYY/MM/DD or MM/DD/YYYY. Relative date masking is permitted. For example, you can specify an asterisk (*) to indicate the current date, or you can specify `*-5` to indicate five days before the current date (up to *-999).

To Time
Specify a time (0000 - 2359) to narrow the search of the Jobname List report. If you do not specify a **From Date**, only those job names matching the date and time specified will be displayed.

GEN *nn*
Specify a job name generation number to report only those job names matching the generation specified.

Jobnames Not Tracked by a Job Group report

The Jobnames Not Tracked by a Job Group report lists jobs that are not tracked by a job group.

The following figure shows an example of a Jobnames Not Tracked by a Job Group report.

```
 SEARCH CRITERIA:
          JOBNAMES => BKM*

 PAGE 1                            CBTI     - JOBNAMES NOT TRACKED BY A JOB
GROUP
 JOBNAME    RC  RCB DATE       TIME
 -------- ---- ---- ---------- --------
 BKMBJA20    0    0 2006/04/25 16:10:17

 BKMBJ012    0    0 2006/04/14 11:28:59

 BKMBJ031    0    4 2006/06/30 12:58:29

 BKMBJ11B    0    0 2006/05/23 15:14:18
```

Report options

Job Name(s)
(Required) Specify up to five job names with a maximum of eight characters each.

What Isn't Backed Up report

The What Isn't Backed Up report identifies what is not backed up at your site to help ensure you are not missing critical data sets. By tracking and providing an inventory of backups, you can easily find data sets that have not been backed up by comparing them to the Backup & Recovery Manager Suite IDS, DFSMShsm BCDS, or both.

The following figure shows an example of a What Isn't Backed Up report.

```
******************** TOP OF DATA **************
 ***************************************************
 *                WHAT ISN'T BACKED UP               *
 *                                                   *
 *  NO BACKUP FOUND FOR THE FOLLOWING:               *
 *  NOTE: "+" = MULTI-VOLUME DATASET                 *
 ***************************************************
   DATASET NAME                                    VOLSER
 -------------------------------------------------------
   DSDB.ARCHLOG1.A0000002                          WORK03
   DSDB.ARCHLOG1.A0000004                          FEND24
   DSDB.ARCHLOG1.B0000001                          FENC03
   DSDB.ARCHLOG1.B0000004                          FENM05
   DSDB.BSDS01                                     ??????
   DSDB.BSDS01.DATA                                LG1005
   DSDB.BSDS01.INDEX                               LG1005
   DSDB.BSDS02                                     ??????
   DSDB.BSDS02.DATA                                LG2008
   DSDB.BSDS02.INDEX                               LG2008
   DSDB.DSNDBC.AASNS001.SSTRINF0.I0001.A001        MIGRAT
   DSDB.DSNDBC.AASNS001.XSTRINF0.I0001.A001        MIGRAT
   DSDB.DSNDBC.AASNS001.XSTRINF1.I0001.A001        MIGRAT
   DSDB.DSNDBC.AASNS001.XSTRINF2.I0001.A001        MIGRAT
   DSDB.DSNDBC.AASNS001.XSTRINF5.I0001.A001        MIGRAT
```

Note: Your installation is responsible for the maintenance of the ICF catalogs. If data set entries exist in the catalog that are either no longer on disk, or are cataloged to non existent volumes, it is considered a catalog maintenance issue. Similarly, if there are disk entries with no associated catalog entry, it is considered a DASD maintenance issue. The What Isn't Backed Up Report identifies these entries, but it does not eliminate them.

Report options

Target IDS
Specify Y to compare the source to the Advanced Backup and Recovery for z/OS Inventory Data Set, otherwise specify N.

Target BCDS
Specify Y to compare the source to the DFSMShsm BCDS, otherwise specify N.

Sort By
Specify N to sort the report by data set name. Specify V to sort the report by volume.

Exclude Empty Datasets
Specify Y to if you want empty data sets omitted from the report. Specify N if you want the empty data sets included in the report.

Include Empty Datasets only
Specify Y to if you want the report to include empty data sets only. Otherwise, specify N.

Exclude Tape
Specify Y to if you want tape data sets omitted from the report. Otherwise, specify N.

Verify Disk
Specify Y to if you verification that there is a VTOC entry for each data set included in the report. Otherwise, specify N.

Exclude date
Specify date or mask to exclude in the report.

Source Parameters

Each parameter has a maximum of 44 characters. To specify additional parameters, select the **Build/ View JCL** option.

Include DSN or Mask
Specify a data set name or mask to include in the compare.

Exclude DSN or Mask
Specify a data set name or mask to exclude in the compare.

Include UCAT or Mask
Specify a user catalog name or mask to include in the compare.

Exclude UCAT or Mask
Specify a user catalog name or mask to exclude in the compare.

Include Vol or Mask
Specify a volume name or mask to include in the compare.

Exclude Vol or Mask
Specify a volume name or mask to exclude in the compare.

Replcode or Mask
Specify a outcode name or mask in the report

Critical in Automated Critical Data Identification with No Backup Record in Critical Backup Tracking and Inventory report

This information describes the Critical in Automated Critical Data Identification with No Backup Record in Critical Backup Tracking and Inventory report.

The following figure shows an example of a Critical in Automated Critical Data Identification with No Backup Record in Critical Backup Tracking and Inventory report.

```
SEARCH CRITERIA:
           DATASET => **
  ACDI APPLICATION => TEST

 PAGE 1                CBTI - CRITICAL WITH NO CBTI BACKUP RECORD          2008/01/29 09:14:34

 DSN(S) WITH NO BACKUP FOR ACDI APPL: TEST         VOLSER   MGMTCLAS STORCLAS STORGRP
 ------------------------------------------------ ------   -------- -------- --------
 BKMI.BKM.IVP.ESDS                                 BKMD01   MCBKMSD  SCBKMSD  SGBKMSD

 BKMI.BKM.IVP.JA30.NOBKP.ESDS                      NOBKP1   MCNOBK   SCNOBK   SGBKMSD

 BKMI.BKM.IVP.JB10.KSDS                            BKMD00   MCBKMSD  SCBKMSD  SGBKMSD

 BKMI.BKM.IVP.JB10.NOBKP.KSDS                      NOBKP2   MCNOBK   SCNOBK   SGBKMSD

 BKMI.BKM.IVP.JB11.ESDS                            BKMD02   MCBKMSD  SCBKMSD  SGBKMSD

 BKMI.BKM.IVP.JB11.KSDS                            BKMD02   MCBKMSD  SCBKMSD  SGBKMSD

 BKMI.BKM.IVP.JB11.NOBKP.ESDS                      NOBKP3   MCNOBK   SCNOBK   SGBKMSD
```

Report options

ACDI APPL Name
Specify the Automated Critical Data Identification application name for which a BKUPEND has been run.

Include HSM
Specify Y to include HSM incremental backups in the search for what is not backed up, otherwise specify N.

Include Physical
Specify Y if you want to include full volume and physical by data set dumps in the search for what is not backed up. Otherwise, specify N.

Exclude Allocate
Specify Y if you want to exclude critical data sets that are specified as ALLOCATE. Otherwise, specify N.

Exclude Accompany
Specify Y if you want to exclude critical data sets that are specified as ACCOMPANY. Otherwise, specify N.

Dataset Name(s) or Mask(s)
Specify up to five data set names or masks to include from the selected Automated Critical Data Identification application BKUPEND.

Critical in Automated Critical Data Identification with Critical Backup Tracking and Inventory Backup Record report

The Critical in Automated Critical Data Identification with Critical Backup Tracking and Inventory Backup Record report lists the critical data sets found in Automated Critical Data Identification for which there is a matching backup record in Critical Backup Tracking and Inventory.

The report is listed by Automated Critical Data Identification application and displays the input data set name backed up, output backup data set name, job name that created the backup, date, time, type of backup, volser of the output data set, and tape location of the output data set.

The following figure shows an example of a Critical in Automated Critical Data Identification with Critical Backup Tracking and Inventory Backup Record report.

```
 SEARCH
CRITERIA:

            DATASET =>
**
   ACDI APPLICATION =>
BTCHSMF1


  PAGE 1      CBTI - CRITICAL WITH CBTI BACKUP RECORD


  DATASET NAME                             JOBNAME  DATE       TIME     TYPE
VOLSER
  ---------------------------------------- -------- ---------- -------- --------
------
  I:KAS.CNTL                               BMPTEST  2005/04/13 09:17:44
ABACKUP
  O:EDP.BKM.BMPTEST.TODISK.C.C01V0163
TODSK1
  O:EDP.BKM.BMPTEST.TODISK.D.C01V0163
TODSK1


  I:KAS.GDG2.G0004V00                       MPTEST  2005/04/13 09:17:44
ABACKUP
  O:EDP.BKM.BMPTEST.TODISK.C.C01V0163
TODSK1
  O:EDP.BKM.BMPTEST.TODISK.D.C01V0163
TODSK1


  I:KAS.ISPF.ISPPROF                       EDPBJ010 2005/03/31 10:31:07 DSS
FVD
  O:EDP.BMT.J010.NOBACKUP.DUMP2
NOBKP3


 ----------------------------
 TOTAL DATASETS FOUND:       3
```

Report options

ACDI APPL Name
Specify the Automated Critical Data Identification application for which a BKUPEND has been run.

Include HSM
Specify Y to include HSM incremental backups in the search for what is not backed up, otherwise specify N.

Include Physical
Specify Y if you want to include full volume and physical by data set dumps in the search for what is not backed up. Otherwise, specify N.

Exclude Allocate
Specify Y if you want to exclude critical data sets that are specified as ALLOCATE. Otherwise, specify N.

Exclude Accompany
Specify Y if you want to exclude critical data sets that are specified as ACCOMPANY. Otherwise, specify N.

SuprAstr
Specify Y if you want to suppress candidate volumes in the report. Otherwise, specify N.

Allbakupsforcycle
Specify Y if you want to include in the report only those critical data set backups created during the current cycle. Otherwise, specify N.

Dataset Name(s) or Mask(s)
Specify up to five data set names or masks to include from the selected Automated Critical Data Identification application BKUPEND.

BKUPEND Summary report

This information describes the BKUPEND Summary report.

The following figure shows an example of a BKUPEND Summary report.

```
SEARCH CRITERIA:
  ACDI APPLICATION => EDG3
             CYCLE => 0

 PAGE 1                   CBTI - BKUPEND SUMMARY                                    2009

                                          ----- BKUPEND BACKUP ----- - MOST RECENT
 DATASET NAME                            JOBNAME  DATE     TIME  RC JOBNAME   DATE
 --------------------------------------  ------------------------- -------------
 KRMI.BKM.IVP.J115.PDSE.OLD              TEST3    20060612 1109   0
 KRMI.BKM.IVP.J115.PRODUCT.TESTX         P390BKPD 20060611 0100   0
 KRMI.TEST3.G0001V00                     **  NO BACKUP FOUND
 KRMI.TEST3.G0002V00                     **  NO BACKUP FOUND
 KRMI.TEST3.G0003V00                     **  NO BACKUP FOUND


 ---------------------------
 TOTAL DATASETS:            5
 DATASETS WITH BACKUP:      2
 DATASETS W/NO BACKUP:      3
```

Figure 6. Example BKUPEND Summary report

Report options

ACDI APPL Name
Specify the Automated Critical Data Identification application name for which a BKUPEND has been run.

Cycle
Specify the BKUPEND relative cycle number to be used for the report.

BKUPEND Validation report

Use the BKUPEND Validation report to verify that the backups identified by BKUPEND still exist and to report the status of the point-in-time application backups recorded at the time at which the BKUPEND program was executed.

Report options

The following figure shows an example of a BKUPEND Validation report.

```
1PAGE       1             CBTI      - BKUPEND VALIDATION
 APPLICATION: BK000079  CYCLE: 00000003  2009/08/20 15:33:30
 DATASET NAME                                AGG/JOB  DATE     TIME  RC  IDS CAT
 ------------------------------------------  -------------------------  --- -
 BK.ACDITEST.GDG12.G0025V00                  * No selected backup *
 BK.ACDITEST.GDG12.G0026V00                  * No selected backup *
 BK.ACDITEST.GDG12.G0027V00                  * No selected backup *

 TOTAL DATASETS:                      3
 DATASETS WITH BACKUP:                0
 DATASETS W/NO BACKUP:                3



1PAGE       1             CBTI      - BKUPEND VALIDATION
 APPLICATION: RONRBKU1  CYCLE: 00000005  2009/08/11 10:39:11
 DATASET NAME                                AGG/JOB  DATE     TIME  RC IDS CAT
 ------------------------------------------  ------------------------- --- —
MW01.$DEMO.BKM.CNTL                          RONRBK1J 20090612 1324 000  YES YES
MW01.$DEMO.INSTJCL                           RONRBK1J 20090612 1324 000  YES YES
MW01.$DEMO.JOB1.PS.DATA                      RONRBK1J 20090612 1324 000  YES YES
MW01.$DEMO.JOB3.PS.DATA                      RONRBK1J 20090612 1324 000  YES YES
MW01.CRPLUS.BKUPTEST                         * No selected backup *
MW01.CRPLUS.EXPLORE                          * No selected backup *
MW01.REPORT.JCL                              * No selected backup *
MW01.REPORT.LIB                              * No selected backup *
RR02.$BKM.CNTL                               ADTSO002 20090807 2211 000  YES NO
RR02.$BKMV21.CNTL                            ADTSO001 20090807 2210 000  YES NO
RR02.$BKMV22.CNTL                            ADTSO001 20090807 2210 000  YES NO
RR02.$DEMO.ABR.CNTL                          ADTSO001 20090731 2210 000  YES NO
RR02.$DEMO.BKUPEND.CNTL                      ADTSO002 20090731 2211 000  YES NO
RR02.ALB.CNTL                                ADTSO004 20090807 2213 000  YES NO
RR02.ALB.SELECTX                             ADTSO003 20090807 2212 000  YES NO
RR02.ALB.SELSDSL                             ADTSO001 20090807 2210 000  YES NO
RR02.ALBREXX.EXTRACT                         ADTSO001 20090807 2210 000  YES NO
RR02.ALBSWG.REXXSAMP                         ADTSO002 20090807 2211 000  YES NO
RR02.ALB62.EXTRACT                           ADTSO004 20090807 2213 000  YES NO
RR02.RONRAG0.SELECT                          ADTSO003 20090807 2212 000  YES NO

TOTAL DATASETS:                     26
DATASETS WITH BACKUP:               22
DATASETS W/NO BACKUP:                4
```

Specify Application Name or Mask
Use this field to specify the name of an application, or a mask for application names, or * to indicate all applications.

Specify Application Cycle or Mask
Use this field to specify a specific relative BKUPEND cycle number or specify * to indicate all cycles. Cycle numbers can be specified as n, or +0 or -n or *, where *n* is an integer in the range -9999999 to 99999999.

Overlap by BKUPEND Application report

This information describes the Overlap by BKUPEND Application report.

The following figure shows an example of an Overlap by BKUPEND Application report.

```
 SEARCH CRITERIA:
  APPLICATION => *

 PAGE 1                        CBTI OVERLAP BY APPLICATION           2008/01/29
14:31:46
 DATASET NAME                                          APPL     DATE       TIME
 ------------------------------------------------- -------- ---------- -------
 PRDI.IVPPLU01.JOBS                                    DEAN     2006/02/07 08:59:06
                                                       KELLY    2005/12/19 09:59:40

 PRDI.IVPPLU01.NOBKP.OUT3                              DEAN     2006/02/07 08:59:06
                                                       KELLY    2005/12/19 09:59:40

 PRDI.IVPPLU01.OUT2                                    DEAN     2006/02/07 08:59:06
                                                       KELLY    2005/12/19 09:59:40
```

Report options

Appl Name(s)
(Required) Specify up to five application names or masks.

TAPEPULL (TapePull report)

Specify these options to generate a TapePull report.

The following figure shows an example of a TapePull report.

```
******************************** Top of Data ********************************
  SEARCH CRITERIA:
            JOB => BL*
     GENERATION => 0, -9999
    INCLPHYDUMP => NO
           COPY => 1
      BADBKUPRC => 8
         SORTBYVOL

 PAGE 1                                        CBTI TAPE PULL LIST

 VOLSER DATASET NAME                                 GEN JOBNAME  DATE       TIME
 ------ ---------------------------------------- ----- -------- ---------- ----

 --------------------------------
 TOTAL VOLUMES FOUND:           0
 TOTAL DATASETS FOUND:          0
```

Report options

Specify a job name, job name mask or job group. You must specify at least one job name or group.

Jobname or mask
Specify job name, job name mask to search to generate a list of tape data sets.

Jobgroup
Specify a job group name to search to generate a list of tape data sets. Job group masks are not permitted.

Advanced options

From GEN
Specify the starting relative generation (0 to -999).

To GEN
Specify the ending relative generation (0 to -999).

From Date or Mask
The starting date or mask of the tapes to search for. The date must be in format YYYY/MM/DD or MM/DD/YYYY. Relative date masking is permitted. For example, you can specify an asterisk (*) to indicate the current date or you can specify *-5 to indicate five days before the current date (up to *-999).

To Date or Mask
The ending date or mask of the tapes to search for. The date must be in format YYYY/MM/DD or MM/DD/YYYY. Relative date masking is permitted. For example, you can specify an asterisk (*) to indicate the current date or you can specify *-5 to indicate five days before the current date (up to *-999).

Exclude Jobname or Mask
Exclude tapes created with the specified job name or mask.

Include Volume Name or Mask
Only include tapes that match the specified volume name or mask.

Exclude Volume Name or Mask
Exclude tapes that match the specified volume name or mask.

Include Dataset Name or Mask
Only include tapes that match the specified data set name or mask.

Exclude Dataset Name or Mask
Exclude tapes that match the specified data set name or mask.

Include Location or Mask
Reduce the list of tapes to only those tapes that match the tape location or mask.

Exclude Location or Mask
Exclude from the list of tapes those tapes that match the tape location or mask.

Sort by
Select one of these options:

- V— Sort the report by volume.
- D—Sort the report by data set name.
- S—Generate a slot-based report. This option is valid for CA 1 tape management environments only.

Copy Number
(Applicable to tapes with copies.) Specify the copy number (1-99). For example, if a backup tape was created with two other backups as exact images of the first, and you want to produce the list of the second copy, then you would specify 2 in this field. The default is 1 (the primary backup tape).

Bad Backup Return Code
Exclude tapes that were created with a return code greater than zero (1-99). For example, if you specify 8, a list of tapes with a return code < 8 will be generated.

Display Volumes only
Select one of these options:

- Y— Display volumes only.
- N—Do not display volumes only.

Include Physical Dumps
Select one of these options:

- Y— The report will include tapes that are physical dumps.
- N—The report will not include tapes that are physical dumps.

Skip Tape Mgmt Info Retrieval
Select one of these options:

- Y— The report does not retrieve tape management meta data. This parameter setting is useful for environments that either do not have a tape management system or have a tape management

system that is unsupported by this product. If your environment does have a tape management system that is supported by this product, do not use this option.

- N— (Default) The report always retrieves tape management meta data.

DB2 Image Copy report

The Db2 Image Copy report displays information from the Db2 image copy records from active Db2 subsystems on the LPAR to which you are currently connected.

Accessing the report

The Db2 Image Copy Report is initiated from the Backup Reports Menu panel, via line 16, DB2 Image Copy Report:

```
                    CBTI - Backup Reports Menu
Command ==>

    Select a Report from the list below:
                                                                      More:   -
    1  Overlap by Dataset
    2  Overlap by Backup Event
   ...................................................................
   15  TapePull Report
   16  DB2 Image Copy Report
   17  ASAP - Critical Files Recovery Space Summary
   18  Not Critical in ASAP with CBTI Backup Record
   19  IMS Image Copy Report

    (Reports are generated using backup data only)
```

Report options

The Db2 Image Copy Report panel contains the standard reporting options to choose type of the report: **Build/View JCL** or **Exec/View Online**.

You can specify search criteria or masks on several image copy record attributes, which restricts the report contents to only those image copy records that match your criteria and masks: Db2 subsystem ID, database name, table space name, the date on which the image copy was taken, and the time at which the image copy was taken (time is valid only if you also specify a date criteria or mask).

```
                    CBTI - Db2 Image Copy Report
Command ===>

(S) SELECT AN OPTION:
   Build/Submit JCL        Build/View JCL          Exec/View Online

REPORT OPTIONS:

Db2 SubSystem     => *         ( * to Sel/Upd/Del)
Database Name     => *         ( * for all databases)
Space Name        => *         ( * for all table and index spaces)

ADVANCED OPTIONS:

        From Date => *           Date or mask (YYYY/MM/DD, *, *-nnn)
        From Time =>             Time (0000 - 2359)
          To Date => *           Date or mask (YYYY/MM/DD, *, *-nnn)
          To Time =>             Time (0000 - 2359)

Image Copies Only => Y           Y or N
```

Tip: When an asterisk (*) is specified in the **Db2 SubSystem** field, or if the specified subsystem has not been configured, you are prompted with a panel (BKMIMG02) where you can select, update, or delete configuration records.

Db2 Image Copy data is obtained from selected columns of SYSIBM.SYSCOPY using static SQL through Call Attach Facility. The configuration per subsystem includes a plan name and the list of load libraries to dynamically allocate. The installation per subsystem includes binding a plan and granting authority.

You can choose to include both image copy records and non-image copy records and to include records from both inactive and active Db2 systems.

After you specify the appropriate criteria and masks, IBM Tivoli Advanced Backup and Recovery for z/OS displays a list of Db2 subsystems on the LPAR to which you are currently connected. From this list, you can do any of the following:

- To further filter the list of subsystems, in the **Db2 Subsystem Name** text entry field, type the name of the subsystem of interest, or type a mask, and press Enter.
- To display image copy records from one of those subsystems, type S in the selection column next to the subsystem ID and press Enter.

  Note: Choose a subsystem ID for which a plan exists. Otherwise, a Call Attach Facility error occurs when attempts are made to retrieve data about the image copy records for that subsystem.
- If no plan exists for the Db2 subsystem of interest, you can type U in the selection column next to the subsystem name and press Enter to update the *IDS* with a plan name and load library information for the subsystem; the Update command does not define a plan to a Db2 subsystem; it only creates a definition in the IDS.

  When the Db2 subsystem list is refreshed, the new plan name appears in the row for that subsystem; however, you cannot use that subsystem and plan to generate a Db2 Image Copy report until you define the new plan to the Db2 subsystem.

When you select a Db2 subsystem and its associated plan from the list, the Db2 Image Copy report displays all the image copy records for that subsystem.

Sample report

The following figure shows an example of a Db2 Image Copy report. Note that you can further filter the list of image copy records using names or masks for database name, table space or index space name, or timestamp (or any combination of these attributes).

```
SEARCH CRITERIA:
         SSID => DBA6
     DATABASE => *
   SPACE NAME => *
       ICONLY

PAGE 1                                          CBTI     - JOBNAME LIST                          2022/12/13 08:40:26

 DBNAME    SPNAME   DSN                                          TIMESTAMP           TYPES  IC DSNUM VOLSER DEVTYPE
 --------  -------- -------------------------------------------- ------------------- ------ -- ----- ------ -------
 CKZKYB0  IX0       RSTEST.TSKYB.IC.CKZKYB0.IX0.P00001           2022/10/24 09:19:52 I F T  FC       R1P145 DASD
 CKZKYB0  TS0       RSTEST.TSKYB.IC.CKZKYB0.TS0.P00001           2022/10/24 09:19:52 T F T  FC       R1P145 DASD
 DBUASMWF NWTQATS1  CSMVDQO.DDS6.NWTQATS1.P00000.HZZ1YFTU        2021/02/23 07:06:25 T F              MIGRAT TAPE
 DBUASMWF NWTQATS1  CSMVDQO.DDS6.NWTQATS1.P00000.HZZ2B4UA        2021/02/23 07:15:43 T F              MIGRAT TAPE
 DBUJHUBJ NWTQATS1  CSMVDQO.DDS6.NWTQATS1.P00000.H5JSNN6M        2021/06/23 22:24:28 T F              MIGRAT TAPE
 DBUJHUBJ NWTQATS1  CSMVDQO.DDS6.NWTQATS1.P00000.H5JSNONR        2021/06/23 22:24:29 T F W            MIGRAT TAPE
 DBVFHGRV NWTTTS1   CSMVDQO.DDS6.NWTTTS1.P00000.HZZ32RY8        2021/02/23 08:00:49 T F              MIGRAT TAPE
 DBWOJZWX NWTTTS1   CSMVDQO.DDS6.NWTTTS1.P00000.HY5Y3R6G        2021/02/05 04:38:40 T F              MIGRAT TAPE
 DBXBTZMI NWTTTS1   CSMVDQO.DDS6.NWTTTS1.P00000.IG1WR57P        2022/02/08 00:55:36 T F              MIGRAT TAPE
```

Report columns

DBNAME
This field shows the name of the database that was included in the image copy.

SPNAME
This field shows the name of the target table space of the index space.

Dataset Nane
This field shows the data set name, or the name of the database and table space or index space in the format *database-name.space-name*, or the field is blank for any row that was migrated from a Db2 release prior to Version 4.

Timestamp
This field shows the date and time at which the record row was inserted. This is the date and time recorded in **ICDATE** and **ICTIME**. The use of **Timestamp** is recommended over that of **ICDATE** and **ICTIME** because the latter two columns might not be supported in future releases of Db2.

For the COPYTOCOPY utility, this value is the date and time at which the row was inserted for the primary local site or primary recovery site copy. For an EXCHANGE DATA statement, this is the time at which the statement is run.

Types

This field contains the OTYPE, ICTYPE, and STYPE columns from the SYSIBM.SYSCOPY table.

OTYPE

Type of object that the recovery information is for:

- I - Index space
- T - Table space

ICTYPE

This field indicates the type of operation:

- A—ALTER
- B—REBUILD INDEX
- C—CREATE
- D—CHECK DATA LOG(NO), where there are no log records for the range available for the RECOVER utility
- E—RECOVER to current point
- F—COPY FULL YES
- I—COPY FULL NO
- M—MODIFY RECOVERY utility
- P—RECOVER TOCOPY or RECOVER TORBA (partial recovery point)
- Q—QUIESCE
- R—LOAD REPLACE LOG(YES)
- S—LOAD REPLACE LOG(NO)
- T—TERM UTILITY command
- V—REPAIR VERSIONS utility
- W—REORG LOG(NO)
- X—REORG LOG(YES)
- Y—LOAD LOG(NO)
- Z—LOAD LOG(YES)

STYP

The meaning of any value shown in this column is dependent upon the ICTYPE for the row:

- When **ICTYPE** is A, **STYP** values have the following meanings:
  - A—A partition was added to a table.
  - C—A column was added to a table and to an index in different commit scopes.
  - E—The data set numbers of a base table and its associated clone table are exchanged.
  - G—An index was regenerated.
  - L—The logging attribute of the table space was altered to LOGGED.
  - N—An index was altered to NOT PADDED.
  - O—The logging attribute of the table space was altered to NOT LOGGED.
  - P—An index was altered to PADDED.
  - R—A table was altered to rotate partitions.
  - V—A column in a table was altered for a numeric data type change and the column participates in an index.

  - Z—A column that participates in the key of an index that was versioned prior to the Db2 Version 8 was altered.
- When **ICTYPE** is C, **STYP** values have the following meanings:
  - L—The logging attribute of the table space was altered to LOGGED.
  - O—The logging attribute of the table space was altered to NOT LOGGED.
- When **ICTYPE** is F, **STYP** values have the following meanings:
  - A—ADD PARTITION execution.
  - C—DFSMS concurrent copy (?I? instance of the table space)
  - J—DFSMS concurrent copy (?J? instance of the table space)
  - R—ROTATE FIRST TO LAST
  - S—LOAD REPLACE(NO)
  - V—ALTER INDEX NOT PADDED
  - W—REORG LOG(NO)
  - X—REORG LOG(YES)
  - (blank)—Db2 image copy

  The MERGECOPY utility, when used to merge an embedded copy with subsequent incremental copies, also produces a record that contains ICTYPE=F and the **STYP** of the original image copy (R, S, W, or X).
- When **ICTYPE** is M and the MODIFY RECOVERY utility was executed to delete SYSCOPY or SYSLGRNX (or both) records, the **STYP** value is R.
- When **ICTYPE** is P, **STYP** values have the following meanings:
  - C—Recover to a point in time without using logonly with consistency.
  - L—Recover to a point in time using logonly without consistency.
  - M—Recover to a point in time using logonly with consistency.
  - (blank)—Recover to a point in time without using logonly without consistency.
- When **ICTYPE** is Q and option WRITE(YES) is in effect when the quiesce point is taken, the **STYP** value is W.
- When **ICTYPE** is R or S, **STYP** values have the following meanings:
  - A—Resetting REORG pending status.
  - T—First materializing the default value for a row change timestamp column.
- When **ICTYPE** is W or X, **STYP** values have the following meanings:
  - A—Resetting REORG pending status or REBALANCE.
  - T—First materializing the default value for a row change timestamp column.
- When **ICTYPE** is W or X, **STYP** values have the following meanings:
  - A—Resetting REORG pending status or REBALANCE.
  - T—First materializing the default value for a row change timestamp column.
- When **ICTYPE** is T, **STYP** indicates which COPY utility was terminated by the TERM UTILITY command or the START DATABASE command with the ACCESS(FORCE) option:
  - F—COPY FULL YES
  - I—COPY FULL NO
- When **ICTYPE** has any other value, the **STYP** column is blank for that record.

When you select an individual image copy record from the list and press Enter, additional information about that record appears in a pop-up panel:

ICBACKUP
The value shown in this field indicates the type of image coy contained in the data set:

- (blank)—LOCALSITE primary copy (first data set named with COPYDDN)
- LB—LOCALSITE backup copy (second data set named with COPYDDN)
- RP—RECOVERYSITE primary copy (first data set named with RECOVERYDDN)
- RB—RECOVERYSITE backup copy (second data set named with RECOVERYDDN)

DSNUM
This field shows the data set number within the table space. For partitioned table spaces, this value corresponds to the partition number for a single partition copy, or zero for a copy of an entire partitioned table space or index space.

Volser
This field contains one of the below values.

- **XXXXXX** - shows the volume serial numbers for the data set
- **MIGRAT** – The dataset is either in ML1 or ML2

DEVTYPE
This field shows the device type for the data set. It contains one of the below values.

- DASD
- TAPE

Critical Files Recovery Space Summary

This report gives the space summary of all the critical files backup up by the specified application.

Accessing the report

The report is initiated from the Backup Reports Menu panel, via line 17, ASAP - Critical Files Recovery Space Summary:

```
                    CBTI - Backup Reports Menu
Command ==>

    Select a Report from the list below:

    1  Overlap by Dataset
    2  Overlap by Backup Event
   ..............................................................
   16  DB2 Image Copy Report
   17  ASAP - Critical Files Recovery Space Summary
```

Report options

You can specify the standard reporting option to choose type of report:: **Build/Submit JCL**, **Build/View JCL** or **Exec/View Online** and **Application name**, for which you want to create the report:

```
                    CBTI - ASAP - Critical Files Recovery Space Summary
Command ===>

(S) SELECT AN OPTION:
   Build/Submit JCL        Build/View JCL         S Exec/View Online

REPORT OPTIONS:

                  APPL Name     =>
```

Sample report

The following figure shows an example of Critical Files Recovery Space Summary report:

Report columns

DEVT
Device type: **D** for disk, **T** for tape.

CAT
Data set is cataloged (**Y**) or not (**N**).

Block Size
Block size.

Blk Per TRK
Number of blocks per one track.

Bytes Per Track
Number of bytes per one track.

Alloc'd Tracks
Number of tracks allocated for data set on disk.

Used Tracks
Number of tracks used for data set on disk.

Alloc'd Bytes
Number of bytes allocated for data set on disk.

Disk Used Bytes
Number of bytes used for data set on disk.

Volser
Volume Serial Number, where data set is allocated.

Job Name
Job name, where data set was backed up.

Files
Number of data sets backed up by this job.

TRKs alloc
Number of tracks allocated for data sets.

TRKs used
Number of tracks used for data sets.

Alloc space
Allocated size of data sets in bytes.

Used space
Used size of data sets in bytes.

Volumes
The number of volumes for data sets on tapes.

Not Critical in Automated Critical Data Identification with Critical Backup Tracking and Inventory Backup Record report

The Not Critical in Automated Critical Data Identification with Critical Backup Tracking and Inventory Backup Record report lists the non-critical data sets found in Automated Critical Data Identification for which there is a matching backup record in Critical Backup Tracking and Inventory.

The report is listed by Automated Critical Data Identification application and displays the input data set name backed up, output backup data set name, job name that created the backup, date, time, type of backup, volser of the output data set, and tape location of the output data set.

The following figure shows an example of a Not Critical in Automated Critical Data Identification with Critical Backup Tracking and Inventory Backup Record report.

```
APPLICATION: SAMPLE                                                2019/05/22 11:09:37
PAGE       1                CBTI   -   NOT CRITICAL WITH BACKUP
DATASET NAME                        AGG/OB      DATE         TIME      RC  TYPE     VOLSER LOC
---------------------------------+-----------+-----------+-------- -+---+--------+----------+
I: ABR.APPLJ1.DC.SAMPLE           APPLB1      2019/05/08   18:44:59  000 IEBGENER
O: ABR.APPLJ1.DC.BACKUP.G0003V00                                                    TSO002

I: ABR.APPLJ2.DR.SAMPLE           APPLH1      2019/05/20   15:05:05
O: ABR.APPLJ2.DC.BACKUP.G0005V00                                                    TSO002
```

Report options

ACDI APPL Name
Specify the Automated Critical Data Identification application for which a BKUPEND has been run.

Include HSM
Specify Y to include HSM incremental backups in the search for what is not backed up, otherwise specify N.

Include Physical
Specify Y if you want to include full volume and physical by data set dumps in the search for what is not backed up. Otherwise, specify N.

Exclude Allocate
Specify Y if you want to exclude critical data sets that are specified as ALLOCATE. Otherwise, specify N.

Exclude Accompany
Specify Y if you want to exclude critical data sets that are specified as ACCOMPANY. Otherwise, specify N.

SuprAstr
Specify Y if you want to suppress candidate volumes in the report. Otherwise, specify N.

Allbakupsforcycle
Specify Y if you want to include in the report only those critical data set backups created during the current cycle. Otherwise, specify N.

Dataset Name(s) or Mask(s)
Specify up to five data set names or masks to include from the selected Automated Critical Data Identification application BKUPEND.

IMS Image Copy report

The IMS Image Copy report displays information from the IMS image copy records from active or inactive IMS subsystems on the LPAR to which you are currently connected.

The following figure shows an example of IMS Image Copy report. Note that you can further filter the list of image copy records using names or masks for database name, area name, FSN, volume, timestamp or device type (or any combination of these attributes).

```
SEARCH CRITERIA:
            SSID => IFK3
        DATABASE => *
            AREA => *

 PAGE 1                                        CBTI - IMS BACKUP REPORT     2022/12/14
02:19:18
 DATABASE AREA     DSN                                              FSN VOLUME START/STOP TIME        RECORDS DEVTYPE
 -------- -------- ----------------------------------------------  ----- ------ ------------------- ---------- -------
 DBFSAMD3 CUSDB     IMS.IFK2.DBFSAMD3.CUSDB.IC103                      1 IFP11R 2022/03/08 09:34:49          0 DASD
                                                                                 2022/03/08 09:34:49
 DBFSAMD3 CUSDB     IMS.IFK2.DBFSAMD3.CUSDB.IC101                      1 IFP11R 2022/03/10 03:02:32          0 DASD
                                                                                 2022/03/10 03:02:32
 DBFSAMD3 CUSDB     IMS.IFK2.DBFSAMD3.CUSDB.IC102                      1 IFP11R 2022/03/10 03:17:30          0 DASD
                                                                                 2022/03/10
03:17:30
 DBFSAMD4 LOAN      IMS.IFK2.DBFSAMD4.LOAN.IC101                       1 IFP11R 2016/07/26 12:14:20        418 DASD
 DBFSAMD4 LOAN      IMS.IFK2.DBFSAMD4.LOAN.IC102                       1 IFP11R 2016/07/26 12:13:56          0 DASD
 DBFSAMD4 LOAN      IMS.IFK2.DBFSAMD4.LOAN.IC103                       1 IFP11R 2016/07/26 12:13:56          0 DASD
 DI21PART DI21PARO  IMS.IFK2.DI21PART.DI21PARO.IC101                   1 IFP11R 2016/07/26 12:14:19         31 DASD
 DI21PART DI21PARO  IMS.IFK2.DI21PART.DI21PARO.IC102                   1 IFP11R 2016/07/26 12:13:56          0 DASD
 DI21PART DI21PARO  IMS.IFK2.DI21PART.DI21PARO.IC103                   1 IFP11R 2016/07/26 12:13:56          0 DASD
 DI21PART DI21PART  IMS.IFK2.DI21PART.DI21PART.IC101                   1 IFP11R 2016/07/26 12:14:19         67 DASD
 DI21PART DI21PART  IMS.IFK2.DI21PART.DI21PART.IC102                   1 IFP11R 2016/07/26 12:13:56          0 DASD
 DI21PART DI21PART  IMS.IFK2.DI21PART.DI21PART.IC103                   1 IFP11R 2016/07/26 12:13:56          0 DASD
 IVPDB1   DFSIVD1   IMS.IFK2.IVPDB1.DFSIVD1.IC101                       1 IFP11R 2016/07/26 12:14:17          2 DASD
 IVPDB1   DFSIVD1   IMS.IFK2.IVPDB1.DFSIVD1.IC102                       1 IFP11R 2016/07/26 12:13:56          0 DASD
 IVPDB1   DFSIVD1   IMS.IFK2.IVPDB1.DFSIVD1.IC103                       1 IFP11R 2016/07/26 12:13:56          0 DASD
 IVPDB1I  DFSIVD1I  IMS.IFK2.IVPDB1I.DFSIVD1I.IC101                    1 IFP11R 2016/07/26 12:14:17          7 DASD
 IVPDB1I  DFSIVD1I  IMS.IFK2.IVPDB1I.DFSIVD1I.IC102                    1 IFP11R 2016/07/26 12:13:56          0 DASD
 IVPDB1I  DFSIVD1I  IMS.IFK2.IVPDB1I.DFSIVD1I.IC103                    1 IFP11R 2016/07/26 12:13:56          0 DASD
 IVPDB2   DFSIVD2   IMS.IFK2.IVPDB2.DFSIVD2.IC101                      1 IFP11R 2016/07/26 12:14:18         83 DASD
 IVPDB2   DFSIVD2   IMS.IFK2.IVPDB2.DFSIVD2.IC102                      1 IFP11R 2016/07/26 12:13:56          0 DASD
 IVPDB2   DFSIVD2   IMS.IFK2.IVPDB2.DFSIVD2.IC103                      1 IFP11R 2016/07/26 12:13:56          0 DASD
 IVPDB3   DFSIVD3A IMS.IFK2.IVPDB3.DFSIVD3A.IC102                     1 IFP11R 2022/01/14 09:12:53          0 DASD
                                                                                 2022/01/14 09:14:58
 IVPDB3   DFSIVD3A  IMS.IFK2.IVPDB3.DFSIVD3A.IC103                    1 IFP11R 2022/02/08 11:12:49          0 DASD
                                                                                 2022/02/08 11:23:47
 IVPDB3   DFSIVD3A  IMS.IFK2.IVPDB3.DFSIVD3A.IC101                    1 IFP11R 2022/03/08 09:31:07          0 DASD
                                                                                 2022/03/08 09:58:05
 IVPDB3   DFSIVD3B  IMS.IFK2.IVPDB3.DFSIVD3B.IC102                    1 IFP11R 2022/01/14 09:14:58          0 DASD
                                                                                 2022/01/14 09:15:22
 IVPDB3   DFSIVD3B  IMS.IFK2.IVPDB3.DFSIVD3B.IC103                    1 IFP11R 2022/02/08 11:23:47          0 DASD
                                                                                 2022/02/08 11:23:51
 IVPDB3   DFSIVD3B  IMS.IFK2.IVPDB3.DFSIVD3B.IC101                    1 IFP11R 2022/03/08 09:58:05          0 DASD
                                                                                 2022/03/08 09:58:11

--------------------------------
TOTAL BACKUPS FOUND:          27
```

Report options

Specify a job name, job name mask or job group. You must specify at least one job name or group.

IMS Subsystem

Specify IMS Subsystem name(s) to search for and display. Patterns (masks) may be entered to select the of subsystem for display.

- EX: IMS SubSystem ===>*
- EX: IMS SubSystem ===>IM*
- EX: IMS SubSystem ===>%M*
- EX: IMS SubSystem ===>IMS1

If Pattern (mask) is entered, the IMS subsystem List screen will appear.

Here's a sample screen.

```
Menu Diagnostics Preferences

BKMIMS02        CBTI - IMS Subsystem List        Row 1 to 14 of 14
Command ===>                                     Scroll ===> PAGE

Primary Command: REF (Refresh)
   Line Command: S (Select For Image Copies)
                 U (Update IDS Record)
                 D (Delete IDS
Record)

      Subsystem  IMS Active  IDS Record                          System ID==> RS27
      *          *           *
-------------------------------------------------------------------------------------------
      IBA0       No          Yes
      IDK3       No          Yes
      IDK5       No          Yes
      IEA1       No          Yes
      IEK1       No          Yes
      IEK2       No          Yes
      IEK4       No          Yes
      IEK5       No          Yes
      IEN2       No          Yes
      IFA1       No          Yes
      IFA2       Yes         Yes
      IFK3       Yes         Yes
      IFK5       No          Yes
      IFN7       No          Yes
```

Primary Command: REF (Refresh)

Refresh the IMS Subsystem List.

Line Command:

S (Select for IMS Image Copies) - Select an IMS Subsystem to see the Image Copies for that subsystem.

U (Update IDS IMS Record) - Select an IMS Subsystem to create or update the IDS IMS record for the subsystem.

D (Delete IDS IMS Record) - Select an IMS Subsystem to delete the IDS IMS record for the subsystem. This only affects the IDS (Inventory Data Set) and in no way affects the actual IMS Subsystem.

Subsystem Name

Contains a list of the IMS Subsystems. The list can be reduced by entering a pattern mask at the top of the column.

- EX: SubSystem ===> *
- EX: SubSystem ===> IM*
- EX: SubSystem ===> %M*
- EX: SubSystem ===> IMS1

IMS Active

Indicates whether the IMS Subsystem is active or not. The list can be reduced by entering Yes or No at the top of the column.

IDS Record

Indicates whether the Inventory Data Set contain a record for the IMS Subsystem. The list can be reduced by entering Yes or No at the top of the column.

If IDS IMS record is not already added by the user for an IMS Subsystem, the Line command ‘S’ will lead the user to “IMS Subsystem Setup” screen. This panel allows for the creation of an IDS record to store IMS Subsystem default information for each individual IMS Subsystem.

Here's a sample screen.

```
Menu Diagnostics Preferences

BKMIMS03        BKM - IMS Subsystem Setup
Command ===>

Primary Command: SAVE - Save updated record to IDS

IMS ID:=> IFA2                                      System ID==> RS27

Loadlibs in IDS:

  Loadlib 1    => AUI.AUI1130.AUI3528.AUIFLGX0
  Loadlib 2    => RSQA.AUI1130:IBMTAPE.SAUIIMOD
  Loadlib 3    => IMS.IFA2.SDFSRESL
  Loadlib 4    => IMS.IFA2.MDALIB
  Loadlib 5    => AUE.TEST.EXITLIB

Loadlibs defined to IMS subsystem

  Loadlib 1    => RSQA.AUI1013:IBMTAPE.SAUIIMOD
  Loadlib 2    => IMS.IFA2.SDFSRESL
  Loadlib 3    => AUI.IFA2.MDALIB
  Loadlib 4    => IMS.IFA2.MDALIB
  Loadlib 5    =>
```

Primary Commands (Commands entered on the Command Line).

- SAVE - Saves the IDS record to the IDS.
- END (PF3) - Leaves the panel saving the IDS record.
- CAN (CANCEL) - Leaves the panel without saving the IDS record.

Current System ID
The MVS System currently operating on.

IMS ID
The IMS Subsystem the IDS record will be created for.

Loadlibs in IDS
The IMS Load Libraries the IDS has stored for the IMS Subsystem currently selected.

Loadlibs defined to IMS Subsystem
The IMS Load Libraries currently defined to the selected IMS Subsystem. These are automatically detected and retrieved from the definition of the selected Subsystem.

Database Name
Specify Database name(s) to search for and display. Patterns (masks) may be entered to reduce the number of databases to display.

- EX: Database Name ===>*
- EX: Database Name ===>DBN*
- EX: Database Name ===>%BN*
- EX: Database Name ===>DBNAME

Area Name
Specify Area name(s) to search for and display. Patterns (masks) may be entered to reduce the the number of areas to display.

- EX: Area Name ===>*
- EX: Area Name ===>ARN*
- EX: Area Name ===>%RN*
- EX: Area Name ===>ARNAME

Advanced options

From Date or Mask
The starting date or mask of the tapes to search for. The date must be in format **YYYY/MM/DD** or **MM/DD/YYYY**. Relative date masking is permitted. For example, you can specify an asterisk (*) to indicate the current date or you can specify ***-5** to indicate five days before the current date (up to *-999).

From Time or Mask
Specify a time to narrow the search of the IMS image copy report. If **To Date** is not specified then only jobnames matching the date and time specified will be displayed.

To Date or Mask
The ending date or mask of the tapes to search for. The date must be in format **YYYY/MM/DD** or **MM/DD/YYYY**. Relative date masking is permitted. For example, you can specify an asterisk (*) to indicate the current date or you can specify ***-5** to indicate five days before the current date (up to *-999).

To Time or Mask
Specify a time to narrow the search of the IMS image copy report. If **From Date** is not specified, then only jobnames matching the date and time specified will be displayed.

Report columns

DATABASE
This field shows the database name.

AREA
This field shows the name of the DBDS or DEDB area to select the records.

DSN
This field shows the data set name that contains the RECON file.

FSN
This field shows the file sequence number of the data set on tape.

VOLUME
This field contains one of the following values:

- XXXXXX - shows the volume serial numbers for the data set.
- MIGRAT - The dataset is either in ML1 or ML2.

START/STOP TIME
This field displays the time range when the copy was done.

RECORDS
This field shows the number of records in the data set.

DEVTYPE
This field shows the device type for the data set. It contains one of the below values:

- DASD
- TAPE

Backup List report

This information describes the Backup List report.

The following figure shows an example of a Backup List report.

```
SEARCH CRITERIA:
             DATASET => CUZSEABK.QAT.RCC.VOLBCP.DASD
                TYPE => ACDFGIMORSTUVWZ

 PAGE 1                                    CBTI - BACKUP LIST REPORT                        2022/12/13
05:24:56
 BACKUP DATASET NAMED                                                 EVENT    DATE       TIME     TYPE     VOLSER DEVTYPE
 ------------------------------------------------------------------   -------- -----      -------- -------- ------ -------
 CUZSEABK.QAT.RCC.VOLBCP.DASD                                         DSSDUMPD 2021/07/21 04:01:29 DSS PDDS EAVS01 DASD
 S:ABRBRM.IAMTEST.TODISK.C.C01V0002
 S:ABRBRM.TEST.GDG.EMPTY
 S:AUV.VTAP0360.LOADLIB
 S:BKMQ.BKM.HLD0204.ABR2175.LOAD
 S:BKMQ.BKM.HLD0204.ABR2529.ISPPLIB
 S:BKMQ.BKM.HLD0204.ABR2597.LOAD
 S:BKMQ.BKM.HLD0204.ABR2658.LOAD
 S:BKMQ.BKM.HLD0204.ABR2782.LOAD
 S:BKMQ.BKM.HLD0204.ABR2926.LOAD
 S:BKMQ.BKM.HLD0204.ABR2981.LOAD
 S:BKMQ.BKM.HLD0204.ABR3000.LOAD
 S:BKMQ.BKM.HLD0204.ABR3034.LOAD
 S:BKMQ.BKM.HLD0204.ABR3043.ISPPLIB
 S:BKMQ.BKM.HLD0204.ABR3062.LOAD
 S:BKMQ.BKM240.ABR3083.RSP.JOBFLAT
 S:BKMQ.BKM240.GA.ACDIDB.TEST
 S:BKMQ.BKM240.GA.ACDIDB.TEST.DATA
 S:BKMQ.BKM240.GA.ACDIDB.TEST.INDEX
 S:BKMQ.BKM240.GA.HDUDB.D2019348
 S:BKMQ.BKM240.GA.HDUDB.D2019348.DATA
 S:BKMQ.BKM240.GA.HDUDB.D2019348.INDEX
 S:BKMQ.BKM240.GA.HDUDB.D2020145
 S:BKMQ.BKM240.GA.HDUDB.D2020145.DATA
 S:BKMQ.BKM240.GA.HDUDB.D2020145.INDEX
 S:BKMQ.BKM240.GA.HDUDB.D2020147
 S:BKMQ.BKM240.GA.HDUDB.D2020147.DATA
 S:BKMQ.BKM240.GA.HDUDB.D2020147.INDEX
 S:BKMQ.BKM240.GA.HDUDB.D2020149
 S:BKMQ.BKM240.GA.HDUDB.D2020149.DATA
 S:BKMQ.BKM240.GA.HDUDB.D2020149.INDEX
 S:BKMQ.BKM240.GA.HDUDB.D2020178
 S:BKMQ.BKM240.GA.HDUDB.D2020178.DATA
 S:BKMQ.BKM240.GA.HDUDB.D2020178.INDEX
 S:BKMQ.BKM240.GA.HDUDB.D2020190
 S:BKMQ.BKM240.GA.HDUDB.D2020190.DATA
 S:BKMQ.BKM240.GA.HDUDB.D2020190.INDEX
 S:BKMQ.BKM240.GA.HDUDB.D2020352
 S:BKMQ.BKM240.GA.HDUDB.D2020352.DATA
 S:BKMQ.BKM240.GA.HDUDB.D2020352.INDEX
 S:BKMQ.BKM240.GA.HDUDB.D2021031
 S:BKMQ.BKM240.GA.HDUDB.D2021031.DATA
 S:BKMQ.BKM240.GA.HDUDB.D2021031.INDEX
```

Report options

Backup Dataset Name(s) or Mask(s)

Required. Specify Backup data set names or masks to include in the report. You can specify up five Backup data set names or masks with a maximum of 44 characters each.

Advanced options

Job Name(s)

Filter the report output by job name or mask. You can specify up to five job names or masks with a maximum of eight characters each.

From Date

The starting date or mask of the Backup data sets or events to search for. The date must be in format **YYYY/MM/DD** or **MM/DD/YYYY**. Relative date masking is permitted. For example, you can specify an asterisk (*) to indicate the current date, you could specify ***-5** to indicate five days before the current date (up to *-999). Valid values are:

- v * — today.
- v *-1 — today and yesterday.
- v *-nnn — today up to 999 days ago.

In the following examples **n** is an integer 1-999. Specifying date criteria in the format ***-n** only displays events equal to **n** relative days from the current date. Type ***-4** to display events four days from today. Type ***+4** to display all events, from today through four days from today.You can also specify a specific date in the format of **YYYY/MM/DD** or **MM/DD/YYYY**. In this case, only events matching the specific date are displayed.

To Date

The ending date or mask of the Backup data sets or events to search for. The date must be in format **YYYY/MM/DD** or **MM/DD/YYYY**. Relative date masking is permitted. For example, you can specify an asterisk (*) to indicate the current date, or you could specify *-5 to indicate five days before the current date (up to *-999). Valid values are:

- v * — today.
- v *-1 — today and yesterday.
- v *-nnn — today up to 999 days ago.

In the following examples **n** is an integer 1-999. Specifying date criteria in the format ***-n** only displays events equal to n relative days from the current date. Type ***-4** to display events four days from today. Type ***+4** to display all events, from today through four days from today.You can also specify a specific date in the format of **YYYY/MM/DD** or **MM/DD/YYYY**. In this case, only events matching the specific date are displayed.

Type(s)

Specify one, or any combination of the following types:

- A—Data sets backed up by ADARUN.
- C—Data sets backed up by ACM or CR+.
- D—Data sets backed up by DSS.
- F—Data sets backed up by FDR.
- G—Data sets backed up by IEBCOPY, IEBGENER, ICEGENER, or SYNCGENR.
- I—Data sets backed up by IDCAMS REPRO or EXPORT.
- M—Data sets backed up by CFCAMS.
- O—Data sets backed up by ICETOOL.
- R—Data sets backed up by Fast Replication Services.
- S—Data sets backed up by ICEMAN, SORT, or SYNCSORT.
- T—Data sets backed up by TAPECOPY.
- U—Data sets backed up by User Defined Backup programs.
- V—Data sets backed up by CA-FAVER.
- W—Data sets backed up by CA-Disk.
- Z—Data sets backed up by VSAM Assist.

Creating and viewing ABARS Manager online reports

This topic describes the ABARS Manager online reporting feature for backup and recovery reports. You can also use the REPORTS command to generate these reports in batch mode.

- Backup reports:
  - The Overlap by Data Set report provides the following information about data sets being backed up by multiple events: the name of the backup event, backup event time stamp, data set type (include, allocate, or accompany), backup level, backup event's base generation number, and Advanced Backup and Recovery for z/OS total space usage.
  - The Overlap by Backup Event report provides the following information about overlapping backup events: the names of the data sets being backed up by multiple events, time stamps for the backup events, data set type (include, allocate, or accompany), backup level, backup event's base generation number, and total space usage.
  - The Data Set List report provides information about every data set that matches the filter criteria you specify. You can filter by data set name or mask, backup event name, volume name, and date.

- The Event List report provides information about every backup event that matches the filter criteria you specify. You can filter by event or logical aggregate name or mask. You can also specify for which relative generations you want information and restrict the report to a range of event dates and times.
- The Tape Pull List report provides event name, generation, date, time, and data set name information for the volume types you specify.
- The Disaster Recovery Space Summary report summarizes space information for every backup event or aggregate that matches the filter criteria you specify. You can filter by event or logical aggregate name or mask. You can also specify for which relative generations you want information and restrict the report to a range of event dates and times.

Recovery reports:

- The Overlap by Data Set Recovery report identifies data sets for which recovery overlaps exist; you can specify a data set name or mask for which to search, and you can explicitly include or exclude recover events.

Creating an online Overlap by Data Set report

This topic provides a procedure for creating an online Overlap by Data Set report.

The Overlap by Data Set report provides the following information about data sets being backed up by multiple events: the name of the backup event, backup event time stamp, data set type (include, allocate, or accompany), backup level, backup event's base generation number, and Advanced Backup and Recovery for z/OS total space usage.

1. Type 6 in the **Command** line on the **Main Menu** panel, and then press Enter to display the **Reports** panel.
2. Type 1 in the **Command** line on the **Reports** panel, and then press Enter to display the **ABM Backup Reports Menu** panel.
3. Type 1 in the **Command** line on the **ABM Backup Reports Menu** panel, and then press Enter to display the **Overlap by Dataset Report** panel.
4. Use the **Overlap by Dataset Report** panel's online help to guide you as you complete the fields on that panel.
5. Select the **Exec/View Online** option, and then press Enter to generate and display the report.

The following figure shows an example Overlap by Data Set report:

```
SEARCH CRITERIA:
          DATASET => *.*

PAGE 1                                    BKM - OVERLAP BY DATASET

DATASET NAME                                        EVENT    DATE       TIME     TYPE
--------------------------------------------------- -------- ---------- -------  --------
BACKUP.ARCHV1.G1283V00                              DS7      2005/10/31 07:37:26 AB EXEC
                                                    DEMO1    2005/10/31 05:33:26 AB EXEC

BACKUP.ARCHV2.G1283V00                              DS7      2005/10/31 07:37:26 AB EXEC
                                                    DEMO1    2005/10/31 05:33:26 AB EXEC

BACKUP.ARCHV3.G0372V00                              DS7      2005/10/31 07:37:26 AB EXEC
                                                    DEMO1    2005/10/31 05:33:26 AB EXEC
```

Creating an online Overlap by Backup Event report

This topic provides a procedure for creating an online Overlap by Backup Event report.

The Overlap by Backup Event backup report provides the following information about overlapping backup events: the names of the data sets being backed up by multiple events, timestamps for the backup events, data set type (include, allocate, or accompany), backup level, backup event's base generation number, and total space usage.

1. Type 6 in the **Command** line on the **Main Menu** panel, and then press Enter to display the Reports panel.
2. Type 1 in the **Command** line on the **Reports** panel, and then press Enter to display the **ABM Backup Reports Menu** panel.

3. Type 2 in the **Command** line on the **ABM Backup Reports Menu** panel, and then press Enter to display the **ABM Overlap by Backup Event Report** panel.
4. Use the **Overlap by Backup Event Report** panel's online help to guide you as you complete the fields on that panel.
5. Select the **Exec/View Online** option, and then press Enter to generate and display the report.

The following figure shows an example Overlap by Backup Event backup report:

```
********************** Top of Data ************************************
 SEARCH CRITERIA:
        EVENT => *
   GENERATION => 0
      COMPGEN => 0,-9999

 PAGE 1                                  ABM - OVERLAP BY AGGREGATE

 DATASET NAME                                          EVENT    DATE       TIME
 ----------------------------------------------------- -------- ---------- -----
 BACKUP.ARCHV1.G1283V00                                DS7      2005/10/31 07:37
                                                       DEMO1    2005/10/31 05:33

 BACKUP.ARCHV2.G1283V00                                DS7      2005/10/31 07:37
                                                       DEMO1    2005/10/31 05:33

 BACKUP.ARCHV3.G0372V00                                DS7      2005/10/31 07:37
```

Use the RIGHT function key to display the TYPE, LEVEL, GEN, IGEN, and TOTAL SPACE columns:

```
ABM - OVERLAP BY AGGREGATE                                2008/05/19 06:51:32

           EVENT    DATE       TIME  TYPE LEVEL   GEN  IGEN TOTAL SPACE
---------- -------- ---------- ----- ---- ----- ----- ----- -----------
           DS7      2005/10/31 07:37 ACC  TAPE     -1            .00 b
           DEMO1    2005/10/31 05:33 ACC  TAPE      0            .00 b

           DS7      2005/10/31 07:37 ACC  TAPE     -1            .00 b
           DEMO1    2005/10/31 05:33 ACC  TAPE      0            .00 b

           DS7      2005/10/31 07:37 ACC  TAPE     -1            .00 b
```

Creating an online Data Set List report

This topic provides a procedure for creating an online Data Set List report, which displays information about every data set that matches the filter criteria you specify.

You can filter by data set name or mask, backup event name, volume name, and date. You can also specify for which relative generations you want information and restrict the report to a range of event dates, by data set level, and by backup type.

To generate an online data set list report:

1. Type 6 in the **Command** line on the **Main Menu** panel, and then press Enter to display the **Reports** panel.
2. Type 1 in the **Command** line on the **Reports** panel, and then press Enter to display the **ABM Backup Reports Menu** panel.
3. Type 3 in the **Command** line on the **ABM Backup Reports Menu** panel, and then press Enter to display the **ABM Dataset List Report** panel.
4. Use the **ABM Dataset List Report** panel's online help to guide you as you complete the fields on that panel.
5. Select the **Exec/View Online** option, and then press Enter to generate and display the report.

The following figure shows an example Data Set List report, which shows the search criteria in the upper left corner:

```
******************************** Top of Data ***************************
 SEARCH CRITERIA:
                DATASET => *
   DSN LEVEL TO DISPLAY => PRIMARY ML1 ML2 TAPE
 BACKUP TYPE TO DISPLAY => INCLUDE ALLOCATE ACCOMPANY

 PAGE 1                                        ABM - DATASET LIST

 DATASET NAME                                          EVENT    DATE       TIME
 ------------------------------------------------- -------- ---------- -----
 @CCKH25.SN1.ISPF.PROFILE                              DS11     2001/01/23 03:50

 @CCKH25.SP1.ISPF.PROFILE                              DS11     2001/01/23 03:50

 @CCKH25.ST1.ISPF.PROFILE                              DS11     2001/01/23 03:50

 @CCKH25.ST1.PROFILE.CLIST                             DS11     2001/01/23 03:50
```

Use the RIGHT function key to display the TYPE, LEVEL, GEN, IGEN, VOL, and TOTAL SPACE columns:

```
2 TAPE
TE ACCOMPANY

     ABM - DATASET LIST                                      2008/05/19 07:24:05

             EVENT    DATE       TIME  TYPE LEVEL   GEN  IGEN VOL(1) TOTAL SPACE
----------- -------- ---------- ----- ---- ----- ----- ----- ------ -----------
             DS11     2001/01/23 03:50 .... .....     0       ...... ............

             DS11     2001/01/23 03:50 .... .....     0       ...... ............

             DS11     2001/01/23 03:50 .... .....     0       ...... ............

             DS11     2001/01/23 03:50 .... .....     0       ...... ............
```

Creating an online Event List report

This topic provides a procedure for creating and viewing an online Event List report.

The Event List report displays information about every backup event that matches the filter criteria you specify. You can filter by event or logical aggregate name or mask. You can also specify for which relative generations you want information and restrict the report to a range of event dates and times.

1. Type 6 in the **Command** line on the **Main Menu** panel, and then press Enter to display the Reports panel.
2. Type 1 in the **Command** line on the **Reports** panel, and then press Enter to display the **ABM Backup Reports Menu** panel.
3. Type 4 in the **Command** line on the **ABM Backup Reports Menu** panel, and then press Enter to display the **ABM Event List Report** panel.
4. Use the **ABM Event List Report** panel's online help to guide you as you complete the fields on that panel.
5. Select the **Exec/View Online** option, and then press Enter to generate and display the report.

The following figure shows an example online Event List report. Notice the filter criteria in the upper left corner of the report:

```
****************************** Top of Data
**************************

 SEARCH CRITERIA:
        EVENT => I*
         SORT => EVENT

 PAGE 1                                        ABM - EVENT LIST

 EVENT      GEN  IGEN TYPE STATUS         RCA RCB DATE       TIME  ELAPSED
 -------- ----- ----- ---- -------------- --- --- ---------- ----- --------
 IAN          0        B   COMPLETE         0   0 2006/06/30 14:54 00:00:35

 IVP          0        B   COMPLETE         0  20 2003/09/12 13:49 **:**:**
             -1        B   COMPLETE         0  20 2003/09/11 15:03 **:**:**
             -2        B   COMPLETE         0  20 2003/07/30 15:09 **:**:**
             -3        B   COMPLETE         0  20 2003/06/02 14:24 **:**:**

 IVPACC       0        B   COMPLETE         0   0 2006/01/04 11:02 00:08:14
```

Use the RIGHT function key to display the **# DATASETS** and **TOTAL SPACE** columns:

```
     ABM - EVENT LIST                                  2008/05/19 07:30:08

 RCA RCB DATE       TIME  ELAPSED  # DSNS TOTAL SPACE
 --- --- ---------- ----- -------- ------ -----------
   0   0 2006/06/30 14:54 00:00:35      2   830.03 Kb

   0  20 2003/09/12 13:49 **:**:**      0      .00 b
   0  20 2003/09/11 15:03 **:**:**      0      .00 b
   0  20 2003/07/30 15:09 **:**:**      0      .00 b
   0  20 2003/06/02 14:24 **:**:**      0      .00 b

   0   0 2006/01/04 11:02 00:08:14      2    55.33 Kb
```

Creating and viewing an online Tape Pull List report

This topic provides a procedure for creating and viewing an online Tape Pull List report, which provides event name, generation, date, time, and data set name information for the volume types you specify.

The Tape Pull List report displays information about tape volumes and their data sets.

1. Type 6 in the **Command** line on the **Main Menu** panel, and then press Enter to display the **Reports** panel.
2. Type 1 in the **Command** line on the **Reports** panel, and then press Enter to display the **ABM Backup Reports Menu** panel.
3. Type 5 in the **Command** line on the **ABM Backup Reports Menu** panel, and then press Enter to display the **ABM Tape Pull List Report** panel.
4. Use the **ABM Tape Pull List Report** panel's online help to guide you as you complete the fields on that panel.
5. Select the **Exec/View Online** option, and then press Enter to generate and display the report.

The following figure shows an example online Tape Pull List report. Notice the filter criteria in the upper left corner of the report. The data in this example report has been sorted by volume, listing both volsers and data set names.

```
 SEARCH CRITERIA:
        EVENT => *
   GENERATION => 0,-9999
    DATE FROM => *
      DATE TO => *-999
       SORTBYVOL
    INCLUDEABARS

 PAGE 1                                    ABM - TAPE PULL LIST

 VOLSER DATASET NAME                                          EVENT      GEN  IGEN
 ------ ----------------------------------------------------- -------- ----- -----
 PROD04 PROD.PRODTLMS.TODISK.D.C01V0055                       PRODTLMS     0

 PROD07 PROD.PRODTEST.TODISK.D.C01V0005                       PRODTEST     0
 PROD10 PROD.PRODTLMS.TODISK.C.C01V0055                       PRODTLMS     0

 PROD3A PROD.PRODTEST.TODISK.O.C01V0005                       PRODTEST     0
 PROD39 PROD.PRODTEST.TODISK.C.C01V0005                                    0
```

Use the RIGHT function key to display the DATE and TIME columns. In the following example report, the data has been sorted by event name. Use the RIGHT function key to display the DATE and TIME columns.

```
 SEARCH
CRITERIA:

        EVENT =>
TEST1

   GENERATION =>
0,0


SORTBYNAME


INCLUDEABARS



 PAGE 1                                    ABM - TAPE PULL LIST

 EVENT      GEN  IGEN DATASET NAME                                    VOLSER
 -------- ----- ----- ---------------------------------------------- ------
 TEST1        0       J12.D.ABARS.TEST1.C.C01V0001                   T30283
              0       J12.D.ABARS.TEST1.D.C01V0001                   T30190



----------------------------

 TOTAL VOLUMES FOUND:
2
```

In this final example, the report data has been sorted by event name, but lists the volsers only.

```
 SEARCH
CRITERIA:

        EVENT =>
TEST1

   GENERATION =>
0,0


SORTBYVOL

    INCLUDEABARS

VOLUMEONLY


 PAGE 1                                        ABM - TAPE PULL LIST

 VOLSER
 ------
 T30190
 T30283

 ----------------------------
 TOTAL VOLUMES FOUND:       2
```

Creating an online Disaster Recovery Space Summary report

The Disaster Recovery Space Summary report provided by ABARS Manager summarizes space information for every backup event or aggregate that matches the filter criteria you specify.

You can filter by event or logical aggregate name or mask. You can also specify for which relative generations you want information and restrict the report to a range of event dates and times.

1. Type 6 in the **Command** line on the **Main Menu** panel, and then press Enter to display the **Reports** panel.
2. Type 1 in the **Command** line on the **Reports** panel, and then press Enter to display the **ABM Backup Reports Menu** panel.
3. Type 6 in the **Command** line on the **ABM Backup Reports Menu** panel, and then press Enter to display the **Disaster Recovery Space Summary Report** panel.
4. Use the **Disaster Recovery Space Summary Report** panel's online help to guide you as you complete the fields on that panel.
5. Select the **Exec/View Online** option, and then press Enter to generate and display the report.

The following figure shows an example Disaster Recovery Space Summary report for generations 0 and -1. Notice the search criteria shown in the upper left corner of the report.

```
******************************** Top of Data ********************************

 SEARCH CRITERIA:
        EVENT => **
   GENERATION => 0,-1
      BYEVENT => NO


 PAGE 1                          BKM - DISASTER RECOVERY SPACE SUMMARY

          DSN TYPE   # DSNS ALLOC SPACE  USED SPACE
          ---------- ------ ----------- -----------
          Primary      6256     1.15 Gb   860.87 Mb
          ML1          1144   758.66 Mb   176.24 Mb
          ML2            24     3.40 Mb     1.23 Mb
          User Tape     350
          Total        7774     1.89 Gb     1.01 Gb
****************************** Bottom of Data ******************************
```

Creating an online Overlap by Data Set Recovery report

The Overlap by Data Set Recovery report identifies data sets for which recovery overlaps exist; you can specify a data set name or mask for which to search, and you can explicitly include or exclude recover events.

1. Type 6 in the **Command** line on the **Main Menu** panel, and then press Enter to display the **Reports** panel.
2. Type 2 in the **Command** line on the **Reports** panel, and then press Enter to display the **ABM Recovery Reports Menu** panel.
3. Type 1 in the **Command** line on the **ABM Recovery Reports Menu** panel, and then press Enter to display the **ABM Overlap by Dataset Recovery Report** panel.
4. Use the **ABM Overlap by Dataset Recovery Report** panel's online help to guide you as you complete the fields on that panel.
5. Select the **Exec/View Online** option, and then press Enter to generate and display the report.

The following figure shows an example Overlap by Data Set Recovery report. Notice the filter criteria shown in the upper left corner of the report.

```
SEARCH CRITERIA:
          DATASET => D*

PAGE 1                 OVERLAP BY DATASET                  2005/05/05 15:34:04

DATASET NAME                                  EVENT    DATE       TIME     TYPE
--------------------------------------------- -------- ---------- -------- ------
DANR.DANR1.ISR0001.BACKUP                     P390BKP6 2004/06/11 15:59:55 DSS FVD
                                              P390BKP5 2004/06/08 20:57:52 DSS FVD

DANR.SI52.MULTIVOL                            ASTBJ610 2004/06/11 11:13:37 FDR FVD
                                              ASTBJ011 2004/06/10 10:12:42 DSS FVD
                                              ASTBJ010 2004/06/10 10:12:09 DSS FVD

DANR.SI52.MULTIVO2                            ASTBJ610 2004/06/11 11:13:37 FDR FVD
                                              ASTBJ011 2004/06/10 10:12:42 DSS FVD
                                              ASTBJ010 2004/06/10 10:12:09 DSS FVD
```

.

Appendix A. Authority requirements

This topic documents the authority requirements for ABARS Manager, Critical Backup Tracking and Inventory, Automated Critical Data Identification, and CATSCRUB.

The sections in this appendix provide the following information:

- The level or type of authority required by specific jobs or users to access the specified data sets or functions
- Information that allows you to create the security profiles that can be used to protect individual ABARS Manager primary or line commands
- **CATSCRUB** SAF profile security information

ABARS Manager required authorities, by data set or function

This topic describes the authorities required by users or jobs for ABARS Manager data sets and functions, organized according to data set or function.

The data set names in this topic use the *prefix.value* placeholder to represent one or more qualifiers.

Data set or function	Job or user	Required authority
Advanced Backup and Recovery for z/OS Inventory Data Set (IDS) *prefix.value*.IDS	All	Read/Write
Advanced Backup and Recovery for z/OS ISPF panel and messages library *prefix.value*.SBKMPENU	All	Read
Advanced Backup and Recovery for z/OS JCL library *prefix.value*.SBKMJCL	All	Read
Advanced Backup and Recovery for z/OS load module library *prefix.value*.SBKMLOAD	All	Read
Advanced Backup and Recovery for z/OS parameter library for configuration entries *prefix.value*.SBKMPARM	All	Read
Selection data sets	Any user	Read/Write
DFSMShsm ABARS activity logs HSMACT.*hostid.abarsfunction..agname*.D*yyyyddd*.T*hhmmss*	All	Read
DFSMShsm ABARS conflict data sets HSMUID.*agname*.CONFLICT.D*yyddd*.T*hhmmss*	All	Create/Delete
DFSMShsm ABARS restart data sets HSMUID.*agname*.RESTART.D*yyddd*.T*hhmmss*	All	Create/Delete
DFSMShsm ABARS skip data sets HSMUID.*agname*.ARCSKEXT.D*yyddd*.T*hhmmss*	All	Create/Delete

Data set or function	Job or user	Required authority
DFSMShsm backup control data set HSMUID.BCDS	All	Read
DFSMShsm migration control data set HSMUID.MCDS	All	Read

Incremental ABARS required authorities, by data set or function

This topic describes the authorities required for incremental ABARS users or jobs, organized according to data set or function.

Data set or function	User or job	Authority
Incremental ABARS DAD *prefix.value*.DAD	Any user BKMSTSKD BKM9254M BKMDDAD	Read/Write Read/Write Read/Write Create/Delete
Incremental ABARS DAD message log data set *prefix.value*.LOG	Any user BKMDLOG BKMSTSKD	Read Create/Delete Read/Write
Incremental ABARS DAD message queue data set *prefix.value*.MSGQ	Any user BKMDMSGQ BKMSTSKD	Read Create/Delete Read/Write
Incremental ABARS DAD overflow queue data set *prefix.value*.OVERFLOW	Any user BKMSTSKD	Read/Write Read/Write

Incremental ABARS required authorities, by user or job

This topic describes the authorities required for incremental ABARS data sets or functions, organized by user or job.

Job or user	Data set or function	Authority
Incremental ABARS BKMSTSKD	DAD DAD message log data set DAD message queue data set DAD overflow data set	Read/Write Read/Write Read/Write Read/Write
Incremental ABARS BKMDDAD	DAD	Create/Delete
Incremental ABARS BKM9254M	DAD	Create/Delete

Job or user	Data set or function	Authority
Incremental ABARS BKMDLOG	DAD message log data set	Create/Delete
Incremental ABARS BKMDMSGQ	DAD message queue data set	Create/Delete
Any user	Selection data sets DAD DAD message log data set DAD message queue data set DAD overflow data set	Read/Write Read/Write Read/Write Read/Write Read/Write

Critical Backup Tracking and Inventory required authorities, by data set or function

This topic describes the authorities required by jobs or users to access the Critical Backup Tracking and Inventory data sets and functions.

Data set or function	Job or user	Authority
Advanced Backup and Recovery for z/OS IDS *prefix.value*.IDS	All	Read/Write
Advanced Backup and Recovery for z/OS panel and messages library *prefix.value*.ISPPLIB	All	Read
Advanced Backup and Recovery for z/OS JCL library *prefix.value*.JCL	All	Read
Advanced Backup and Recovery for z/OS load module library *prefix.value*.LOAD	All	Read
Advanced Backup and Recovery for z/OS parameter library for product licensing and configuration entries *prefix.value*.PARMLIB	All	Read
Critical Backup Tracking and Inventory message log data set *prefix.value*.MSGLOG	Any user BKMSETUP BKMSTSKA	Read/Write Create/Delete Read/Write
Critical Backup Tracking and Inventory message queue data set *prefix.value*.MSGQUEUE	Any user BKMSETUP BKMSTSKA	Read/Write Create/Delete Read/Write

Data set or function	Job or user	Authority
Critical Backup Tracking and Inventory intercept file *prefix.value*.BIF	Any user BKMSETUP BKMSTSKA	Read/Write Create/Delete Read/Write
Critical Backup Tracking and Inventory flat file data set Sequential flat file CBTI.flatfile.*prefix.value*.D*yyyyddd*.T*hhmmsst*	All	Create/Delete
Critical Backup Tracking and Inventory VSAM flat file CBTI.flatfile.*vsamds*.	All	Alter
DFSMShsm backup control data set HSMUID.BCDS	All	Read
DFSMShsm migration control data set HSMUID.MCDS	All	Read

Automated Critical Data Identification required authorities, by data set or function

The following table describes the authorities required for jobs and users accessing Automated Critical Data Identification data sets and functions.

Data set or function	Job or user	Authority
Automated Critical Data Identification ISPF panel and messages library *prefix.value*.ISPPLIB	All	Read
Automated Critical Data Identification JCL library *prefix.value*.JCL	All	Read
Automated Critical Data Identification load module library *prefix.value*.LOAD	All	Read
Automated Critical Data Identification execution library *prefix.value*.PACKED	All	Read
Automated Critical Data Identification parameter library for product authorization and configuration entries *prefix.value*.PARMLIB	All	Read
Automated Critical Data Identification application reference files	Any user	Read/Write

Data set or function	Job or user	Authority
Automated Critical Data Identification database	Any user	Read/Write
	BKMAPLEN	Read/Write
	BKMMNT01	Read/Write
	BKMAPPCP	Read/Write
	BKMBLDJB	Read/Write
	BKMJRFSH	Read/Write
	BKMJSWAP	Read/Write
	BKMSTSKR	Read/Write
RSP job flat file	BKMBLDJB	Read/Write
prefix.value.JOBFLAT	BKMJRFSH	Read/Write
	BKMRSPSU	Read/Write
	BKMSTSKR	Read/Write
RSP message log data set	Any user	Read
prefix.value.MSGLOG	BKMAPLEN	Read/Write
	BKMRSPSU	Create/Delete
	BKMSTSKR	Read/Write
RSP message queue data set	Any user	Read/Write
prefix.value.MSGQUEUE	BKMAPLEN	Read/Write
	BKMJRFSH	Read/Write
	BKMJSWAP	Read/Write
	BKMFILTR	Read/Write
	BKMSTSKR	Read/Write
RSP overflow data set	Any user	Read/Write
prefix.value.OVERFLOW	BKMSTSKR	Read/Write
Selection data set	Any user	Read/Write
	BKMAPLEN	Read/Write

Automated Critical Data Identification required authorities, by job or user

The following table describes the authorities required to access the Automated Critical Data Identification data sets or functions, organized according to the job or user.

Job or user	Data set or function	Authority
Any user	Automated Critical Data Identification application reference files Automated Critical Data Identification database *prefix.value*.MSGLOG *prefix.value*.MSGQUEUE Selection data set	Read/Write Read/Write Read Read/Write Read/Write
BKMAPLEN	Automated Critical Data Identification database *prefix.value*.MSGLOG *prefix.value*.MSGQUEUE Selection data set	Read/Write Read/Write Read/Write Read/Write
BKMMNT01	Automated Critical Data Identification database	Read/Write
BKMAPPCP	Automated Critical Data Identification database	Read/Write
BKMJRFSH	Automated Critical Data Identification database *prefix.value*.JOBFLAT	Read/Write Read/Write
BKMJSWAP	Automated Critical Data Identification database *prefix.value*.MSGQUEUE	Read/Write Read/Write
BKMRSPSU	*prefix.value*.MSGLOG *prefix.value*.JOBFLAT	Create/Delete Create/Delete
BKMSTSKR	*prefix.value*.MSGLOG *prefix.value*.MSGQUEUE *prefix.value*.OVERFLOW Automated Critical Data Identification database *prefix.value*.JOBFLAT	Read/Write Read/Write Read/Write Read/Write Read
BKMFILTR	*prefix.value*.MSGQUEUE	Read/Write

Authorities required for ISPF panel security

You can control user access to IBM Tivoli Advanced Backup and Recovery for z/OS ISPF panels and to primary and line command execution by defining SAF class profiles that are recognized by RACF, ACF2, and Top Secret.

Product functionality covered by ISPF panel security

You can apply ISPF panel security to the following product functionality:

- **Aggregate Management** functions accessible from the **Main Menu** panel:
  - The **Aggregate Management with SMS Interface** option available from the **Setup and Configuration** menu
  - The **ABARS Events** and **Logical Aggregates** options available from the **Backup and Recovery Management** menu.
- **DAD Jobname Management** function available through the **Setup and Configuration** option from the **Main Menu** panel:
- **User Defined Backup Utilities Management** function available through the **Setup and Configuration** option from the **Main Menu** panel

Considerations

If your computing environment is one that defaults to *not* protecting all resources and you do not want to implement security for these features, no security setup is required.

However, if your computing environment defaults to protecting all resources, you must set up security profiles or rules to allow users full access to the DAD Jobname Management, User Defined Backup Utilities Management, and Aggregate Management panel features. In this type of computing environment, specifying a security profile or rule granting update access to `IBM.TIVOLI.ABR.**` allows the use of primary and line commands on these ISPF panels.

Quick setup

If your computing environment protects all resources by default, you can grant access to the product functionality describing in Product functionality covered by ISPF panel security section by a security profile or rule that grants update access to *`prefix.value`*`.BKM.**`.

DAD Jobname Management ISPF security protection

Primary command	Description
(CAN)CEL	Leave the DAD Jobname Filter Table ISPF option without saving records.
DATED	Changes the date display format to DD/MM/YYYY.
DATEY	Changes the date display format to YYYY/MM/DD.
END(PF3)	Leave the DAD Jobname Filter Table ISPF option after saving the Jobname Filter Table.
(REF)RESH	Refresh the DAD Jobnames Filter Table by reloading it from the IDS.
SAVE	Save the DAD Jobnames Filter Table to the IDS. When SAVE is invoked, the Jobname Filters in the table are written to the IDS, overlaying the DAD Jobname Filters already stored in the IDS.
SORTA	Causes the DAD Jobname Filter Table to be displayed in ascending Jobname order.

Primary command	Description
SORTD	Causes the DAD Jobname Filter Table to be displayed in descending Jobname order.

Line command	Description
D	Deletes a table entry from the DAD Jobnames Filter Table. If the SAVE primary command is subsequently issued, those deleted table entries are deleted from the IDS.
I	Inserts a blank record in the DAD Jobnames Filter Table, with the intention of entering a DAD Jobname Filter. If nothing is entered in the Jobname column and the SAVE command is issued subsequently, the blank entries are not saved in the IDS.
R	Repeats an existing DAD Jobnames Filter Table entry, with the intention of updating the duplicate entry. If the duplicate Jobname column is not modified, and the SAVE command is issued subsequently, the original table entry is saved in the IDS but the duplicate table entry is not saved in the IDS.

The following PROFILE and FACILITY Class detail is provided for each command:

```
In the DAD Job Management Panel
*
ACCESS TO THE DAD JOB MANAGEMENT ISPF PANEL
PROFILE(IBM.TIVOLI.ABR.DADJBFLT.ACCESS.**) CLASS(FACILITY)
ACCESS OF NONE RESTRICTS ACCESS TO THE DAD JOB MANAGEMENT ISPF PANEL.
ACCESS OF UPDATE ALLOWS ACCESS TO THE DAD JOB MANAGEMENT ISPF PANEL.
*
LINE COMMAND D
PROFILE(IBM.TIVOLI.ABR.DADJBFLT.DELETE.**)CLASS(FACILITY)
ACCESS OF NONE RESTRICTS LINE COMMAND D
ACCESS OF UPDATE ALLOWS ACCESS TO LINE COMMAND D
*
LINE COMMANDS I AND R
PROFILE(IBM.TIVOLI.ABR.DADJBFLT.ADD.**)CLASS(FACILITY)
ACCESS OF NONE RESTRICTS LINE COMMANDS I AND R
ACCESS OF UPDATE ACCESS ALLOWS ACCESS TO LINE COMMANDS I AND R
*
PRIMARY COMMAND SAVE
PROFILE(IBM.TIVOLI.ABR.DADJBFLT.SAVE.**) CLASS(FACILITY)
ACCESS OF NONE RESTRICTS PRIMARY COMMAND SAVE
ACCESS OF UPDATE ACCESS ALLOWS ACCESS TO PRIMARY COMMAND SAVE
*
```

User Defined Backup Utilities Management ISPF security protection

Primary command	Description
SAVE	Save the User Defined Backup Utilities Table to the IDS.

Line command	Description
D	Deletes a User Defined Backup Utility record from the User Defined Backup Utility Table.
I	Adds a blank record to the User Defined Backup Utility Table, with the intention of adding a User Defined Backup Utility record.
R	Repeats an existing User Defined Backup Utility Table entry, with the intention of updating the duplicate record to become a new record.

The following PROFILE and FACILITY Class detail is provided for each command:

```
TO ALLOW/DENY ACCESS TO USER DEFINED BACKUP UTILITIES MANAGEMENT ISPF OPTION
PROFILE(IBM.TIVOLI.ABR.USRDEFUT.ACCESS.**) CLASS(FACILITY)
ACCESS OF NONE RESTRICTS ACCESS TO DEFINED BACKUP UTILITIES MANAGEMENT
ACCESS OF UPDATE ALLOWS ACCESS TO DEFINED BACKUP UTILITIES MANAGEMENT
*
PRIMARY COMMAND SAVE
PROFILE(IBM.TIVOLI.ABR.USRDEFUT.SAVE.**) CLASS(FACILITY)
ACCESS OF NONE RESTRICTS PRIMARY COMMAND SAVE
ACCESS OF UPDATE ACCESS ALLOWS ACCESS TO PRIMARY COMMAND SAVE
*
LINE COMMAND D
PROFILE(IBM.TIVOLI.ABR.USRDEFUT.DELETE.**) CLASS(FACILITY)
ACCESS OF NONE RESTRICTS LINE COMMAND D
ACCESS OF UPDATE ACCESS ALLOWS ACCESS TO LINE COMMAND D
*
LINE COMMANDS I and  R
PROFILE(IBM.TIVOLI.ABR.USRDEFUT.ADD.**) CLASS(FACILITY)
ACCESS OF NONE RESTRICTS LINE COMMANDS  I and R
ACCESS OF UPDATE ACCESS ALLOWS ACCESS TO LINE COMMANDS  I and R
*
```

Aggregate Management ISPF security protection

You can control user access to the following ABARS Manager primary and line commands by creating SAF profiles. Be aware that unless you explicitly specify SAF protections, all of these commands are accessible to all users.

Primary command	Description
ADD	Adds a new logical aggregate

Line command	Description
A	Adds an aggregate to the IDS.
B	Backs up an aggregate.
C	Changes an aggregate.
CL	Changes SDSL for an aggregate.
D	Deletes an aggregate from the IDS.
DE	Deletes an ABARS event record.
D#	Deletes an ABARS event record and all associated records. **Important:** Incorrect use of this command could result in a loss of data. Because of this potential danger, use of this command should be restricted to storage administrators only.
EL	Edits the SDSL for an aggregate.
R	Submits an ARECOVER for an aggregate.

The following PROFILE and FACILITY Class detail is provided for each command:

```
In the Aggregate Management option of the Setup and Configuration panel
*
LINE COMMANDS A and C
PROFILE(IBM.TIVOLI.ABR.AGGMNG.MANAGE.**)CLASS(FACILITY)
ACCESS OF NONE RESTRICTS LINE COMMANDS A AND C
ACCESS OF UPDATE ALLOWS ACCESS TO LINE COMMANDS A AND C
*
LINE COMMANDS B
PROFILE(IBM.TIVOLI.ABR.BACKUP.**)CLASS(FACILITY)
ACCESS OF NONE RESTRICTS LINE COMMAND B
ACCESS OF UPDATE ACCESS ALLOWS ACCESS TO LINE COMMAND B
*
LINE COMMANDS CL AND EL
PROFILE(IBM.TIVOLI.ABR.AGGMNG.EDITSDSL.**) CLASS(FACILITY)
ACCESS OF NONE RESTRICTS THE LINE COMMANDS CL AND EL
ACCESS OF UPDATE ALLOWS ACCESS TO LINE COMMANDS CL AND EL
*
LINE COMMANDS D
PROFILE(IBM.TIVOLI.ABR.AGGMNG.DELETE.**) CLASS(FACILITY)
ACCESS OF NONE RESTRICTS LINE COMMAND D
ACCESS OF UPDATE ACCESS ALLOWS ACCESS TO LINE COMMAND D
*
```

```
In the ABARS Events option of the Backup & Recovery Management panel
LINE COMMANDS DE
PROFILE(IBM.TIVOLI.ABR.EVENTMNG.DELETE.**) CLASS(FACILITY)
ACCESS OF NONE RESTRICTS LINE COMMAND DE
ACCESS OF UPDATE ACCESS ALLOWS ACCESS TO LINE COMMAND DE
*
LINE COMMANDS D#
PROFILE(IBM.TIVOLI.ABR.EVENTMNG.DELPND.**) CLASS(FACILITY)
ACCESS OF NONE RESTRICTS LINE COMMAND D#
ACCESS OF UPDATE ACCESS ALLOWS ACCESSS TO LINE COMMAND D#
*(Very dangerous command and should be restricted to the Storage Administrator only).
*
LINE COMMANDS R
PROFILE(IBM.TIVOLI.ABR.RECOVER.**) CLASS(FACILITY)
ACCESS OF NONE RESTRICTS LINE COMMAND R
ACCESS OF UPDATE ACCESS ALLOWS ACCESS TO LINE COMMAND R
```

```
In the Logical Aggregates option
of the Backup & Recovery Management panel

PRIMARY COMMANDS ADD
PROFILE(IBM.TIVOLI.ABR.LAGGMNG.ADD) CLASS(FACILITY)
ACCESS OF NONE RESTRICTS PRIMARY COMMAND ADD
ACCESS OF UPDATE ACCESS ALLOWS ACCESS TO PRIMARY COMMAND ADD
*
LINE COMMANDS B
PROFILE(IBM.TIVOLI.ABR.LAGGMNG.BKP) CLASS(FACILITY)
ACCESS OF NONE RESTRICTS LINE COMMAND B
ACCESS OF UPDATE ACCESS ALLOWS ACCESS TO LINE COMMAND B
*
LINE COMMANDS A
PROFILE(IBM.TIVOLI.ABR.LAGGMNG.CHG) CLASS(FACILITY)
ACCESS OF NONE RESTRICTS LINE COMMAND A
ACCESS OF UPDATE ACCESS ALLOWS ACCESS TO LINE COMMAND A
*
LINE COMMANDS DL AND DA
PROFILE(IBM.TIVOLI.ABR.LAGGMNG.DEL) CLASS(FACILITY)
ACCESS OF NONE RESTRICTS LINE COMMANDS DL AND DA
ACCESS OF UPDATE ACCESS ALLOWS ACCESS TO LINE COMMAND DL AND DA
*
LINE COMMANDS R
PROFILE(IBM.TIVOLI.ABR.LAGGMNG.RECOVER) CLASS(FACILITY)
ACCESS OF NONE RESTRICTS LINE COMMAND R
ACCESS OF UPDATE ACCESS ALLOWS ACCESS TO LINE COMMAND R
```

SAF class profiles for the Critical Backup Tracking and Inventory started task

If you have specified FACILITY profile for CSVDYLPA.ADD.* , provide UPDATE access to allow the Critical Backup Tracking and Inventory started task, BKMSTSKA, to load its modules into DLPA. If profile CSVDYLPA.ADD.* has not been specified, you do not need to do anything.

SAF class profiles for controlling CATSCRUB command execution

You can use SAF profiles to control user access to CATSCRUB command execution. These profiles are recognized by RACF, ACF2, and Top Secret.

Because CATSCRUB command execution has the potential to be destructive, the SAF profiles are established at the command level. The SAF class profile is as follows:

Profile	Class	Access required
IBM.TIVOLI.CATSCRUB	Facility	READ

To protect the CATSCRUB command in batch mode, you can choose to define the profile. If no profile is found, access is granted to all users and is then determined by the user's authority to access the ICF user catalog.

- For each user catalog that is selected by user keywords, CONTROL access is checked using the catalog's VOLSER.
- If a catalog cannot be located or it does not have CONTROL access, a status message is issued and that catalog is not scrubbed.

Appendix B. BKMINI configuration values

Use the BKMINI product parameter library member to define the global parameters that affect Advanced Backup and Recovery for z/OS usage and options available within your installation.

Coding the BKMINI keyword values

You can use an ISPF EDIT session to specify BKMINI keyword values.

Be aware of the following guidelines as you specify values for BKMINI keywords (or tokens):

- Some BKMINI keywords cannot be modified. This appendix documents all modifiable BKMINI keywords.
- Do not delete from BKMINI or modify any of the BKMINI keywords for software that is not installed at the client site.

Keywords are documented in the following format:

```
Keyword_Name = assigned-value [ + ] [ /* Comments... */ ]
```

Example keyword statements

The following examples show valid keyword assignments and keyword statement format:

```
TOKEN1 = VALUE1                /* Comment1... */
TOKEN2 = 'IN QUOTES'
TOKEN3 = THIS TOKEN VALUE IS  +
         CONTINUED ACROSS     +
         MULTIPLE LINES
```

Modifying data set name high order nodes

Data set names in the BKMINI product parameter library member are shown with the first and second high order nodes in format *prefix.value*.V*vv*R*rr*.

Where:

- *prefix.value* is the high level qualifier string specified during installation
- V is a constant.
- *vv* is the 2-digit product version level.
- R is a constant.
- *rr* is the 2-digit product release level.

You can use this generic structure with change commands to modify the BKMINI product parameter library member.

:INIMERGE_VALUES section keyword

The keywords in the :INIMERGE_VALUES section of the BKMINI member of the product parameter library are described in this topic.

SPECIAL_SECTIONS
Do not change the value already assigned to this keyword. If you think you need to change this value, contact Technical Support for assistance.

:PRODUCT_INFO section keywords

Use the information in this topic to specify the values for the keywords in the :PRODUCT_INFO section of the BKMINI member of the product parameter library.

BKM_REL=V*v.r*
This keyword specifies the version and release level for your Advanced Backup and Recovery for z/OS installation. You do not need to modify the value assigned to this keyword.

BKM_REL_DATE=*dd mmm yyyy*
This keyword specifies the release date for Advanced Backup and Recovery for z/OS. You do not need to modify the value assigned to this keyword.

BKM_TITLE1=V*v.r*
The value assigned to this keyword is used to display product release level information on its ISPF panels. You do not need to change the value assigned to this keyword.

BKM_TITLE2='Copyright (C) *yyyy* Rocket Software, Inc.'
The value assigned to this keyword is used to display copyright information on the product's ISPF panels. You do not need to change the value assigned to this keyword.

BKM_TITLE3='All Rights Reserved.'
The value assigned to this keyword is used to display copyright information on the product's ISPF panels. You do not need to change the value assigned to this keyword.

:PRODUCT_SELECTABLE_UNITS section keywords

Use the information in this section to specify the values for the keywords in the :PRODUCT_SELECTABLE_UNITS section of the BKMINI member in the product parameter library. These keywords indicate which product functions are in use.

ACDI_ACTIVE=YES|NO
The value assigned to this keyword indicates whether the Automated Critical Data Identification function is in use.

CBTI_ACTIVE=YES|NO
The value assigned to this keyword indicates whether the Critical Backup Tracking and Inventory function is in use.

ABM_ACTIVE=YES|NO
The value assigned to this keyword indicates whether the ABARS Manager function is in use.

INC_ACTIVE=YES|NO
The value assigned to this keyword indicates whether the Incremental ABARS function is in use.

CATSCRUB_ACTIVE=YES|NO
The value assigned to this keyword indicates whether the CATSCRUB function is in use.

FRS_ACTIVE=YES|NO
The value assigned to this keyword indicates whether the Fast Replication Services function is in use.

CA_DISK_ACTIVE=YES|NO
Use this keyword to specify whether Critical Backup Tracking and Inventory is tracking CA-Disk backups.

:INSTALLATION_DATASETS section keywords

Use the information in this section to specify appropriate values for the keywords in the :INSTALLATION_DATASETS section of member BKMINI in the product parameter library. The data set names you specify must match those chosen during product installation and configuration.

Names are fully qualified without quotes.

LOAD1 = ?.SBKMLOAD
This library must be APF authorized.

Note: If your system PARMLIB member IEASYS00 has parameter LNKAUTH=LNKLST specified, Advanced Backup and Recovery for z/OS does not honor it. The product LOAD library must be APF authorized in either the IEAPFxx or the PROGxx member.

The name for the product LOAD data set you specify here must match the name you chose when the product was installed. If possible, keep the LOAD qualifier.

ISPPLIB1 = ?.SBKMPENU
This library contains ISPF panels and messages.

ISPTLIB1 = ?.SBKMTENU
This library contains the tables used by the Critical Backup Tracking and Inventory product interface.

ISPMLIB1 = ?.SBKMMENU
This library contains the ISPF messages issued through the product's ISPF interface.

PARMLIB = ?.SBKMPARM
This library contains BKMINI variables, the startup REXX EXEC, product messages, and so forth. The name for the product parameter data set you specify here must match the name chosen when the product was installed.

MSGLIB = ?.SBKMMSGS
This library contains the Advanced Backup and Recovery for z/OS messages. The name you specify for the product message data set you specify here must match the name chosen when the product was installed.

SKELLIB = ?.SBKMSKEL
This library contains the default skeleton restore JCL used during restores of backups tracked by Critical Backup Tracking and Inventory. The name for the product SKELLIB data set you specify here must match the name chosen when the product was installed.

PACKED = ?.SBKMEXEC
Product PACKED data set name. The name you specify for this keyword must match the name chosen when the product was installed.

JCL = ?.SBKMJCL
Product JCL data set name. The name you specify for this keyword must match the name chosen when the product was installed. Keep the JCL qualifier if possible. JCL members must be tailored by running BKMJCLBL.

CBTI_USE_LINKLIST=Y|N
The default value for this keyword is N. Specify Y to indicate that the Critical Backup Tracking and Inventory LOADLIB is in the LNKLST. If LNKLST is used for one system , it should be used for all systems. You must remember to refresh LLA (Library Lookaside) after you apply Advanced Backup and Recovery for z/OS maintenance.

:PRODUCT_DATASETS section keywords

Use the information in this topic to specify values for the keywords in the :PRODUCT_DATASETS section of the BKMINI member in the product parameter library; this section defines data sets created during installation for use by the product. The data set names you specify here must match to names chosen during data set creation.

Names are fully qualified without quotes.

BKLEVEL=NULLFILE
This is the default BKLEVEL data set name used with Prevent Backlevel. *Do not specify a value for this keyword* until necessary during disaster recovery.

INVENTORY_DATASET=NULLFILE
This keyword specifies the name of a VSAM KSDS structure that serves as the Advanced Backup and Recovery for z/OS Inventory Data Set (IDS).

The name you specify here for the IDS must match the name chosen when the IDS was created during installation with JCL member BKMDEFDB.

DAD_DATASET=NULLFILE
Name of the Data Set Activity Database (DAD). The value you specify here for the DAD must match the name chosen when the DAD was defined using the JCL member BKMDDAD.

ADABAS_LOADLIB=LINKLST
ADABAS_LOADLIB is required to build restore jobs for Adabas backups. The default is LINKLST, which means that the Adabas LOAD library is included in the LINKLIST. If that is not the case, specify the data set name of the Adabas LOAD library.

CR+LOADLIB=LINKLST
The default is LINKLIST, which means that the CR+ LOAD library is included in the LINKLIST. If that is not the case, specify the data set name of the CR+ LOAD library.

CR+PARMLIB=NULLFILE
Specify the data set name of the CR+ PARMLIB library.

FAVER_LOADLIB=LINKLST
FAVER_LOADLIB is required to build restore jobs for Brightstor CA-FAVER backups. The default is LINKLST, which means that the Brightstor CA-FAVER LOAD library is included in the LINKLIST. If that is not the case, specify the data set name of the Brightstor CA-FAVER LOAD library.

VASST_LOADLIB=LINKLST
VASST_LOADLIB is required to build restore jobs for VSAM Assist backups. The default is LINKLST, which means that the VSAM Assist LOAD library is included in the LINKLIST. If that is not the case, specify the data set name of the VSAM Assist LOAD library.

FDR_LOADLIB=LINKLST
The default is LINKLST, which means the FDR LOAD library is included in the LINKLIST. If that is not the case, specify the data set name of the FDR LOAD library.

CADISK_LOADLIB=LINKLST
CADISK_LOADLIB is required to build restore jobs for CA Disk backups. The default is LINKLST, which means that the CA Disk LOAD library is included in the LINKLIST. If that is not the case, specify the data set name of the CA Disk LOAD library.

CADISK_FILES=DSName|NULLFILE
This is a required keyword for building restore jobs for CA Disk backups. The FILES data set is the data base containing the CA Disk backup and archive inventory. Use this keyword to specify the CA Disk inventory data base if you are using Critical Backup Tracking and Inventory to track CA Disk backups.

CADISK_PARMLIB=DSName|LINKLST
This is a required keyword for building restore jobs for CA Disk backups. LINKLST means the CA Disk PARM library is included in the LINKLIST. If that is not the case, specify the data set name for the CA Disk PARM library.

CADISK_ARCHVOL_VERIFY=NO|YES
Specify YES for this keyword when multiple CA Disk tracked backup jobs might be running concurrently. When you specify YES, additional validity checking is performed by verifying that the DSNINDEX record backup volume information is the same as the backup output volume, which causes a slight increase in flat file processing time.

Leave this keyword set to the default NO unless you are otherwise directed by Customer Support.

JCLOUT=DSName|NULLFILE
This is the default JCLOUT data set name that is used when building restore JCL.

ACDIDB=DSNAME|NULLFILE
The *DSName* value is the name to be used to create the Automated Critical Data Identification VSAM database. This file will be used by the Automated Critical Data Identification facility. If the JCL member BKMDEFN will be used to create the Automated Critical Data Identification VSAM file, then the name that will be used must be supplied in this field. No default value is supplied.

HDU_DSN=DSNAME
Use this keyword to specify the data set name you want used when JCL member BKMDHDU is run to create the Historical Dataset Usage database.

Note: This token is ignored unless token RSP_Start_HDU_Task=YES.

FRZ_CNTL_FILE_DSN=NULLFILE
If you are using Fast Replication Services, use this keyword to specify the Fast Replication Services control library. The default value for this keyword is NULLFILE.

FRZ_LOADLIB=NULLFILE
If you are using Fast Replication Services, use this keyword to specify the Fast Replication Services LOAD library. The default value for this keyword is SYS1.LINKLIB.

FRZ_ISPPLIB=NULLFILE
If you are using Fast Replication Services, use this keyword to specify the Fast Replication Services ISPF panels and messages library. The default value for this keyword is NULLFILE.

FRZ_ISPMLIB=NULLFILE
If you are using Fast Replication Services, use this keyword to specify the Fast Replication Services ISPF messages library. The default value for this keyword is NULLFILE.

FRZ_MSGLIB=NULLFILE
If you are using Fast Replication Services, use this keyword to specify the Fast Replication Services messages library. The default value for this keyword is NULLFILE.

:RESOURCE_SERIALIZATION section keywords

Use the information in this topic to specify values for the keywords in the :RESOURCE_SERIALIZATION section of the BKMINI member in the product parameter library.

Installations running CA-MIM/MII with multiple systems and shared DASD need to set the following parameter to YES to ensure that when CA-MIM/MII GDIF is inactive, the Advanced Backup and Recovery for z/OS data sets are protected from data corruption.

MIM_GDIF=NO|YES
The default value for this keyword is NO.

If you have CA-MIM/MII with multiple systems and shared DASD, this token should be changed to YES to ensure that when CA-MIM/MII GDIF is inactive, the Advanced Backup and Recovery for z/OS data sets are protected from data corruption.

:HSM section keywords

Use the information in this topic to specify values for the keywords in the :HSM section of the BKMINI member of the product parameter library.

The :HSM section keywords define the DFSMShsm data sets accessed by Advanced Backup and Recovery for z/OS.

MCDS=*DSName*
Use this keyword to specify the fully qualified name of your installation's first or only DFSMShsm migration control data set.

MCDS2=*DSName*
If your installation is using split MCDSs, use this keyword to specify the fully qualified name of your installation's second DFSMShsm migration control data set.

MCDS3=*DSName*
If your installation is using split MCDSs, use this keyword to specify the fully qualified name of your installation's third DFSMShsm migration control data set.

MCDS4=*DSName*
If your installation is using split MCDSs, use this keyword to specify the fully qualified name of your installation's fourth DFSMShsm migration control data set.

BCDS=*DSName*
Use this keyword to specify the fully qualified name of your installation's first or only DFSMShsm backup control data set.

BCDS2=*DSName*
If your installation uses split control data sets, use this keyword to specify the fully qualified name of your installation's second DFSMShsm backup control data set.

BCDS3=*DSName*
If your installation uses split control data sets, use this keyword to specify the fully qualified name of your installation's third DFSMShsm backup control data set.

BCDS4=*DSName*
If your installation uses split control data sets, use this keyword to specify the fully qualified name of your installation's fourth DFSMShsm backup control data set.

:PROCESS_OPTIONS_ISPF section keywords

Use the information in this topic to specify values for the keywords in the :PROCESS_OPTIONS_ISPF section of the BKMINI member (in the product parameter library), which control the appearance of the Automated Critical Data Identification ISPF panels or how options function on the ISPF panels.

ALT_MAIN_MENU=(blank)|newpanelname
By default, no value is specified for this keyword. Specify a new panel name if you want to create a customized main menu. The new name you specify must be a Select Services panel that resides in the Automated Critical Data Identification ISPPLIB.

RESTRICT_HRECOVER_REPLACE=Y|N
The default value for this keyword is N. Specify Y to restrict the **Replace** field value on the **HRECOVER Options** panel so that the value cannot be changed and will be used during the HRECOVER process. Specify N if you want to allow users to change the **Replace** field value on the **HRECOVER Options** panel before they submit an HRECOVER.

RESTRICT_HRECOVER_HSEND=Y|N
The default value for this keyword is N. Specify Y to restrict the **HSEND** field value on the HRECOVER Options panel so that the value cannot be changed and will be used during the HRECOVER process. Specify N if you want to allow users to change the **HSEND** field value on the HRECOVER Options panel before they submit an HRECOVER.

RESTRICT_HRECOVER_NONSMS=Y|N
The default value for this keyword is N. Specify Y to restrict the **NON-SMS** field value on the **HRECOVER Options** panel so that the value cannot be changed and will be used during the HRECOVER process. Specify N if you want to allow users to change the **NON-SMS** field value on the **HRECOVER Options** panel before they submit an HRECOVER.

RESTRICT_HRECOVER_TOVOLUME=Y|N
The default value for this keyword is N. Specify Y to restrict the **ToVolume** field value on the **HRECOVER Options** panel so that the value cannot be changed and will be used during the HRECOVER process. Specify N if you want to allow users to change the **ToVolume** field value on the **HRECOVER Options** panel before they submit an HRECOVER.

RESTRICT_HRECOVER_UNIT=Y|N
The default value for this keyword is N. Specify Y to restrict the **Unit** field value on the **HRECOVER Options** panel so that the value cannot be changed and will be used during the HRECOVER process. Specify N if you want to allow users to change the **Unit** field value on the **HRECOVER Options** panel before they submit an HRECOVER.

DEFAULT_HRECOVER_REPLACE=Y|N
The default value for this keyword is N. Specify Y to have the software display Y in the **Replace** field on the **HRECOVER Options** panel. Specify N to have the software display N in the **Replace** field value on the **HRECOVER Options** panel.

HRECOVER_NEWNAME_HLQ=newhlq
Specify 1 to 8 characters for the high level qualifier. The value of the BKMINI member's HRECOVER_NEWNAME_APPEND keyword determines whether this new high level qualifier is appended to or replaces the high level qualifier for the data set selected for recovery.

HRECOVER_NEWNAME_APPEND=Y|N
The default value for this keyword is Y.

Specify Y if you want the new high level qualifier specified for the HRECOVER_NEWNAME_HLQ keyword to be appended to the HSM data set selected for recovery.

Specify N if you want the new high level qualifier specified for the HRECOVER_NEWNAME_HLQ keyword to replace the high level qualifier for the HSM data set selected for recovery.

SKIP_RESTORE_REPLACE=Y|N
The default value for this keyword is N. The value you specify for this keyword determine whether the **Replace Existing Datasets** window pops up during a Quick Restore.

Specify Y if you do not want the pop-up window to appear during a Quick Restore. The value of the **Replace** field on the **HRECOVER Options** panel defaults to the value you specified for the DEFAULT_RESTORE_REPLACE keyword in BKMINI.

Specify N if you want the pop-up window to appear during a Quick Restore. The user can then use the pop-up window to specify a Replace option.

DEFAULT_RESTORE_REPLACE=Y|N
The default value for this keyword is N.

Specify Y if you do want the data sets replaced during a Quick Restore.

Specify N if you do *not* want the data sets replaced during a Quick Restore. .

RESTORE_DEFAULT=Q|S|(blank)
By default, there is no value assigned to this keyword. This keyword determines which windows appear during the restore process.

Specify Q if you want the software to perform an online build of the restore JCL.

Specify S if you want the Set Up Restore panel displayed so users can select the appropriate options for building the restore JCL.

Leave this keyword value blank if you want the **Select Restore Option** pop-up window displayed.

RESTORE_VIEW_DEFAULT=J|S|(blank)
By default, there is no value assigned to this keyword. Use this keyword to specify the return code.

Specify J if you want to display the restore JCL.

Specify S if you want to display the restore summary.

Leave this keyword value blank if you want the **Select Restore Viewing Option** pop-up window displayed.

QUICK_RESTORE_BADBKUPRC=nnnn|(blank)
Use this keyword to control Quick Restore JCL generation, based on backup return codes. Restore JCL will be generated for only those backups with return codes less than the value you specify for this keyword. If you do not specify a value for this keyword, the default value (8) is used.

You many specify any integer in the range 0-9999 (inclusive).

:PROCESS_OPTIONS section keywords

Use the information in this topic to specify values for the keywords in the :PROCESS_OPTIONS section of the BKMINI member in the product parameter library.

JOB1=sample job card
Use this keyword for Tivoli Enterprise Portal (TEP) batch job submission. For example, `JOB1='//EDPTEPJ1 JOB(ACCT),'TEP',CLASS=A,MSGCLASS=X'`

JOB2=sample JCLLIB card
Use this keyword for Tivoli Enterprise Portal (TEP) batch job submission. For example, `JOB2='//JCLLIB JCLLIB ORDER=(BKM0202.EDP.JCL)'`

JOB3=sample comment card
Use this keyword for Tivoli Enterprise Portal (TEP) batch job submission. For example, `JOB3='//*'`

JOB4=sample comment card
Use this keyword for Tivoli Enterprise Portal (TEP) batch job submission. For example, `JOB4='//*'`

SMF_RECTYPE=n
Use this keyword to specify the SMF record type to be used for Tivoli Enterprise Portal (TEP) batch job status notification. For example, SMF_RECTYPE=255.

TAPE_METADATA=Y|N
Use this keyword to specify whether you want to extract tape data set metadata from the Tape Management System database. The default value for this keyword is N.

MIGRATION_FACILITY=HSM|(blank)
Use this keyword to specify which migration tool your site uses to migrate or archive data. Specify HSM if your site uses HSM. Remove HSM from the value specification if your site does not use HSM.

MAX_BACKUP_TASKS=n
Use this keyword to specify the maximum number of concurrent backup tasks allowed in any one job. The value you specify must be an integer in the range 1-64 (inclusive). The default value for this field is 40.

MAX_RECOVER_TASKS=n
Use this keyword to specify the maximum number of concurrent recover tasks allowed in any one job. The value you specify must be an integer in the range 1-64 (inclusive). The default value for this field is 10.

EVENT_LIST_LIMIT=n
Use this keyword to specify the number of events you want included in the backup summary. The value you specify must be an integer in the range 0-9999 (inclusive); do not use commas in your value specification. The default value for this field is 5.

SDSL_FILTER_MESSAGES=Y|N
Use this keyword to specify whether you want the software to issue messages resulting from SDSL filtering statements. Your specification does not affect warning and error messages, which are always produced. The default value for this keyword is Y.

SDSL_DETAIL_MESSAGES=Y|N
Use this keyword to specify whether you want the software to issue a report containing one message for each dead set name for which an SDSL filter is applied. The report is sent to DD name DEBUGxx. Because this report can be quite large, it is recommended that you leave this keyword set to its default value (N) unless you need to enable it for problem diagnosis.

SDS_EXCLUDE_MESSAGES=Y|N
Use this keyword to specify whether you want the software to issue messages resulting from SDSL EXCLUDE statements. Your specification does not affect warning and error messages, which are always produced. The default value for this keyword is Y.

UNLOADIDS_SPACE='CYLINDERS SPACE(1 2)'
Use this keyword to specify space for the output data set used when the UNLOADIDS command is invoked. You can specify TRACKS, CYLINDERS, BLOCK, or AVBLOCK.

ARCSKEXT_DS_SPACE='CYLINDERS SPACE(10 10)'
Use this keyword to specify space for the output data set passed between the ABARS Manager task and ABARS. You can specify TRACKS, CYLINDERS, BLOCK, or AVBLOCK.

WORKFILE_SPACE='CYLINDERS SPACE(10 20)'
Use this keyword to specify space for the work files used when the ABARS Manager utilities are executed. You can specify TRACKS, CYLINDERS, BLOCK, or AVBLOCK.

UNLOADIDS_UNIT=SYSALLDA
Use this keyword to specify space for the unit to use for the output data set when the UNLOADIDS command is invoked.

ARCSKEXT_UNIT=SYSALLDA
Use this keyword to specify the unit for the data set used to pass data between the ABARS Manager task and ABARS.

WORKFILE_UNIT=SYSALLDA
Use this keyword to specify the unit you want used when allocating space for work files when the ABARS Manager utilities are executed.

SYSOUT_CLASS=*
Use this keyword to send all SYSOUT to SAR (or other such tool) and use the JOBCARD MSGCLASS parameter. Use this parameter to reroute the dynamically allocated SYSOUT to a specific class. If further SYSOUT changes are required, modify the SYSOUT parameters in BKMBPROC

XRENAME_SPACE='CYLINDERS SPACE(10 10)'
Use this keyword to specify space to use for the FULLRENAME output file, if it has not been pre-allocated and you are using the FullRename feature.

XRENAME_UNIT='SYSALLDA'
Use this keyword to specify the unit type to use for the FULLRENAME output file, if it has not been pre-allocated and you are using the FullRename feature.

DISK=SYSALLDA
Use this keyword to specify the default DASD unit name. The default value for this keyword is SYSALLDA.

SOUT="*".
Use this keyword to specify the SYSOUT class used in JCL. The default value for this keyword is *.

BKMIDSHC_PARM="SCAN"
Use this keyword to specify the default execution parameter for the IDS Health Check utility. The default value for this keyword is SCAN.

IAM_DATASETS=Y|N
Use this keyword to specify whether to process IAM (Innovation Access Method) data sets. The default value for this keyword is NO.

Note the following information about related keywords when IAM_DATASETS=Y:

- If IAM_DATASETS=Y, when RSP is started, IAM_SMFTYPE must have a valid user-defined SMF record type value.
- If IAM_DATASETS=Y, when Critical Backup Tracking and Inventory is started IAM_HLQ must be assigned a valid qualifier and the HLQ.IAMPRINT.WORK work data set must have ALTER access.

IAM_SMFTYPE=*n*
Use this keyword to specify the SMF record type of the IAM data sets for RSP. You may specify any integer value in the range 128 - 255. There is no default value for this keyword.

Note: If IAM_DATASETS=Y, IAM_SMFTYPE must have a valid value when RSP is started.

IAM_HLQ=*highlvlqualifier*
Use this keyword to specify the high-level qualifier for the work data set *HLQ*.IAMPRINT.WORK.

Note: If IAM_DATASETS=Y, when Critical Backup Tracking and Inventory is started, IAM_HLQ must be assigned a valid qualifier and the work data set *HLQ*.IAMPRINT.WORK must have ALTER access.

CR+_PROGNAME_PREFIX=CAT
Use this keyword to specify the first three characters of the Catalog Management program name prefix. Valid values are CAT, MCR, or CKM. The default value for this keyword is CAT.

ALBPGNM=IEFBR14
Use this keyword to specify the program name to be used in optional step 2 of the Aggregate LoadBalancer procedure found in member BKMALBP of the product JCL library. To use a REXX EXEC in step 2, specify the batch TSO TMP program name IKJEFT01.

REXXEXEC=BKMREXX1
Use this keyword to specify the REXX EXEC name to be used in optional step 2 of the Aggregate LoadBalancer procedure found in member BKMALBP of the product JCL library.

SYSEXEC=NULLFILE
Use this keyword to specify the REXX EXEC library name to be used in optional step 2 of the Aggregate LoadBalancer procedure found in member BKMALBP of the product JCL library.

DEFAULT=DEFAULT
Use this keyword to specify the default ACDI application name to be used in the APPLEND procedure found in member BKMAPLEP of the product JCL library.

NULLFILE=NULLFILE
Use this keyword to specify the default data set name to be used in numerous procedures found in the product JCL library.

JOBNAME_LIST_INCLUDE_DSNE = Y|N
Use this keyword to specify whether you want a separate column **DSNE** in the Jobname List Report to display number of backup errors.

The default value is **Y**.

DSNLIST_INCLUDE_HSM_FULL = Y|N
Use this keyword to specify whether you want the product to display all HSM backup events in the Dataset List report. The default value for this keyword is N.

:BACKUP_OPTIONS section keywords

Use the information in this topic to specify values for the keywords in the :BACKUP_OPTIONS section of the BKMINI member in the product parameter library.

The values specified here are defaults that can be set subsequently by aggregate in the appropriate ISPF panels. Careful thought should be given to the setting of these options in BKMINI before aggregates are defined to ABARS Manager to avoid the need to change options for specific aggregates.

The keyword values are used as defaults when a new aggregate is added to ABARS Manager, or are the default values displayed when a backup is launched. Values that can be set by aggregate appear on the **Setup** panel when an aggregate is first defined.

If aggregates are maintained by individual users (as opposed to a centralized approach), it is likely that not everyone will have knowledge of the options, and therefore might accept the defaults as determined by the BKMINI keyword values. The values set for the BKMINI keywords, as shipped with the product, are not necessarily the most appropriate choices for your installation.

SDSL_COLLISION_COUNT=*n*
The default value for this keyword is 10.

Specify the SDSL catalog processing collision count allowance. You may specify an integer in the range 0-9999, inclusive.

SERIALIZE_BEFORE_BACKUP=Y|N
The default value for this keyword is N.

If you aggressively migrate data, and are getting frequent ABACKUP return code 073s, you can use this keyword to help ensure that data sets are not moved during the ABACKUP. If an ABACKUP is executing and migration of a data set in the ABACKUP is attempted, the migration fails and can be retried later.

RECORD_EMPTY_INCR_EVENT=Y|N
Use this keyword to specify whether you want ABARS Manager to record in the IDS all incremental ABARS backups that had no data sets to back up. These backups do not have a generation associated with them.

Specify Y to have the software record in the IDS this type of backup.

Specify N if you do not want the software to record in the IDS this type of backup. This is the default value for this keyword.

NO_DS_INCLUDED_IN_SDS_RC=n
Use this keyword to specify the return code to be issued when no data sets are included in the Selection Data Set. This token value is used for ACDI APPLEND and ABARS Manager (including Aggregate LoadBalancer) processing of Selection Data Sets. The default value for this token is 20.

ADD_EMPTY_GDG_BASE_TO_ALLOCATE=Y|N
The default value for this keyword is N. If an ABARS INCLUDE mask catches an empty GDG base, place the GDG base in the ALLOCATE list instead of discarding the entry.

If you specify Y, this keyword does not produce specific messages in the SYSOUT because the switch is at the global level, and once it has been set, it pertains to all aggregates. However, empty GDG bases found in an INCLUDE mask are automatically placed in the ALLOCATE list.

Consider whether you want to use the GDG REPLACE feature on the ARECOVER command. If you GDG base limits change periodically during processing, and you did not specifically put the GDG base in the ALLOCATE list but are doing do by using this keyword, take care to ensure you do not replace the GDG base by accident.

The value you assign to this keyword overrides Backup_Option: Remove GDG base name references other than ALLOCATE.

CCTRIGGER=Y|N
Use this keyword to specify whether you want to execute a REXX EXEC when Concurrent Copy has committed; REXX EXEC submits a job that can be used as a trigger for the scheduler. It is used as a trigger for sites that do not have AUTO Ops.

Before you specify Y for this keyword, look at the sample member AMPCCTRG in the product PARMLIB and modify the job card to ensure it doesn't interfere with your site's job naming standards.

SDS_MODIFICATION_OK=Y|N
Y is the default value for this keyword.

Specify Y if you want ABARS Manager to use the Backup Options settings when an ABACKUP is issued. ABARS Manager stores a copy of the original Selection Data Set in IDS, then offloads the Selection Data Set, goes through the backup option settings, corrects category conflicts, removes uncataloged data sets, moves unsupported data sets to the correct category, and rewrites the Selection Data Set for the ABACKUP. When the ABACKUP is complete, the original Selection Data Set is restored to its original state; a before and an after version of the Selection Data Set are kept in the IDS, which can be very helpful if you have to bring the IDS back from a disaster test to do analysis at a later time.

Specify N if you do not want ABARS Manager to check for error conditions and ignore all Backup Options settings; the ABACKUP is sent directly to ABARS. This keyword is ignored when SDSL or incremental ABARS are used; the Backup Options remain valid under these circumstances. Normal SDSL messages are written to DD DEBUG01.

DEFAULT_PROCESS=V|E|VE
The default value for this keyword is V.

Specify V for ABARS verify only.

Specify E for ABARS execute mode.

Specify VE for ABARS verify and execute.

DEFAULT_UNIT=unitname|3490
Use this keyword to specify the unit for ABARS backup output data sets. *unitname* can be 1-8 characters in length. The default value for this keyword is 3490.

RECEIVE_ABARS_MSGS=Y|N
Use this keyword to specify whether ABARS messages are to be directed to the submitting user's TSO session. The default value for this keyword is Y.

STACK=Y|N
Use this keyword to specify whether the ABARS STACK option is used. STACK under SMS 1.4 or greater causes the C file to be placed on the same tape as the D file. The default value for this keyword is N.

SKIP(PPRC)=Y|N
Use this keyword to specify whether to skip data sets that reside entirely on PPRC primary volumes. The default value for this keyword is N.

SKIP(XRC)=Y|N
Use this keyword to specify whether to skip data sets that reside entirely on XRC primary volumes. The default value for this keyword is N.

LIST(SKIPPED)=Y|N
Use this keyword to specify whether to record a message in the activity log for each data set skipped because of the SKIP(PPRC) or SKIP(XRC) keyword settings. The default value for this keyword is N.

OPTIMIZE=n
Use this keyword to specify the ABARS optimize option. Your specification must be an integer in the range 1-4 (inclusive). The default value for this keyword is 3.

MOVE=Y|N
Use this keyword to specify whether the ABARS backup should be a MOVE function. The default value for this keyword is N.

Specify N if you want ABARS backups to copy the data sets rather than move them.

PROCESS_ONLY_L0=Y|N
Specify Y for this keyword value to specify that only those data sets residing on DASD are processed. Y is the default value for this keyword.

PROCESS_ONLY_L1=Y|N
Specify Y for this keyword value to specify that only those data sets residing on DFSMShsm migration level 1 DASD are processed. Y is the default value for this keyword.

PROCESS_ONLY_L2=Y|N
Specify Y for this keyword value to specify that only those data sets residing on DFSMShsm migration level 2 DASD are processed. Y is the default value for this keyword.

PROCESS_ONLY_UT=Y|N
Specify Y for this keyword value to specify that only those data sets residing on user tape are processed. Y is the default value for this keyword.

PROC_NAME=BKMBPROC
Use this keyword to specify the name of the JCL procedure for ABACKUP and other function submission through the ISPF panels. The default value for this keyword is BKMBPROC.

PROC_STEP=BKMBATCH
Use this keyword to specify the name of the JCL procedure step for ABACKUP submission through the ISPF panels. The default value for this keyword is BKMBATCH.

PROC_BPARM=NONE
Use this keyword to specify the parameter value to be passed to the PROC specified by the PROC_NAME token. The default value for this keyword is NONE.

ALWAYS_SHOW_JC=Y|N
Use this keyword to specify whether you want the job card displayed when backup or recovery jobs are submitted from the ISPF panels.

Specify Y if you want the job card displayed.

Specify N if you do not want the job card displayed.

MOVE_MULTI_CAT_DSN_TO_INC=Y|N
Use this keyword to specify what you want done with data set names found in multiple categories (INCLUDE, ALLOCATE, ACCOMPANY).

Specify Y if you want the data set moved into one appropriate category, which is usually INCLUDE.

Specify N if you do not want the data set moved.

Note: Use the MOVE_MULTI_CAT_DSN_TO_INC_RC keyword to specify the return code you want returned whenever this situation is encountered, regardless of which value you specified for the MOVE_MULTI_CAT_DSN_TO_INC keyword.

MOVE_MULTI_CAT_DSN_TO_INC_RC=0|4|8
Use this keyword to specify the return code you want returned when ABARS finds data set names in multiple categories (INCLUDE, ALLOCATE, ACCOMPANY). This keyword is paired with the MOVE_MULTI_CAT_DSN_TO_INC keyword.

If you specified MOVE_MULTI_CAT_DSN_TO_INC=Y, specify either 0 or 4.

If you specified MOVE_MULTI_CAT_DSN_TO_INC=N, specify 8, which causes the ABARS backup to be bypassed.

CHG_ALC_OF_UN_DSORG_TO_INC=Y|N
Use this keyword to specify what you want done when data sets with an unsupported DSORG are found in the ALLOCATE category.

Specify Y if you want the data set with the unsupported DSORG moved into the INCLUDE category. This is the default value for this keyword.

Specify N if you do not want the data set with the unsupported DSORG moved from the ALLOCATE category to the INCLUDE category.

Note: Use the CHG_ALC_OF_UN_DSORG_TO_INC_RC keyword to specify the return code you want returned whenever this situation is encountered, regardless of which value you specified for the CHG_ALC_OF_UN_DSORG_TO_INC keyword.

CHG_ALC_OF_UN_DSORG_TO_INC_RC=0|4|8
Use this keyword to specify the return code you want returned when ABARS finds a data set with an unsupported DSORG is found in the ALLOCATE category. This keyword is paired with the CHG_ALC_OF_UN_DSORG_TO_INC keyword.

If you specified CHG_ALC_OF_UN_DSORG_TO_INC=Y, specify either 0 or 4.

If you specified CHG_ALC_OF_UN_DSORG_TO_INC=N, specify 8, which causes the ABARS backup to be bypassed.

CHG_DASD_DSN_ACC_TO_INC=Y|N
Use this keyword to specify what you want done when a DASD data set is found in the ACCOMPANY category.

Specify Y if you want the DASD data set moved from the ACCOMPANY category into the INCLUDE category. This is the default value for this keyword.

Specify N if you do not want the DASD data set moved from the ACCOMPANY category to the INCLUDE category.

Note: Use the CHG_DASD_DSN_ACC_TO_INC_RC keyword to specify the return code you want returned whenever this situation is encountered, regardless of which value you specified for the CHG_DASD_DSN_ACC_TO_INC keyword.

CHG_DASD_DSN_ACC_TO_INC_RC=0|4|8
Use this keyword to specify the return code you want returned when ABARS finds a DASD data set in the ACCOMPANY category. This keyword is paired with the CHG_DASD_DSN_ACC_TO_INC keyword.

If you specified CHG_DASD_DSN_ACC_TO_INC=Y, specify either 0 or 4.

If you specified CHG_DASD_DSN_ACC_TO_INC=N, specify 8, which causes the ABARS backup to be bypassed.

CHG_TAPE_DSN_ALC_TO_INC=Y|N
Use this keyword to specify what you want done when a tape data set is found in the ALLOCATE category.

Specify Y if you want the tape data set moved from the ALLOCATE category into the INCLUDE category. This is the default value for this keyword.

Specify N if you do not want the tape data set moved from the ALLOCATE category to the INCLUDE category.

Note: Use the CHG_TAPE_DSN_ALC_TO_INC_RC keyword to specify the return code you want returned whenever this situation is encountered, regardless of which value you specified for the CHG_TAPE_DSN_ALC_TO_INC keyword.

CHG_TAPE_DSN_ALC_TO_INC_RC=0|4|8
Use this keyword to specify the return code you want returned when ABARS finds a tape data set in the ALLOCATE category. This keyword is paired with the CHG_TAPE_DSN_ALC_TO_INC keyword.

If you specified CHG_TAPE_DSN_ALC_TO_INC=Y, specify either 0 or 4.

If you specified CHG_TAPE_DSN_ALC_TO_INC=N, specify 8, which causes the ABARS backup to be bypassed.

CHG_MIGRATED_DSN_ACC_TO_INC=Y|N
Use this keyword to specify what you want done when a migrated data set is found in the ACCOMPANY category.

Specify Y if you want the migrated data set moved from the ACCOMPANY category into the INCLUDE category. This is the default value for this keyword.

Specify N if you do not want the migrated data set moved from the ACCOMPANY category to the INCLUDE category.

Note: Use the CHG_MIGRATED_DSN_ACC_TO_INC_RC keyword to specify the return code you want returned whenever this situation is encountered, regardless of which value you specified for the CHG_MIGRATED_DSN_ACC_TO_INC keyword.

CHG_MIGRATED_DSN_ACC_TO_INC_RC=0|4|8

Use this keyword to specify the return code you want returned when ABARS finds a migrated data set in the ACCOMPANY category. This keyword is paired with the CHG_MIGRATED_DSN_ACC_TO_INC keyword.

If you specified CHG_MIGRATED_DSN_ACC_TO_INC=Y, specify either 0 or 4. The default value for this keyword is 4.

If you specified CHG_MIGRATED_DSN_ACC_TO_INC=N, specify 0, 4, or 8.

CHG_ALC_DSN_B32760_TO_INC=Y|N

Use this keyword to specify what you want done when a data set with a BLKSIZE > 32760 is found in the ALLOCATE category.

Specify Y if you want the data set moved from the ALLOCATE category into the INCLUDE category. This is the default value for this keyword.

Specify N if you do not want the data set moved from the ALLOCATE category to the INCLUDE category.

Note: Use the CHG_ALC_DSN_B32760_TO_INC_RC keyword to specify the return code you want returned whenever this situation is encountered, regardless of which value you specified for the CHG_ALC_DSN_B32760_TO_INC keyword.

CHG_ALC_DSN_B32760_TO_INC_RC=0|4|8

Use this keyword to specify the return code you want returned when ABARS finds a data set with BLKSIZE > 32760 in the ALLOCATE category. This keyword is paired with the CHG_ALC_DSN_B32750_TO_INC keyword.

If you specified CHG_ALC_DSN_B32750_TO_INC=Y, specify either 0 or 4. The default value for this keyword in this situation is 4.

If you specified CHG_ALC_DSN_B32760_TO_INC=N, specify 8, which causes the ABARS backup to be bypassed.

REMOVE_CAT_ERROR_DSNS=Y|N

Use this keyword to specify what you want done when a catalog error occurs while information is being obtained on the data sets in the Selection Data Set.

Specify Y if you want the data sets that generate catalog lookup errors removed from the Selection Data Set. This is the default value for this keyword.

Specify N if you do not want the data sets that generate catalog lookup errors removed from the Selection Data Set.

Note: Use the REMOVE_CAT_ERROR_DSNS_RC keyword to specify the return code you want returned whenever this situation is encountered, regardless of which value you specified for the REMOVE_CAT_ERROR_DSNS keyword.

REMOVE_CAT_ERROR_DSNS_RC=0|4|8

Use this keyword to specify the return code you want returned when a catalog lookup error occurs while information is being obtained on the data sets in the Selection Data Set. This keyword is paired with the REMOVE_CAT_ERROR_DSNS keyword.

If you specified REMOVE_CAT_ERROR_DSNS=Y, specify either 0 or 4. The default value for this keyword in this situation is 4.

If you specified REMOVE_CAT_ERROR_DSNS=N, specify 8, which causes the ABARS backup to be bypassed.

REMOVE_UNCAT_DSNS=Y|N
Use this keyword to specify what you want to happen when ABARS Manager encounters an uncataloged data set as it performs preprocessing tasks for the ABACKUP.

If ABARS Manager submits the data set to ABARS as uncataloged, the data set might be bypassed by the ARCBEEXT, which would prevent an error. However, if ARCBEEXT is not installed and ABARS Manager submits the uncataloged data set to ABARS anyway, the ABACKUP could result in an ABEND.

Alternatively, ABARS Manager can remove the uncataloged data set from the Selection Data Set list prior to submitting the list to ABARS for processing.

Specify Y if you want ABARS Manager to remove the uncataloged data sets from the Selection Data Set before passing the list to ABARS.

Specify N if you do not want the uncataloged data sets removed from the Selection Data Set. ABARS Manager sets the value for this keyword to N in situations such as the following:

```
INCLUDE(BKMI.IVP.FB.**  +
    BKMI.IVP.CNTL       +
    BKMI.CNTL)
```

where the interpretation of the continuation character results in the data set name BKMI.IVP.CNTL.BKMI.CNTL, which identifies an nonexistent (and therefore uncataloged) data set. If this data set were to be removed by ABARS Manager from the Selection Data Set list, you would be unaware that a problem exists.

Be aware that if you specify N for this keyword *and you specify REMOVE_UNCAT_DSNS_RC=4*, an RC4 is issued if the ARCBEEXT skips the data set. If ARCBEEXT is installed and the it bypasses the uncataloged data set, an RC0 is issued.

Note: Use the REMOVE_UNCAT_DSNS_RC keyword to specify the return code you want returned whenever this situation is encountered, regardless of which value you specified for the REMOVE_UNCAT_DSNS keyword.

REMOVE_UNCAT_DSNS_RC=0|4|8
Use this keyword to specify the return code you want returned when ABARS Manager encounters an uncataloged data set in the Selection Data Set list. This keyword is paired with the REMOVE_UNCAT_DSNS keyword.

If you specified REMOVE_UNCAT_DSNS=Y, specify either 0 or 4.

If you specified REMOVE_UNCAT_DSNS=N, specify 0, 4, or 8. Specify 8 to bypass the ABARS backup.

CHG_ALIAS_TO_TRUENAME=Y|N
Use this keyword to specify what you want done when ABARS Manager encounters an alias in the Selection Data Set list.

Specify Y if you want the alias replaced with the true data set name before the Selection Data Set list is passed to ABARS. This is the default value for this keyword.

Specify N if you do not want the alias replaced with the true data set name before the Selection Data Set list is passed to ABARS.

Note: Use the CHG_ALIAS_TO_TRUENAME_RC keyword to specify the return code you want returned whenever this situation is encountered, regardless of which value you specified for the CHG_ALIAS_TO_TRUENAME keyword.

CHG_ALIAS_TO_TRUENAME_RC=0|4|8
Use this keyword to specify the return code you want returned when ABARS Manager encounters an alias in the Selection Data Set. This keyword is paired with the CHG_ALIAS_TO_TRUENAME keyword.

If you specified CHG_ALIAS_TO_TRUENAME=Y, specify either 0 or 4. The default value for this keyword in this situation is 4.

If you specified CHG_ALIAS_TO_TRUENAME=N, specify 8, which causes the ABARS backup to be bypassed.

REMOVE_ORPHANED_ALIASES=Y|N
Use this keyword to specify what you want done when ABARS Manager finds in the Selection Data Set list an alias for an uncataloged data set.

Specify Y if you want the alias removed from the Selection Data Set list before it is passed to ABARS. This is the default value for this keyword.

Specify N if you do not want the alias removed from the Selection Data Set list before it is passed to ABARS.

Note: Use the REMOVE_ORPHANED_ALIASES_RC keyword to specify the return code you want returned whenever this situation is encountered, regardless of which value you specified for the REMOVE_ORPHANED_ALIASES keyword.

REMOVE_ORPHANED_ALIASES_RC=0|4|8
Use this keyword to specify the return code you want returned when ABARS Manager finds in the Selection Data Set list an alias for an uncataloged data set. This keyword is paired with the REMOVE_ORPHANED_ALIASES keyword.

If you specified REMOVE_ORPHANED_ALIASES=Y, specify either 0 or 4. The default value for this keyword in this situation is 4.

If you specified REMOVE_ORPHANED_ALIASES=N, specify 8, which causes the ABARS backup to be bypassed.

CHG_CATLG_TO_ALC=Y|N
Use this keyword to specify what you want done when ABARS Manager finds an ICF catalog name in the Selection Data Set.

Specify Y if you want the ICF catalog moved to the ALLOCATE list before it is passed to ABARS. This is the default value for this keyword.

Specify N if you do not want the ICF catalog moved to the ALLOCATE list before it is passed to ABARS.

Note: Use the CHG_CATLG_TO_ALC_RC keyword to specify the return code you want returned whenever this situation is encountered, regardless of which value you specified for the CHG_CATLG_TO_ALC keyword.

CHG_CATLG_TO_ALC_RC=0|4|8
Use this keyword to specify the return code you want returned when ABARS Manager finds a catalog name in the Selection Data Set. This keyword is paired with the CHG_CATLG_TO_ALC keyword.

If you specified CHG_CATLG_TO_ALC=Y, specify either 0 or 4. The default value for this keyword in this situation is 4.

If you specified CHG_CATLG_TO_ALC=N, specify 8, which causes the ABARS backup to be bypassed.

REMOVE_GDGB_REF_IF_NOT_ALC=Y|N
Use this keyword to specify what you want done when ABARS Manager finds in the Selection Data Set a GDG base in any category other than ALLOCATE.

Specify Y if you want the GDG base name reference removed from the Selection Data Set list before it is passed to ABARS. This is the default value for this keyword.

Specify N if you do not want the GDG base name removed from the Selection Data Set list before it is passed to ABARS.

Note: Use the REMOVE_GDGB_REF_IF_NOT_ALC_RC keyword to specify the return code you want returned whenever this situation is encountered, regardless of which value you specified for the REMOVE_GDGB_REF_IF_NOT_ALC keyword.

REMOVE_GDGB_REF_IF_NOT_ALC_RC=0|4|8
Use this keyword to specify the return code you want returned when ABARS Manager finds in the Selection Data Set list a GDG base name reference in a category other than ALLOCATE. This keyword is paired with the CHG_CATLOG_TO_ALC keyword.

If you specified REMOVE_GDGB_REF_IF_NOT_ALC=Y, specify either 0 or 4. The default value for this keyword in this situation is 4.

If you specified REMOVE_GDGB_REF_IF_NOT_ALC=N, specify 8, which causes the ABARS backup to be bypassed.

FUZZY_CAT_STATUS_RC=0|4|8

Use this keyword to specify the return code you want returned when ABARS Manager preprocessing for an ABACKUP detects changes to the catalog entry of a data set referenced in the Selection Data Set.

These changes can occur when the application is still running and a volume is added to a data set or a new generation data set is created. If catalog changes are ignored by ABARS Manager, errors can occur and the ABACKUP is no longer a point-in-time. If the return code specified is 0 or 4, the backup is submitted even though it might have problems or fail. Return code 8 causes the backup to be aborted. Valid return codes are 0, 4 and 8.

REM_DCOL_DSN_CAT_DIFF_VOL=Y|N

When data set names are obtained using SDSL's DCOLLECT option, it is possible that a data set from the selected volumes is cataloged, but the catalog entry points to the same data set name on another volume (`(NOT CAT 2)`, for example). Use this keyword to specify what you want done when ABARS Manager finds a catalog entry that points to a data set on a different volume.

Specify Y if you want ABARS Manager to remove such data set names from the Selection Data Set list. This is the default value for this keyword.

Specify N if you do not want such data sets removed from the Selection Data Set list before it is passed to ABARS.

Note: Use the REM_DCOL_DSN_CAT_DIFF_VOL_RC keyword to specify the return code you want returned whenever this situation is encountered, regardless of which value you specified for the REM_DCOL_DSN_CAT_DIFF_VOL keyword.

REM_DCOL_DSN_CAT_DIFF_VOL_RC=0|4|8

This is an SDSL-only keyword.

Use this keyword to specify the return code you want returned when ABARS Manager finds in the Selection Data Set list a catalog entry that points to a data set on a different volume. This keyword is paired with the REM_DCOL_DSN_CAT_DIFF_VOL keyword.

If you specified REM_DCOL_DSN_CAT_DIFF_VOL=Y, specify either 0 or 4. The default value for this keyword in this situation is 4.

If you specified REM_DCOL_DSN_CAT_DIFF_VOL=N, specify 8, which causes the ABARS backup to be bypassed.

BKM_RC_EQUAL_ABARS_RC=Y|N

Use this keyword to specify what you want done when the ABARS Manager return code differs from the ABARS return code in the :ABARS_RC_OVERRIDES section of the activity log.

Specify Y if you want the ABARS Manager return code set equal to the ABARS return code, except in situations where the ABARS Manager return code is higher than the ABARS return code or the value specified for a backup option keyword causes a higher return code than the ABARS return code.

Specify N if you do not want to override the activity log return code. This is the default value for this keyword.

Note: Return codes that were set because ARCBEEXT bypassed a data set cannot be overridden. If SDS_MODIFICATION=N, the BKM_RC_EQUAL_ABARS_RC keyword applies automatically.

SDSL_HLQ=(blank)|dsname

This is an SDSL-only keyword.

Use this keyword to specify the high level qualifier you want to use for the name of the temporary data set created when you are using the EL line command to edit SDSL. The aggregate name will be appended to the high level qualifier you specify as the value for this keyword. You can prevent multiple users from editing the same temporary data set by assigning a data set name as the value for this keyword.

If you leave the value assignment for this keyword blank, the standard name is used for the temporary data set created when a user edits SDSL.

:ABARS_RC_OVERRIDES section keywords

Use the information in this topic to set appropriate values for the keywords in the :ABARS_RC_OVERRIDES section of the BKMINI member in the product parameter library. If BKM_RC_EQUAL_ABARS_RC=Y, the values you specify for these keywords affect the default return codes generated by the ABARS Manager batch backup job. ARCBEEXT return codes cannot be overridden.

If ABARS produces a return code higher than the value you specify for one of these keywords, or an error occurs, the ABARS return code is returned.

Note: The keywords in this section are commented. Copy the keyword you want to use, then paste it directly after the :ABARS_RC_OVERRIDES line. Then uncomment the pasted keyword line. Do not uncomment or delete the original set of commented keywords from BKMINI.

ADR321E=0|4|8
Specify the override value you want returned when ABARS encounters a data set name not on volume situation. The default value for this keyword is 8.

ADR380E=0|4|8
Specify the override value you want returned when ABARS encounters a data set name not processed situation. The default value for this keyword is 8.

ADR383W=0|4|8
Specify the override value you want returned when ABARS encounters a data set name not on volume situation. The default value for this keyword is 4.

ADR411W=0|4|8
Specify the override value you want returned when ABARS encounters a `not serialized on request` situation. The default value for this keyword is 4.

ADR427E=0|4|8
Specify the override value you want returned when ABARS encounters an `error in (WDS | WDS image)` situation. The default value for this keyword is 8.

ADR428E=0|4|8
Specify the override value you want returned when ABARS encounters a `VTOC entries not found for cluster` situation. The default value for this keyword is 8.

ADR730W=0|4|8
Specify the override value you want returned when ABARS encounters a `cluster is open` situation. The default value for this keyword is 4.

ADR735W=0|4|8
Specify the override value you want returned when ABARS encounters a `use of concurrent copy failed` situation. The default value for this keyword is 4.

ADR740W=0|4|8
Specify the override value you want returned when ABARS encounters a `migrated–no longer available for selection` situation. The default value for this keyword is 4.

ADR758E=0|4|8
Specify the override value you want returned when ABARS encounters a `volume checking failed for data set` situation. The default value for this keyword is 8.

ADR804W=0|4|8
Specify the override value you want returned when ABARS encounters an `EOF not found where expected` situation. The default value for this keyword is 4.

ARC6059E=0|4|8
Specify the override value you want returned when ABARS encounters an error in obtaining DFSMShsm control. The default value for this keyword is 8.

ARC6073E=0|4|8
Specify the override value you want returned when ABARS encounters an error when attempting to enqueue. The default value for this keyword is 8.

ARC6077E=0|4|8
Specify the override value you want returned when ABARS encounters a volume already in the ABARS RACF tape volume set. The default value for this keyword is 8.

ARC6158E=0|4|8
Specify the override value you want returned when ABARS encounters a catalog error while executing a function. The default value for this keyword is 8.

ARC6160E=0|4|8
Specify the override value you want returned when ABARS encounters an I/O error while reading or writing data. The default value for this keyword is 8.

ARC6196W=0|4|8
Specify the override value you want returned when ABARS encounters an 'SMS construct unable to be retrieved' situation. The default value for this keyword is 4.

:RECOVERY_OPTIONS section keywords

Use the :RECOVERY_OPTIONS section in member BKMINI of the product parameter library to specify values for the keywords that determine how ABARS Manager processes a recovery. The values you specify in BKMINI are default values that individual users can override for a selected aggregate by specifying other recovery options on Advanced Backup and Recovery for z/OS ISPF panels.

DEFAULT_PROCESS=V|E|VE
Use this keyword to specify the default recovery process.

- V–Perform an ABARS verify operation only.
- E–Perform an ABARS execute.
- VE–Perform an ABARS verify operation and then an execute.

DATASET_CONFLICT=R|B|(blank)
Use this keyword to specify what you want done in situations where ABARS Manager is asked to recover a data set that is already cataloged.

- R–Delete the cataloged data set and replace it with the data set to recover.
- B–Do not recover any data set that is already cataloged.
- blank–Do not pass to ABARS a value for this keyword. This is the default value for DATASET_CONFLICT.

Note: You can use this keyword in conjunction with the RECOVERNEWNAMEALL or RECOVERNEWNAMELEVEL parameters.

DEFAULT_UNIT=*unitname*|3490
Use this keyword to specify the default tape unit for the ABARS ARECOVER C, D, I, and O input files.

- *unitname*–Specify a valid tape unit type: 3400-3, 3400-4, 3400-5, 3400-6, 3400-9, 3480, 3480X, 3490, 3590-1.
- 3490–This is the default value.

XMIT=Y|N
Use this keyword to specify the unit you want ARECOVER to use when allocating the control and data files.

- Y–ARECOVER searches the catalog for the control and data files, and if found, uses the unit and volser from the catalog when allocating these data sets.
- N–ARECOVER uses the unit specified. This is the default value for the XMIT keyword.

RECEIVE_ABARS_MSGS=Y|N
Use this keyword to specify how you want ABARS Manager to do with HSM messages.

- Y–Send the messages to the originating user's terminal. This is the default value.
- N–Do not send the HSM messages to the originating user's terminal. All HSM messages appear in the Advanced Backup and Recovery for z/OS processing log.

DEFAULT_COPYSET=*n*
Use this keyword to specify the copyset to be recovered by ARECOVER. Specify an integer in the range 1-15, inclusive. The default value for this keyword is 1.

RECOVER_MIGRATED=ML1|ML2|SOURCELEVEL|SOURCE|(blank)
Use this keyword to specify the level to which you want migrated data sets recovered.

- ML1–All migrated data sets are recovered to ML1.
- ML2–All migrated data sets are recovered to ML2.
- SOURCELEVEL or SOURCE–All migrated data sets are recovered to their original migration level.
- blank–If you do not specify a value for this keyword, the value defaults to the ABARS value. This is the default value for RECOVER_MIGRATED.

RECOVER_ACTLOG=Y|N
Use this keyword to specify whether you want to recover the activity log if it exists on the output tape for the aggregate being recovered.

- Y–Recover the activity log.
- N–Do not recover the activity log. This is the default value.

RECOVER_INSTDSN=Y|N
Use this keyword to specify whether you want to recover the instruction data set if it exists on the output tape for the aggregate being recovered.

- Y–Recover the instruction data set.
- N–Do not recover the instruction data set. This is the default value.

PERCENT_UTILIZED=nnn|(blank)
Use this keyword to specify the percentage to which you want DFSMSdss to fill the L0 volumes in the associated ARPOOL for an aggregate.

- *nnn*–Specify an integer in the range 1-100, included.
- (blank)–Leave this keyword value blank if you want the value to default to the ABARS value. This is the default value for PERCENT_UTILIZED.

VOLCOUNT_ANY=Y|N
Use this keyword to specify whether you want to allow DFSMSdss to allocate as many volumes as necessary (up to a maximum of 59) for all L0 data sets by overriding the SETSYS ABARSVOLCOUNT command.

- Y–Override the SETSYS ABARSVOLCOUNT command.
- N–Do not override the SETSYS ABARSVOLCOUNT command. This is the default value for VOLCOUNT_ANY.

TARGET_GDS=DEFERRED|ACTIVE|ROLLEDOFF|SOURCE
Use this keyword to specify the status you want assigned to the target data set by overriding the SETSYS ARECOVERTGTGDS command.

- DEFERRED–Assign the target data set the DEFERRED status.
- ACTIVE–Assign the target data set the ACTIVE status.
- ROLLEDOFF–Assign the target data set the ROLLEDOFF status.
- SOURCE–Assign the target data set the same status as that of the source data set.

TARGET_UNIT=unittype
Use this optional keyword to specify the tape unit type for data sets to be recovered to tape. Specify a valid tape unit type: 3400-3, 3400-4, 3400-5, 3400-6, 3400-9, 3480, 3480X, 3490, or 3590-1

MENTITY=racfmodeldsn
Use this keyword to specify the RACF model data set name.

RECOVER_ONLY_L0=Y|N
Use this keyword to specify whether you want to restrict the recovery to only those data sets residing on DASD.

- Y–Recover only those data sets residing on DASD. This is the default value.
- N–Do not restrict the recovery to data sets residing on DASD.

RECOVER_ONLY_L1=Y|N
Use this keyword to specify whether you want to restrict the recovery to only those data sets residing on DFSMShsm migration level 1 DASD.

- Y–Recover only those data sets residing on DFSMShsm migration level 1 DASD. This is the default value.
- N–Do not restrict the recovery to data sets residing on DFSMShsm migration level 1 DASD.

RECOVER_ONLY_L2=Y|N
Use this keyword to specify whether you want to restrict the recovery to only those data sets residing on DFSMShsm migration level 2 tape or DASD.

- Y–Recover only those data sets residing on migration level 2 tape or DASD. This is the default value.
- N–Do not restrict the recovery to data sets residing on migration level 2 tape or DASD.

RECOVER_ONLY_UT=Y|N
Use this keyword to specify whether you want to restrict the recovery to only those data sets residing on user tape.

- Y–Recover only those data sets residing on user tape. This is the default value.
- N–Do not restrict the recovery to data sets residing on user tape.

PREVENT_BACKLEVEL=Y|N
Use this keyword to specify whether you want Prevent_BackLevel applied to all aggregate ARECOVERS.

- Y–Apply Prevent_BackLevel to all aggregate ARECOVERS.
- N–Do not apply Prevent_BackLevel to aggregate ARECOVERS. This is the default value.

BACKLEVEL_DSN=*vsambackleveldsn*
Use this keyword to specify the VSAM backlevel data set name you want used.

PROC_NAME= BKMBPROC
Use this keyword to specify the name of the JCL procedure for ARECOVER submission through the Advanced Backup and Recovery for z/OS ISPF panels. The default value for this keyword is BKMBPROC.

PROC_STEP= BKMBATCH
Use this keyword to specify the name of the JCL procedure step for ARECOVER submission through the Advanced Backup and Recovery for z/OS ISPF panels. The default value for this keyword is BKMBATCH.

ALWAYS_SHOW_JC=Y|N
Use this keyword to specify whether you want the job card displayed whenever a backup or recovery is submitted through the Advanced Backup and Recovery for z/OS ISPF panels.

- Y–Display the job card. This is the default value.
- N–Do not display the job card.

GDGREPLACE=Y|N
Use this keyword to specify how you want GDG base catalog entries treated *when there is no ABARS conflict resolution specified*.

In native ABARS, DATASETCONFLICT(REPLACE) must be specified to force GDG data sets to restore, even though they might not exist in the target catalog. For each GDG data set to be restored, ABARS treats the need to update the GDG base catalog entry as a conflict; this might be undesirable if you do not want existing data sets replaced.

The GDGREPLACE keyword uses the ABARS CR exit to allow GDG base catalog entries to be updated, but considers existing data sets to be conflicts.

Be aware that active GDG data sets (data sets that have not been rolled off) do not have separate catalog entries, even though the ISPF 3.4 panel might lead you to think they do. Active GDG data sets for a given base are represented collectively by a single catalog entry; therefore, each data set in a restored GDG causes the base entry to be updated.

- Y–In a single restore, update the GDG base catalog entries without replacing existing data sets.
- N–In a single restore, do not update the GDG base catalog entries. The native ABARS DATASETCONFLICT(REPLACE) parameter is used to determine whether the GDG base catalog entries are updated. This is the default value.

AGGENQ=Y|N
Use this keyword to specify whether you want ABARS Manager to restore one or more data sets from an aggregate group while a backup of the same aggregate is executing.

- Y–Prevent the restoration of data sets from an aggregate group while the backup of that aggregate is executing. This is the default value.
- N–Do not allow serialization by ABARS Manager on the aggregate name.

BYPASS_CATALOG_LOOKUP_DURING_ARECOVER=Y|N
Use this keyword to specify whether you want to bypass catalog lookup during ARECOVER postprocessing.

- Y–Bypass catalog lookup during postprocessing to speed up the ARECOVER. Be aware that specifying Y means data set space information is not recorded.
- N–Do not bypass catalog lookup during postprocessing. This is the default value.

ALLOWFULLRENAME=Y|N
Use this keyword to enable the renaming of the data sets being recovered.

- Y–Enable the FullRename feature. You can use the FULLRENAMEMASK parameter to specify masks that rename any part of the data set name. This is the default value.
- N–Do not enable the FullRename feature.

OUTPUTRENAMEDSN=dsn
Use this keyword to specify the name of the ALTER output data set created by FullRename to hold the IDCAMS ALTER control statements. By default, this output data set is named &HSMUID.&HOSTID.&AGGNAME.&DATE.&TIME.&USERID, where:

- &HSMUID is the authorized HSM User-ID value (refer to the SETSYS options).
- &HOSTID is the HSM Host-ID (refer to ARCCMDXX in the HSM PARMLIB).
- &AGGNAME is the name of the aggregate being recovered.
- &DATE is the current date in D*yyyyddd* format.
- &TIME is the current time in T*hhmmss* format.
- &USERID is the user ID of the user who submitted the ARECOVER.

You can specify a data set name that is fully qualified or contains any of the above values (or both).

FINALRENAMEACTION=FILE|EXEC
Use this keyword to specify how you want the final FullRename action performed.

- FILE–Create and write the ALTER control statements to an output file whose name is determined by the value you assign to the OUTPUTRENAMEDSN keyword. You can verify the final ALTER statements in this data set before you submit it as input to a separate IDCAMS job for execution. The output file is retained on the system for reuse. This is the default value.
- EXEC–Create and execute immediately the ALTER commands for renaming the data set. The value assigned to the BKMINI RENAMEDSNDISP keyword determines the output data set disposition.

RENAMEDSNDISP=KEEP|DELETE
Use this keyword to specify the disposition of the file containing the ALTER control statements for renaming recovered data sets. This keyword and its assigned value are ignored if you specified OUTPUTRENAMEDSN=FILE.

Note: You can use the ARECOVER command statement to override this keyword and its assigned value.

- KEEP–Save the ALTER control statement file as a cataloged data set. KEEP is the default value for this keyword.
- DELETE–Delete the ALTER control statement file when the IDCAMS functions have been completed.

RENAME_GDGS=Y|N
Use this keyword to specify whether you want the generation data sets renamed during a FullRename.

- N–Do not rename the generation data sets, but issue an error message when a generation data set is encountered during the FullRename. This is the default value.
- Y–Rename the generation data sets.

RENAME_GDG_DEFINE=Y|N
Use this keyword to specify what you want done when a generation data set is being renamed (as part of a FullRename), but the target GDG base does not exist.

- Y–Define a target GDG base entry, using the new name and limit number from the original name and last ABACKUP.
- N–Do not define a target GDG base entry. This is the default value for this keyword.

RENAME_GDG_SCRUB=Y|N
Use this keyword to specify what you want done when a generation data set is being renamed (as part of a FullRename), but the target GDG base is populated already.

- Y–Delete the existing entries only; do not delete the base. The limit remains unchanged.

  Note: Be aware that if you specify both RENAME_GDG_SCRUB=Y and RENAME_GDG_DELETE=Y, RENAME_GDG_DELETE takes precedence and the base is deleted and redefined.
- N–Do not delete existing entries. This is the default value for this keyword.

RENAME_GDG_DELETE=Y|N
Use this keyword to specify what you want done with the base entry in FullRename situations where the target base already exists.

- Y–Delete the existing base entry and all its associated GDG entries. Define the GDG base with the limit recorded from the most current ABACKUP.

  Note: Be aware that if you specify both RENAME_GDG_SCRUB=Y and RENAME_GDG_DELETE=Y, RENAME_GDG_DELETE takes precedence and the base is deleted and redefined.
- N–Do not delete the existing base entry. This is the default value for this keyword.

NATIVE_MODE_PDS=dsname
Use this native-mode-only keyword to specify that you want HSEND ARECOVER commands written into the specified partitioned data set, and that you do not want the aggregate submitted to HSM.

The allocation rules for the partitioned data set are as follows:

- It must be a partitioned data set.
- The partitioned data set must already exist as LRECL=80.
- The partitioned data set must be allocated large enough and with enough directory blocks to support all the aggregates.

:LOAD_ACTIVITY_LOG section keywords

Use the information in this topic to set appropriate values for the keywords in the :LOAD_ACTIVITY_LOG section of the parmlib's BKMINI member.

DEFAULT_ACTLOG_MASK=HSMACT.*.A*.*.D*.T*
Use this keyword to specify the default data set name or mask to search for and display when using submenu Utilities/Load Activity Log.

:BKMSTSKD_OPTIONS section keywords

Use the information in this topic to set appropriate values for the keywords in the :BKMSTSKD_OPTIONS section of member BKMINI in the product parameter library. The values you assign to these keywords affect the behavior of incremental ABARS and the CHANGED_DATASET_TRACKING feature of Critical Backup Tracking and Inventory .

DAD_LOG=???.????????.LOG {(*)|(sysid)}
Use this keyword to specify the names of the log file associated with each BKMSTSKD started task. By default, there is one log file that can be shared across the BKMSTSKD started tasks. :DAD_LOG=DAD.BKMSTSKD.LOG (*).

You can specify a different log for each BKMSTSKD started task; however, unless Technical Support specifically recommends doing so, it is recommended that you share a single log file across all BKMSTSKD started tasks. If you do specify separate log files for each BKMSTSKD started task, you must include the appropriate SMF system ID in each keyword value you specify, as shown in the following example:

```
DAD_LOG=DAD.BKMSTSKD.PROD.LOG(PROD)    +
    DAD.BKMSTSKD.DVLP.LOG(DVLP)    +
    DAD.BKMSTSKD.TEST.LOG(TEST)
```

DAD_AUTOSTART=[{YES(*|sysid)}|{NO(*|sysid)}]
Use this keyword to specify whether you want the BKMSTSKD started tasks to start their internal functions automatically. YES is the default value for all BKMSTSKD started tasks: AUTOSTART=YES(*)

You can specify NO for all or for individual BKMSTSKD started tasks, which requires the operator to use F commands to start the functions. If you choose to specify NO for some instances of BKMSTSKD, you must include the appropriate SMF system ID in each keyword value specification. For example:

```
AUTOSTART=YES(PROD)    +
          NO(DVLP)
```

DAD_DSPC_BUFFERS=5000 (*|sysid)
Use this keyword to specify for every BKMSTSKD or for each individual instance of BKMSTSKD the number of data space buffers to hold DMF data awaiting placement in the DAD. The default value for this keyword is 5000 and it applies to all instances of BKMSTSKD. For example:

DAD_DSPC_BUFFERS=5000(*)

You can specify a different number of buffers for all instances of BKMSTSKD or for individual instances of BKMSTSKD; however, you should specify differing numbers of buffers for individual instances of BKMSTSKD only if recommended by Technical Support. If you do specify the number of buffers for each instance of BKMSTSKD, you must include in your value specification the appropriate SMF system ID for each. For example:

```
DAD_DSPC_BUFFERS=05000(SYSA)    +
                 10000(SYSB)
```

DAD_PURGE_UNCATLG=[{YES(*|sysid)}|{NO(*|sysid)}]
Use this keyword to specify whether you want the BKMSTSKD to remove records from the DAD if the data set represented is uncataloged. The default value for this keyword is YES and it applies to all instances of BKMSTSKD. For example:DAD_PURGE_UNCATLG=YES(*)

If you specify NO, BKMSTSKD does not delete the DAD records of uncataloged data sets.

You can apply different values for this keyword to individual instances of BKMSTSKD. If you choose to do so, you must include in each value specification the appropriate SMF system ID, as shown in the following example:

```
DAD_PURGE_UNCATLG=YES(SYSA)    +
                  NO(SYSB)
```

DAD_PURGE_AFTER_DAYS=0
Use this keyword to specify how many days you want to wait after an uncataloging before the DAD records for the uncataloged data set are deleted. The only valid value for this keyword is 0, which means the record is deleted from the data as soon as the SMF delete transaction occurs.

DAD_PURGE_START_TIME=13.45
Values assigned to this keyword are ignored by this release of the software. When nonzero values of the PURGE_AFTER_DAYS are supported, support for the DAD_PURGE_START_TIME keyword will be enabled.

DAD_PURGE_SUSPEND=0
This release of the software does not support the DAD_PURGE_SUSPEND keyword.

DAD_MSGQUEUE=msgqueue.dsn
Use this keyword to specify the name of the DAD MSGQUEUE data set. All instances of BKMSTSKD updating the same DAD use the same MSGQUEUE data set.

DAD_OVERFLOW_DSN=overflow.dsn(sysid1)...overflow.dsn(sysidn)
Use this keyword to specify the name of the overflow data set for each instance of the BKMSTSKD started task. For each value specification, you must include the appropriate SMF system ID. For example:

```
DAD_OVERFLOW_DSN=DAD.PROD.OVERFLOW(PROD)    +
                 DAD.DVLP.OVERFLOW(DVLP)    +
                 DAD.TEST.OVERFLOW(TEST)
```

The overflow data sets are allocated dynamically the first time the DAD task starts up on each system, using the values you specify for the DAD_OVERFLOW_PARM keyword.

DAD_OVERFLOW_PARMS=BLKSIZE(hlq) UNIT(xxxx SPACE(nn nn) CYLINDERS|TRACKS RACCHECK(YES|NO) VOLUME(volser)
Use this keyword to specify the allocation values for the overflow data sets on each instance of the BKMSTSKD started task. By default, the following allocation values are used:

```
BLKSIZE(0) UNIT(SYSALLDA) SPACE(10 2) CYLINDERS +
RACCHECK(YES)
```

Adhering to the TSO ALLOCATE command syntax, you can add other parameters, but the total length you specify cannot exceed 136. The overflow data set is fixed block and has a 300-byte record size.

- If you want to allocate the overflow data set on a specific volume, include the VOLUME(*volser*) parameter, where *volser* is the volume on which you want the overflow data set to reside.
- Specify RACCHECK(YES) if you want RACF checking done.
- You cannot change the LRECL, DDNAME, DSORG, RECFM, or DISP attributes for the overflow data sets.
- The overflow data set should not be pre-allocated.

:SI027_VALUES section keywords

Make changes or additions to the keyword specifications in this section only under the guidance of Technical Support.

:BKMSTSKA section keywords

Use the information in this topic to set appropriate values for the BKMINI keywords that affect the behavior of the Critical Backup Tracking and Inventory started task, BKMSTSKA.

CBTI_AUTOSTART=YES|NO
Use this keyword to specify whether you want all the individual processes within the BKMSTSKA structure started automatically. YES is the default value for this keyword.

If you specify NO, the individual processes within the BKMSTSKA structure are started separately.

CBTI_MSGLOG=?..MSGLOG
Use this keyword to specify the name of the message log data set that records the informational, warning, and error messages generated by the BKMSTSKA started task. If you have multiple instances of BKMSTSKA, they all share the same message log data set.

CBTI_BIF=?..BIF
Use this keyword to specify the name of the batch intercept file (BIF) used by Critical Backup Tracking and Inventory to identify job filters to track. If you have multiple instances of BKMSTSKA, they all share the same batch intercept file.

CBTI_MSGQUEUE=?..MSGQUEUE
Use this keyword to specify the name of the message queue data set used for cross-system communication between participating SMF systems running BKMSTSKA and their users. If you have multiple instances of BKMSTSKA, they all share the same message queue data set.

CBTI_MSGQUEUE_WAIT=1
Use this keyword to specify how many 5-second intervals you want BKMSTSKA to wait before checking the message queue data set (identified by the CBTI_MSGQUEUE keyword value) for cross-system commands. For example, if you want the started task to wait 20 seconds before checking the message queue data set for these commands, you would specify CBTI_MSGQUEUE_WAIT=4.

If you have multiple instances of BKMSTSKA running, they all use the same wait interval.

AUTO_PROCESS_FLAT_FILES=Y
Specify Y to indicate you want the flat file copy task to be started when the BKMSTSKA started task is started. Specify N to indicate you do not want the started task to process flat file data automatically. The interval between attempts to discover new flat file data is controlled by the PROCESS_FLAT_FILES_INTERVAL token setting.

Note: This keyword is ignored if CBTI_AUTOSTART=N.

PROCESS_FLAT_FILES_INTERVAL=15
Use this keyword to specify the number of 2-second intervals you want to elapse between BKMSTSKA started task attempts to discover any new flat file data to process.

You can also use this token to control BKUPEND delays waiting for completion of flat file processing.

Note: Setting this value too low might cause unacceptable BKMSTSKA performance results.

AUTO_DISCOVER_MULTILVL_BKUPS=N
Specify Y to indicate you want the BKMSTSKA started task to discover automatically multilevel backups The interval between discovery attempts is controlled by token MLB_DISCOVER_INTERVAL.

Specify N to indicate you do not want the started task to discover multilevel backups automatically.

Note: Regardless of the value you specify for this keyword, the batch job may be run as often as needed.

MLB_DISCOVER_INTERVAL=1800
Use this keyword to specify the number of 2-second intervals you want to elapse between attempts to discover any new multilevel backups. This keyword is ignored if AUTO_DISCOVER_MULTILVL_BKUPS=N.

Note: Setting this value too low might cause unacceptable BKMSTSKA performance results.

WAIT_FOR_CURRENT_BACKUPS=N
Specify Y to indicate you want the BKMSTSKA started task to wait for current running backups to finish before proceeding to full shut down. In that case a WTOR question BKMH0178I will be asked which will allow to cancel the wait at any time.

If N is specified (default value) the BKMSTSKA started task will proceed to full shut down regardless of any current running backups.

Note: Regardless of what value is set, the list of the current running backups will be shown in the log during shut down processing.

:CBTI_DUMP_BACKUP_PROCESS_OPTIONS section keywords

Use the information in this topic to specify appropriate values for the keywords in the :CBTI_DUMP_BACKUP_PROCESS_OPTIONS section of member BKMINI in the product parameter library.

CBTI_FLATFILE_OPTION=VSAM|FLATFILE
Use this keyword to specify the data set type to be used to capture the spin-off data set information created by the Critical Backup Tracking and Inventory intercept modules. If you are using the VSAM option, use the CBTI_FLATFILE_VSAMDS token to specify the VSAM data set name. If you are using the FLATFILE option, use the CBTI_FLATFILE_PREFIX token to specify the sequential data set name prefix.

CBTI_FLATFILE_VSAMDS=*hlq*.VSAMFLAT
Use this keyword to specify the name of the VSAM data set used to capture the spin-off data created by the Critical Backup Tracking and Inventory intercept modules.

The BKMSTSKA started task and user IDs associated with backup jobs to be tracked must have ALTER authority for this data set.

To reduce Critical Backup Tracking and Inventory serialization delays, it is highly recommended that you use the VSAM Record Level Sharing (RLS) option for the VSAM spin-off data set. Refer to the topic describing the :SI040_VALUES keywords for more information about using the RLS option.

Over time, the high volume record creation and deletion usage characteristics of the VSAM flat file can result in the VSAM cluster containing a large number of dead Control Areas (empty, but not reusable), which in turn, causes the VSAM cluster to grow larger than necessary. To address this issue, you can do either of the following:

- Reorganize the VSAM cluster by doing either of the following:
  - Stop all of the BKMSTSKA started tasks and use IDCAMS REPRO.
  - Use the Advanced Backup and Recovery for z/OS dynamic reorganization feature (refer to the BKMREORG member in the product JCL library for more information). This approach does not require you to stop all the BKMSTSKA started tasks.
- Use the CA Reclaim option for the VSAM cluster. Contact your storage administrator for more information about defining VSAM clusters with the CA Reclaim attribute.

Note: This keyword is used when CBTI_FLATFILE_OPTION=VSAM and it is ignored when CBTI_FLATFILE_OPTION=FLATFILE.

CBTI_FLATFILE_PREFIX=?
Use this keyword to specify the *prefix.datasetname* qualifiers used for sequential spin-off data sets created by the BKMSTSKA started task. One spin-off data set is created for each backup tracked by the started task.

The prefix you specify must adhere to the following rules:

- The prefix name should be unique because Critical Backup Tracking and Inventory checks each sequential flat file matching the prefix name, and if the file does not contain a Critical Backup Tracking and Inventory version record, Critical Backup Tracking and Inventory sends a message and then skips the file. Critical Backup Tracking and Inventory reissues the message after an hour has elapsed. If you use a unique prefix name, the only time you will get this message is if something is found in error.
- The last two nodes of the sequential data set name are reserved for the date and time information in the format `Dyyyyddd.Thhmmsst`.
- The *prefixdatasetname* qualifiers cannot exceed 26 characters in length.
- Select the *prefixdatasetname* qualifiers that allow these data sets to go to temporary volumes because they will be deleted after being processed. In addition, the BKMSTSKA started task and user IDs associated with backup jobs to be tracked must have ALTER authority to data sets with this prefix.
- The spin-off sequential flat files cannot be on volumes that will be enqueued by any type of backup process or they will be bypassed by Critical Backup Tracking and Inventory.

CBTI_FLATFILE_SPACE=?
Use this keyword to specify the allocation attributes (250 character limit) for the Critical Backup Tracking and Inventory sequential flat files. These are the supported attributes:

- `TRACKS|CYLINDERS|BLOCK|AVBLOCK`–DASD quantity type for the allocation request.
- `SPACE(`*`primary secondary`*`)`–DASD primary and secondary quantities for the allocation request.
- `RELEASE`–Release overallocated space.
- `VOLUME(`*`volserlist`*`)`–List of target volumes for the allocation request.
- `DATACLAS(`*`dataclass`*`)`–DFSMS Data Class name for the allocation request.
- `STORCLAS(`*`storageclass`*`)`–DFSMS Storage Class name for the allocation request.
- `MGMTCLAS(`*`managementclass`*`)`–DFSMS Management Class name for the allocation request.

Example:

```
CBTI_FLATFILE_SPACE= TRACKS SPACE(10 10) RELEASE
```

Note: This keyword is used when `CBTI_FLATFILE_OPTION=FLATFILE` and it is ignored when `CBTI_FLATFILE_OPTION=VSAM`.

CBTI_FLATFILE_UNIT=?
Use this keyword to specify the device unit type (3390, SYSALLDA, and so forth) for the Critical Backup Tracking and Inventory sequential flat files.

EXPIRE_BAD_RC=n
Use this keyword to specify the minimum backup step return code that causes the BKMEXPIRE function to remove Critical Backup Tracking and Inventory tracked backups from the Inventory Data Set. This mechanism provides an automatic method of removing unwanted backups, based on the step return code.

CHANGED_DATASET_TRACKING=Y|N
Use this keyword to specify whether you want Critical Backup Tracking and Inventory to track changes that occur against input data sets to a backup.

- Y–Track changes to the input data sets. Be aware that the DAD started task and the DAD data base must be set up before tracking can begin.

  Note: Data sets from full volume physical dumps are excluded from the change tracking and are not added to the DAD.
- N–Do not track changes to the input data sets. This is the default value for this keyword.

CATLG_DATA_FOR_PHY_DUMPS=Y|N
Use this keyword to specify how you want Critical Backup Tracking and Inventory to process data sets backed up by physical dumps.

- Y–Record the data sets and extract catalog data for data sets backed up by physical dumps.
- N–Record the data sets but do not extract catalog data for data sets backed up by physical dumps. This is the default value for this keyword.

CATLG_DATA_FOR_LOG_BKUPS=Y|N
Use this keyword to specify how you want Critical Backup Tracking and Inventory to process data sets backed up logically by ADRDSSU or FDR.

- Y–Record the data sets and extract catalog data for data sets backed up logically by ADRDSSU or FDR.
- N–Record the data sets but do not extract catalog data for data sets backed up logically by ADRDSSU or FDR. This is the default value for this keyword.

CBTI_HRDCPY_MSGS_ONLY=Y|N
Use this keyword to control where Critical Backup Tracking and Inventory tracking messages are written. By default, messages are written to both the system hardcopy log and to the batch job spool output. Use this keyword to prevent the messages from being written to the batch job spool output.

- Y–Both log data and intercept messages are written to the hardcopy console log only.
- N–Job step CBTIPRINT is allocated and used for log data; intercept messages are written to the job JESMSGLG file.

:CBTI_BACKUP_END_OPTIONS section keywords

Use the information in this section to specify appropriate values for the keywords in the :CBTI_BACKUP_END_OPTIONS section of member BKMINI in the product parameter library.

BKUPEND_REQUIRED=N|W|E
Use this keyword to specify the relationship between the Automated Critical Data Identification Application End and Critical Backup Tracking and Inventory Backup End functions.

- N—Backup End for the current application cycle is not required before the next application end processes.
- W—Backup End for the current application cycle is required before the next application end processes. If Backup End was not run, application end executes and produces return code 4.
- E—Backup End for the current application cycle is required before the next application end processes. If Backup End was not run, application end terminates and produces return code 8.

BACKUP_CYCLES_TO_RETAIN=*n*
Use this keyword to specify the number of Automated Critical Data Identification Application End and Critical Backup Tracking and Inventory Backup End cycles are to be retained in the database. You may specify an integer in the range 0-999 (inclusive).

APPL_ONLY_BACKUPS=Y|N
Use this keyword to specify which critical data set backups are to be recorded by the Critical Backup Tracking and Inventory Backup End process.

- Y—Use backups produced by application jobs only. Application backup jobs are defined using the Automated Critical Data Identification BACKUP JOB LIST option on the Application AutoBuild Options panel.
- N—Use any backups recorded by Automated Critical Data Identification.

INCLUDE_HSM=Y|N
Use this keyword to specify whether the Critical Backup Tracking and Inventory Backup End process is to examine the DFSMShsm Incremental Backup inventory.

- Y—Use Incremental Backups produced by DFSMShsm.

- N—Do not use Incremental Backups produced by DFSMShsm.

INCLUDE_PHYSICAL=Y|N
Use this keyword to specify whether the Critical Backup Tracking and Inventory Backup End process is to examine the full volume physical dump inventory.

- Y—Use the full volume physical dump inventory.
- N—Do not use the full volume physical dump inventory.

EXCLUDE_ACCOMPANY=Y|N
Use this keyword to specify whether the Critical Backup Tracking and Inventory Backup End process is to exclude backups for critical data sets listed in the Automated Critical Data Identification ACCOMPANY list.

- Y—Exclude backups for critical data sets listed in the ACCOMPANY list.
- N—Do not exclude backups for critical data sets listed in the ACCOMPANY list.

EXCLUDE_ALLOCATE=Y|N
Use this keyword to specify whether the Critical Backup Tracking and Inventory Backup End process is to exclude backups for critical data sets listed in the Automated Critical Data Identification ALLOCATE list.

- Y—Exclude backups for critical data sets listed in the ALLOCATE list.
- N—Do not exclude backups for critical data sets listed in the ALLOCATE list.

BAD_BKUP_RC=4|8
Use this keyword to specify the lower limit return code value that indicates a bad backup was detected by Critical Backup Tracking and Inventory Backup End process. If one or more current backups ended with a return code equal to or greater than this value, an error message is produced by the Backup End process.

NO_BKUP_FOUND_RC=0|4|8
Use this keyword to specify whether the return code value you want produced by the Critical Backup Tracking and Inventory Backup End process when a backup was not found for one or more critical data sets.

- 0—When a backup is not found, produce a return code of 0 and an informational message.
- 4—When a backup is not found, produce a return code of 4 and a warning message.
- 8—When a backup is not found, produce a return code of 8 and an error message.

CHANGED_SINCE_LAST_BKUP_RC=0|4|8
Use this keyword to specify the return code you want to indicate a situation in which BKUPEND finds a data set identified as critical by Automated Critical Data Identification has an associated backup in Critical Backup Tracking and Inventory, but the data set has changed since the last backup.

:CBTI_RESTORE_OPTIONS section keywords

Use the information in this section to specify appropriate values for the keywords in the :CBTI_RESTORE_OPTIONS section of member BKMINI in the product parameter library.

NEWHLQ=?
Use this keyword to specify the high-level qualifier to use for all data sets being restored.

PROC_NAME=BKMRPROC
Use this keyword to specify the name of the JCL procedure for Critical Backup Tracking and Inventory RESTORE submission. The default value for this keyword is BKMRPROC.

JCLOUT_ALLOC1='text', JCLOUT_ALLOC2='text', JCLOUT_ALLOC3='text'
Use these keywords to specify the allocation parameters for the new JCLOUT file used in the RESTORE JCL submitted through the ISPF panels.

The text string you specify for each keyword cannot exceed 50 characters in length. Collectively, the content of these three text stings must adhere to standard JCL syntax rules.

You can also modify the JCL for a Critical Backup Tracking and Inventory RESTORE job manually, prior to submitting the job for execution.

The default values for these keywords are as follows:

```
JCLOUT_ALLOC1='DISP=(NEW,CATLG,DELETE),UNIT=SYSALLDA,'
JCLOUT_ALLOC2='SPACE=(CYL,(1,1,50),RLSE),DSNTYPE=LIBRARY,'
JCLOUT_ALLOC3='DCB=(RECFM=FB,LRECL=80,BLKSIZE=0,DSORG=PO'
```

INCLUDE_PHYSICAL_DUMPS=Y|N
Use this keyword to specify whether you want physical dumps included when building data set RESTORE jobs.

- Y–Include data sets from full volume dumps during data set RESTOREs.
- N–Do not include data sets from full volume dumps during data set RESTOREs. This is the default value for this keyword. Be aware that specifying N for this keyword has no effect on RESTORE VOL(***).

SKELLIB_DEFAULT_ADRDSSU_PD=BKMDSSPD
Use this keyword to specify the skeleton JCL default for DFSMSdss physical dump by data set.

SKELLIB_DEFAULT_ADRDSSU_LD=BKMDSSLD
Use this keyword to specify the skeleton JCL default for DFSMSdss logical data set dump.

SKELLIB_DEFAULT_ADRDSSU_ZFS=BKMDSSLZ
Use this keyword to specify the skeleton JCL default for DFSMSdss Unix Files Backup.

SKELLIB_DEFAULT_ADRDSSU_FVD=BKMDSSFD
Use this keyword to specify the skeleton JCL default for DFSMSdss full volume dump.

SKELLIB_DEFAULT_FDR_FVD=BKMFDRVD
Use this keyword to specify the skeleton JCL default for FDRFVD.

SKELLIB_DEFAULT_FDR_ABR_FDR=BKMFDRFD
Use this keyword to specify the skeleton JCL default for FDRABRFDR.

SKELLIB_DEFAULT_FDR_ABR_ABR=BKMFDRAB
Use this keyword to specify the skeleton JCL default for FDRABR.

SKELLIB_DEFAULT_FDR_ABR_DSF=BKMFDRDS
Use this keyword to specify the skeleton JCL default for FDRDSF.

SKELLIB_DEFAULT_FDR_ABR_APPL=BKMFDRAP
Use this keyword to specify the skeleton JCL default for FDRAPPL.

SKELLIB_DEFAULT_FDR_DSF=BKMFDRDS>
Use this keyword to specify the skeleton JCL default for FDRDSF.

SKELLIB_DEFAULT_CFCAMS_REP=BKMREPRO
Use this keyword to specify the skeleton JCL default for CFCAMS REPRO.

SKELLIB_DEFAULT_CFCAMS_EXP=BKMIMPRT
Use this keyword to specify the skeleton JCL default for CFCAMS EXPORT.

SKELLIB_DEFAULT_IDCAMS_REP=BKMREPRO
Use this keyword to specify the skeleton JCL default for IDCAMS REPRO.

SKELLIB_DEFAULT_IDCAMS_EXP=BKMIMPRT
Use this keyword to specify the skeleton JCL default for IDCAMS IMPORT.

SKELLIB_DEFAULT_IEBGENER=BKMIEBGN
Use this keyword to specify the skeleton JCL default for IEBGENER.

SKELLIB_DEFAULT_ICEGENER=BKMICEGN
Use this keyword to specify the skeleton JCL default for ICEGENER.

SKELLIB_DEFAULT_SYNCGENR=BKMSYNCG
Use this keyword to specify the skeleton JCL default for SYNCGENR.

SKELLIB_DEFAULT_IEBCOPY=BKMIEBCP
Use this keyword to specify the skeleton JCL default for IEBCOPY.

SKELLIB_DEFAULT_HSMAUTO=BKMAUTOD
Use this keyword to specify the skeleton JCL default for DFSMShsm AUTODUMP.

SKELLIB_DEFAULT_SORT=BKMSORT
Use this keyword to specify the skeleton JCL default for all supported SORT products.

SKELLIB_DEFAULT_ABARS=BKMABRS1|BKMABRS2
Use this keyword to specify the skeleton JCL default for ABARS Manager recoveries.

- BKMABRS1—Use this value to restore from a relative backup version.
- BKMABRS2–Use this value to restore from a specific backup version.

SKELLIB_DEFAULT_ARCINBAK=BKMARCBK
Use this keyword to specify the skeleton JCL default for DFSMShsm ARCINBAK.

SKELLIB_DEFAULT_CR+_RECOVERBCS=BKMBKBCS
Use this keyword to specify the skeleton JCL default for CR+ RECOVER BCS.

SKELLIB_DEFAULT_CR+_RECOVERDSN=BKMBKDSN
Use this keyword to specify the skeleton JCL default for CR+ RECOVER DSN.

SKELLIB_DEFAULT_CR+_RECOVERVVDS=BKMBKVVD
Use this keyword to specify the skeleton JCL default for CR+ RECOVER VVDS.

SKELLIB_DEFAULT_CRK_RECOVERBCS=BKMBKBCK
Use this keyword to specify the skeleton JCL default for CKM BCS restore from CKM BACKUP DSN.

SKELLIB_DEFAULT_CRK_RECOVERDSN=BKMBKDSK
Use this keyword to specify the skeleton JCL default for CKM Data Set restore from CKM BACKUP DSN.

SKELLIB_DEFAULT_CRK_RECOVERVVDS=BKMBKVVK
Use this keyword to specify the skeleton JCL default for CKM VVDS restore from CKM BACKUP VVDS.

SKELLIB_DEFAULT_CRR_RECOVERBCS=BKMBKBCR
Use this keyword to specify the skeleton JCL default for MCR BCS restore from MCR BACKUP DSN.

SKELLIB_DEFAULT_CRR_RECOVERDSN=BKMBKDSR
Use this keyword to specify the skeleton JCL default for MCR Data Set restore from MCR BACKUP DSN.

SKELLIB_DEFAULT_CRR_RECOVERVVDS=BKMBKVVR
Use this keyword to specify the skeleton JCL default for MCR VVDS restore from MCR BACKUP VVDS.

SKELLIB_DEFAULT_FAVER=BKMFAVR1
Use this keyword to specify the skeleton JCL default for CA FAVER.

SKELLIB_DEFAULT_ADAFULL=BKMADBS1
Use this keyword to specify the skeleton JCL default for Adabas backup (full).

SKELLIB_DEFAULT_ADADELTA=BKMADBS2
Use this keyword to specify the skeleton JCL default for Adabas backup (incremental).

SKELLIB_DEFAULT_ADALOGC=BKMNORES
Use this keyword to specify the skeleton JCL default for Adabas PLOG COPY backups.

SKELLIB_DEFAULT_ADRDSSU_FVC=BKMDSSFC
Use this keyword to specify the skeleton JCL default for DFSMSdss Full Volume Copy.

SKELLIB_DEFAULT_ADRDSSU_FC=BKMDSSPC
Use this keyword to specify the skeleton JCL default for DFSMSdss Physical Copy.

SKELLIB_DEFAULT_ADRDSSU_LC=BKMDSSLC
Use this keyword to specify the skeleton JCL default for DFSMSdss Logical Copy.

SKELLIB_DEFAULT_ADRDSSU_CD=BKMDSSCD
Use this keyword to specify the skeleton JCL default for DFSMSdss Copy Dump.

SKELLIB_DEFAULT_USERDEF=BKMGUTL1
Use this keyword to specify the skeleton JCL default for User Defined Backups.

SKELLIB_DEFAULT_ICETOOL=BKMICETL
Use this keyword to specify the skeleton JCL default for ICETOOL.

SKELLIB_DEFAULT_VSAMASST=BKMVASST
Use this keyword to specify the skeleton JCL default for VSAM Assist.

SKELLIB_DEFAULT_CADISK=BKMCADSK
Use this keyword to specify the skeleton JCL default for CA Disk.

SKELLIB_DEFAULT_FRZDSNB=BKMFRED1
Use this keyword to specify the skeleton JCL default for Fast Replication Services generated restore jobs.

:ACDI section keywords

Use the information in this topic to set appropriate values for the keywords in the :ACDI section of BKMINI.

CYCLE_CONTROL=APPLEND|JOB
Use this keyword to specify the default Automated Critical Data Identification Setup option to determine how the application end process is performed. Specify APPLEND if you want to use an application end job submitted by the scheduler after the last job completes for a cycle. APPLEND is the default value for this token.

Specify JOB if you want an application end job submitted automatically by the BKMSTSKR started task whenever an application cycle end job completes. Use the APPLEND_JOBn tokens to specify the job card format to be used for the application end job submitted by the BKMSTSKR started task.

APPLEND_JOB1=
Use this keyword to specify card image one for the job card you want the BKMSTSKR started task to use when submitting an application end job.

APPLEND_JOB2=
Use this keyword to specify card image two for the job card you want the BKMSTSKR started task to use when submitting an application end job.

APPLEND_JOB3=
Use this keyword to specify card image three for the job card you want the BKMSTSKR started task to use when submitting an application end job.

APPLEND_JOB4=
Use this keyword to specify card image four for the job card you want the BKMSTSKR started task to use when submitting an application end job.

HDU_DAYS=*n*
Use this keyword to specify the maximum number of days that you want SMF data stored in the Historical Dataset Usage database. The value you specify for this parameter must be an integer in the range 1-90, inclusive. The default value for this parameter is 90 days.

Note: This token is ignored unless token RSP_Start_HDU_Task=YES.

APPLID=SIS
Use this keyword to specify the TSO APPLID. You can change the default value (SIS) to meet your organization's requirements, but it is recommended that you keep the default value.

MCDS_ERROR_RC=0|4|8
Use this keyword to specify the return code you want generated in situations where one or more inquires to the MCDS fail. The value assigned to this keyword is meaningless unless you are using DFSMShsm for space management.

- 0–MCDS error is ignored. This is the default value for this keyword.
- 4–Causes a warning SELECTION DATA SET BUILT WITH ERRORS to be issued.
- 8–The Selection Data Set will not be built.

CA7ALTSCHID=1|nnn
Use this keyword to specify the default alternative schedule ID value used by Unicenter CA-7 for job collection. You can specify any value in the range 1-255, inclusive. The default value for this keyword is 1.

CA7COLLECTOPTIONS=AutoSave(MER|REP) UseCA7Logon(YES|NO) NonSchJobs(YES|NO) RetainPrtFile(YES|NO)
Use this keyword to specify the appropriate Unicenter CA-7 collection event options.

- `AutoSave(MER|REP)`–Specify how you want the Unicenter CA-7 jobs and anchor structures handled for each collection event:
  - MER–Current or new (or both) job elements are merged into the Automated Critical Data Identification database.
  - REP–Current job elements collected replace the existing entries within the Automated Critical Data Identification database. This is the default value for the AutoSave keyword.
- `UseCA7Logon(YES|NO)`–Specify whether you want the CA-7 /LOGONNAME and /LOGONPASS statements included as part of the job collection control statements. By default, Automated Critical Data Identification includes this statement to satisfy Unicenter CA-7 security checking for the Batch Terminal logon.
  - YES–The values for /LOGONNAME and /LOGONPASS are used within the Batch CA-7 control input as a /LOGON statement.
  - NO–If your version of Unicenter CA-7 does not require the /LOGON statement, you can specify NO to exclude the /LOGON statement.

  Note: The Unicenter CA-7 job scheduling product input panel requires that values be provided for the **CA7 User ID Name** field and **CA7 Password** field, but the entries made in these fields are ignored in the batch statement stream if you specify UseCA7Logon(NO).
- `NonSchJobs(YES|NO)`–Use this keyword to specify whether you want the unscheduled Unicenter CA-7 jobs and job anchor structures included in the final job list for all applications using Unicenter CA-7 as their job scheduling facility.
  - YES–Include the unscheduled jobs with the existing scheduled jobs. This is the default value for the NonSchJobs keyword.
  - NO–Do not include the unscheduled jobs with the existing scheduled jobs; collect the existing scheduled jobs only.

  Note: If you want to include all unscheduled jobs but exclude specific jobs by job name, use the **Job Name Filtering** screen to specify the job names you want excluded.
- `RetainPrtFile(YES|NO)`–A batch Unicenter CA-7 job collection event is a 2-step job: the first step invokes a Unicenter CA-7 utility to display the job structure information to a SYSPRINT file, and the second step is the Automated Critical Data Identification analysis and updating operation that uses the SYSPRINT data set name as input. Use this keyword to specify whether you want to retain the batch SYSPRINT data set used for the Unicenter CA-7 job collection event after the event is complete.
  - YES–Retain the data set name assigned to the batch SYSPRINT data set.
  - NO–Delete the batch SYSPRINT data set after the job collection event is complete. This is the default value for the RetainPrtFile keyword.

CA7LOGONNAME=LogonName|UCC7JOBS
Use this keyword to specify the name of the Unicenter CA-7 Batch Logon Userid value that is used if Unicenter CA-7 job collection is initiated. The default value for this keyword is UCC7JOBS.

Note: During the Unicenter CA-7 Job Collection dialog invocation for a specific event execution, you can change the value specified for CA7LOGONNAME.

CA7LOGONPASS=PasswordName|UCC7JOB
Use this keyword to specify the name of the Unicenter CA-7 Batch Logon Password value that is used if Unicenter CA-7 job collection is initiated. The default value for this keyword is UCC7JOB.

Note: During the Unicenter CA-7 Job Collection dialog invocation for a specific event execution, you can change the value specified for CA7LOGONPASS.

CA7PROCNAME=ProcName|CA73BTI
Use this keyword to specify the name of the Unicenter CA-7 Batch Logon Procedure that is used if Unicenter CA-7 job collection is initiated. The default value for this keyword is CA73BTI.

Note: During the Unicenter CA-7 Job Collection dialog invocation for a specific event execution, you can change the value specified for CA7PROCNAME.

CA7PROCSTEPNAME=ProcStepName|BTERM
Use this keyword to specify the name of the Unicenter CA-7 Batch Logon Proc step that is used if Unicenter CA-7 job collection is initiated. The value you specify is used to override the existing stepname for inclusion of the SYSPRINT and SYSIN statements. The default value for this keyword is BTERM.

CA7SCHID=1
This keyword is inactive.

CA_INCLUDE_ALTSCHID=YES|NO
Use this keyword to specify whether you want the CA-7 SCHID specified in CA7ALTSCHID added to the other SCHIDs specified for an application. This is a global token: all applications using Unicenter CA-7 job collection are affected by this token.

- YES–Add the CA7ALTSCHID to other SCHIDs. This is the default value for this keyword.
- NO–No not add the CA7ALTSCHID to other SCHIDs.

CTLM_INPUTUTLJCL=DSName[(Member)]
Use this keyword to specify the name of the data set and, if the data set is partitioned, the name of the member where the CTMRFLW procedure is located. The CTMRFLW procedure is used for CONTROL-M job collection.

For initial job collection events, Automated Critical Data Identification opens and read the input JCL statements and appends any scheduling file data set names (refer to the description of the CTLM_SCHEDFILES keyword for more information).

CTLM_SCHEDFILES=File1(Name1) File2 File3 File4(Name4)
Use this keyword to specify the list of scheduling file data set names, and member names in cases where the file is a PDS, that are used as global values. You can specify multiple data set names, using blank spaces to separate one file name from the next and continuation characters if you need multiple lines to specify all the file names.

If you do not specify a list of file names, the input scheduling files and their associated member names are obtains from the CONTROL-M dialog input screen.

JOBCOLLECTION_PRESTAGEPDS=*|DSName1 JOBCOLLECTION_PRESTAGESEQ=*|DSName2
Use this keyword to identify the correct input job pre-staging input file for job collection. You can specify either data set name value or both data set name values. If you specify both, only the value you specify for the JOBCOLLECTION_PRESTAGEPDS keyword is used.

DSName1 must be a partitioned data set. Specify only the base name. For job pre-staging events, Automated Critical Data Identification does not locate the application name as the member name within the partitioned data set. The contents of the member are used as input to the job pre-staging function, from the CONTROL-M dialog input screen.

DSName2 must be a physical sequential file. For job pre-staging events, Automated Critical Data Identification locates and uses the contents of this data set as input to the job pre-staging function.

JOBCOLL_CA7_UID
Use this keyword to specify that you want a CA-7 /UID,R= token value used during Unicenter CA-7 job collection. If the /UID,R= token has not been assigned a value, the /UID statement is not used.

If you specify this keyword, you need to specify the UseCA7Logon(NO) value for the CA7COLLECTOPTIONS keyword.

JOBCOLL_CTLM_VERSION=Cnnn|C400
Use this keyword to specify the version of CONTROL-M. The default value for this keyword is C400.

Specify C500 for NEW DIMENSION SOFTWARE, LTD., release 5.0.0 or higher.

Specify C606 for BMC Software release 6.0.06 or higher.

JOBCOLL_DROP_ALL_UNRESOLVED_PREDECESSORS=Y|N
Use this keyword to specify what you want done with unresolved predecessors during job collection by these schedulers: Unicenter CA-7, Cybermation ESP, CONTROL-M, TWS, and ASG-Zeke.

- N–Unresolved predecessors are handled individually and the job chain is kept intact. Automated Critical Data Identification retains all the job names for the application. Be aware that processing can take considerable time in situations where multiple schedules containing the same job name are passed to Automated Critical Data Identification. This is the default value for this keyword.
- Y–Unresolved predecessors are dropped in a single pass and the job chain is broken. The job report might indicate some jobs are standalone. If the JOBCOLL_EXCLUDE_DEP=Y, additional EXCL_JOB filters might be required because the job chain was broken. Automated Critical Data Identification retains all the job names for the application.

JOBCOLL_ESP_VERSION=Ennn|E530
Use this keyword to specify the version of Cybermation ESP. The default value for this keyword is E530.

Specify E530 for Cybermation ESP release 5.3.1 or higher.

JOBCOLL_EXCLUDE_DEP=Y|N
Use this keyword to specify whether you want dependent jobs excluded in situations where a predecessor job is excluded by a job collection filter. This keyword applies to jobs collected from CONTROL-M (version 606 and higher) and ASG-Zeke.

- N–Do not exclude dependent jobs in situations where a predecessor job is excluded by a job filter. This is the default value for this keyword.
- Y–Exclude dependent jobs in situations where a predecessor job is excluded by a job filter.

JOBCOLL_ZEKE_ABNORMAL_JOBS=Y|N
Use this keyword to specify whether you want to include in Automated Critical Data Identification job collection those jobs that are scheduled only when a previous ASG-Zeke event has ended abnormally. This keyword is applicable to job collection for ASG-Zeke only.

- N–Do not such jobs. This is the default value for this keyword.
- Y–Include such jobs.

JOBCOLL_ZEKE_JCLLIB=Dsname
Use this keyword to identify the library containing the ZEKEUTL procedure. If ZEKEUTL resides in a system PROCLIB, you do not need to specify a value for this keyword.

JOBCOLL_ZEKE_JCLLIB02=Dsname02
Use this keyword to identify the library containing the ZEKEUTL02 procedure. If ZEKEUTL02 resides in a system PROCLIB, you do not need to specify a value for this keyword.

JOBCOLL_ZEKE_JCLLIB03=Dsname03
Use this keyword to identify the library containing the ZEKEUTL03 procedure. If ZEKEUTL03 resides in a system PROCLIB, you do not need to specify a value for this keyword.

JOBCOLL_ZEKE_VERSION=Znnn|Z510
Use this keyword to specify the version of ASG-Zeke. The default value for this keyword is Z510.

Specify Z450 if your organization uses an earlier release of ASG-Zeke than 510.

Specify Z510 if your organization uses ASG-Zeke release 510 or later.

JOBCOLL_ZEKE_ZEKEUTL=ProcedureName|ZEKEUTL
Use this keyword to specify the name of the ZEKEUTL procedure. ZEKEUTL is the default value for this keyword.

JOBCOLL_ZEKE_ZEKEUTL02=ProcedureName|ZEKEUTL02
Use this keyword to specify the name of the ZEKEUTL02 procedure. ZEKEUT02L is the default value for this keyword.

JOBCOLL_ZEKE_ZEKEUTL03=ProcedureName|ZEKEUTL03
Use this keyword to specify the name of the ZEKEUTL03 procedure. ZEKEUTL03 is the default value for this keyword.

JOBTRACSCHEDOUT=DSName[(Member)]
Use this keyword to specify the name (as a data set name and, optionally for partitioned data sets, a member name) of the complete previously processed GJTRMAIV output SYSPRINT file to be used as input to the Unicenter CA-Jobtrac job collection event, in place of the standard GJTRMAIV JCL statements. The data set name and member name (if any) you specify for this keyword is used directly in the AMPCMD control statements.

There is no default value for this keyword.

Note: If you specify a value for this keyword, Automated Critical Data Identification opens, reads, and processes the file identified by its value, which negates the Unicenter CA-Jobtrac History JCL step from which the GJTRMAIV JCL statements normally are constructed prior to the Automated Critical Data Identification job collection step.

PERMFILESQUAL=PrefixValue|USERPREFIX
Use this keyword to specify the high order prefix value qualifier you want use to name the permanent application reference files allocated for use during Automated Critical Data Identification job collection and auditing events.

If you specify USERPREFIX, the individual user's prefix value is used. This is the default value for this keyword.

Some data sets are created dynamically within Automated Critical Data Identification. If you set PERMFILESQUAL=USERPREFIX, the user ID running Automated Critical Data Identification is used as the high level qualifier for these data sets. However, these files are also touched by the RSPUPDAT task in the process of building a Selection Data Set. If RSP is being used, PERMFILESQUAL must be set to the high level qualifier you want used for these data sets instead of a user ID being associated with the started task.

PERMFILESMC=MCValue PERMFILESSC=SCValue PERMFILESDC=DCValue PERMFILESVL=VolumeName
Use the PERMFILESMC, PERMFILESSC, and PERMFILESDC keywords to specify the SMS management class, storage class, and data class names to be used for allocation of permanent or intermediate application files. Use the PERMFILESVL keyword to specify the volume names used for the VOLUME parameter of the ALLOC command. There are no default values for these keywords.

Note: If you do not specify values for the PERMFILESMC, PERMFILESSC, and PERMFILESDC keywords, SYSALLDA is used as the device type name. You can change the device type name by assigning a different value to the DEFAULTUNIT keyword.

BASEAPL_PDSNAME=*|hlq
Use this keyword to specify the high level qualifiers you want used to form the partitioned data set for application information. The members belonging to this partitioned data set contain last event information for the application, defined by the applname qualifier.

Specify * if you want the default used: USERPREFIX.AMPACDI.*applname*.ENTRIES, where USERPREFIX takes its value from the PERMFILESQUAL keyword value, and the lower level qualifiers are reserved and will be assigned by Automated Critical Data Identification when the application is created.

If you specify a high level qualifier other than *, it is appended to applname.ENTRIES. For example, if you specify BASEAPL_PDSNAME=PROD.ZSELCC, the resulting data set name will be PROD.ZSELCC.*applname*.ENTRIES.

Note: You can specify a multiple level high level qualifier if the total length of the fully qualified data set name does not exceed 44 characters in length, including the periods (.).

BASEAUDITING_PDSNAME=*|hlq
Use this keyword to specify which high level qualifiers you want used to form the name of the partitioned data set for application Auditing entries. This allows all Auditing results for any application to be maintained in a separate file. The members belonging to the partitioned data set will have the

names of the applications and will contain the most recent results from a job, JCL, or an SMF Audit event (or some combination of these).

Specify * if you want the default used: USERPREFIX.AMPACDI.AUDITING.ENTRIES, where USERPREFIX takes its value from the PERMFILESQUAL keyword value, and the lower level qualifiers are reserved and will be assigned by Automated Critical Data Identification when the application is audited for the first time.

If you specify a high level qualifier other than *, it is appended to the reserved lower level qualifiers.

Note: You can specify a multiple level high level qualifier if the total length of the fully qualified data set name does not exceed 44 characters in length, including the periods (.).

BASEAUDITHIST_PDSNAME=*|hlq
Use this keyword to specify which high level qualifiers you want used to form the name of the partitioned data set for application Auditing History entries. This allows all Auditing History results for any application to be maintained in a separate file. The members belonging to the partitioned data set will have the names of the applications and will contain the most recent results from a job, JCL, or an SMF Audit event (or some combination of these).

Specify * if you want the default used: USERPREFIX.AMPASAP.AUDITING.HISTORY, where USERPREFIX takes its value from the PERMFILESQUAL keyword value, and the lower level qualifiers are reserved and will be assigned by Automated Critical Data Identification when the application is audited for the first time.

If you specify a high level qualifier other than *, it is appended to the reserved lower level qualifiers.

Note: You can specify a multiple level high level qualifier if the total length of the fully qualified data set name does not exceed 44 characters in length, including the periods (.).

BASEAOVRLP_PDSNAME=*|hlq
Use this keyword to specify which high level qualifiers you want used to form the name of the partitioned data set for application data set Overlap information. The members belonging to the partitioned data set will have the names of the applications and will contain the most recent overlap data set event information for the application.

Specify * if you want the default used: USERPREFIX.AMPASAP.OVRLAP.ENTRIES, where USERPREFIX takes its value from the PERMFILESQUAL keyword value, and the lower level qualifiers are reserved and will be assigned by Automated Critical Data Identification when the application is created.

If you specify a high level qualifier other than *, it is appended to the reserved lower level qualifiers.

Note: You can specify a multiple level high level qualifier if the total length of the fully qualified data set name does not exceed 44 characters in length, including the periods (.).

BASESELBAKUP_PDSNAME=*|hlq
Use this keyword to specify which high level qualifiers you want used to form the name of the partitioned data set for application data set backup copies. The members belonging to the partitioned data set will have the names of the applications and will contain the previous contents of their Selection Data Set information, prior to the current update.

This data set is used only in situations where the VERSELBACKUP feature was enabled for the application during final verification processing. When VERSELBACKUP is enabled, a copy of the application's Selection Data Set contents are written to this data set prior to any changes being made.

Specify * if you want the default used: USERPREFIX.AMPASAP.SELCTDSN.BACKUPS, where USERPREFIX takes its value from the PERMFILESQUAL keyword value, and the lower level qualifiers are reserved and will be assigned by Automated Critical Data Identification when the application is created.

If you specify a high level qualifier other than *, it is appended to the reserved lower level qualifiers.

Note: You can specify a multiple level high level qualifier if the total length of the fully qualified data set name does not exceed 44 characters in length, including the periods (.).

DEFAULTUNIT=SYSALLDA|UnitName
Use this keyword to specify the default generic or esoteric unit name to be used for intermediate, nonpermanent, nontemporary data sets. If SMS information is provided, the value assigned to this keyword is not used. The default value for this keyword is SYSALLDA.

MAXPENDMSG=10|Number
This keyword is not used by this release of Advanced Backup and Recovery for z/OS. Do not change the default value assigned to this keyword.

MAXLOGMSGS=100|Number
Use this keyword to specify how many event log messages you want retained for display by an application user. Event log messages are retained in the Automated Critical Data Identification database file until this maximum number is exceeded, at which point records prior to the current date are deleted until there enough available entries to handle the current request.

The number you specify must an integer in the range 1-999, inclusive. If you specify zero, the event log recording facility is disabled. The default value for this keyword is 100.

VERSELCOPY=YES|NO|1|0
Use this keyword to specify whether you want a copy of the current Selection Data Set entries made in addition to the normal verification update process.

- YES–Create an additional copy of the Selection Data Set entries, having the name USERPREFIX.AMPASAP.SELCTDSN.CURRENT, unless you specify a nondefault value for the BASESELCOPY_PDSNAME keyword. USERPREFIX is the value assigned to the PREMFILESQUAL keyword. As subsequent Selection Data Set updates are made, Automated Critical Data Identification overwrites the previous selection copy data set.
- NO–Do not create an additional copy of the Selection Data Set entries. This is the default value for this keyword.

VERSELBACKUP=YES|NO|1|0
Use this keyword to specify whether you want a copy of the previous Selection Data Set entries made prior to replacing the entries.

- YES–Create an additional copy of the selection backup data set entries, having the name USERPREFIX.AMPASAP.SELCTDSN.BACKUP, unless you specify a nondefault value for the BASESELBAKUP_PDSNAME keyword. USERPREFIX is the value assigned to the PREMFILESQUAL keyword. As subsequent Selection Data Set updates are made, Automated Critical Data Identification overwrites the previous selection backup data set. This is the default value for this keyword.
- NO–Do not create an additional copy of the selection backup data set entries.

VERCTLGCHECK=YES|NO|1|0
Use this keyword to specify whether you want Automated Critical Data Identification to check each included entry against the catalog before updating the application's Selection Data Set. Specifying YES or 1 for this keyword ensures that only cataloged entries exist in the Selection Data Set when the ABACKUP event occurs.

- YES–Check each included entry against the catalog before updating the application's Selection Data Set. This is the default value for this keyword.
- NO–Do not perform a catalog check before updating the application's Selection Data Set. If the application's unique data set entries were collected from SMF data sets that were created from a different system, specifying NO forces those entries to be included because they are not cataloged on the current system.

VERTAPE_INCLUDESIZE =NONE|nnnnnn|[MB]
Use this keyword to specify the maximum threshold value (in terms of megabytes) you want Verification processing to use when determining whether a user tape data set should be placed in the Selection Data Set as an INCLUDE entry or as an ACCOMPANY entry. This keyword is relevant only in situations where the application has a backup type of ABARS.

- NONE–No additional user tape size checking occurs. This is the default value for this keyword.

- *nnnnnn*–The threshold size (expressed in terms of whole megabytes) against which the user's tape data set size is compared. If the tape data set size is less than the specified threshold, the data set is marked as an INCLUDE entry; if the tape data set size exceeds the threshold, the data set is marked as an ACCOMPANY entry.

Note: You can use the Verification section of the **Global Application SETUP Options** panel to specify and change threshold values that are unique to individual applications.

VERDSN_OPTIONS=ALLOWUSERADDS(YES|NO|1|0) INCSELECTDSN(YES|NO|1|0) INCINFODSN(YES|NO|1|0)

Use this keyword to specify the options that determine how the backup list is handled during the final verification processing.

- ALLOWUSERADDS–Specify a value for this option to determine whether Automated Critical Data Identification verification processing should automatically include additional data sets to be backed up during final Automated Critical Data Identification processing.
  - YES or 1–Allow Automated Critical Data Identification to include automatically additional data sets to be backed up during final processing. This is the default value for this option.
  - NO or 0–Do not allow Automated Critical Data Identification to include automatically additional data sets to be backed up during final processing.
- INCSELECTDSN–Use this option to specify whether you want the application's Selection Data Set included in the backup list during final verification processing. Using this option is global for all current applications. YES is the default value for this option.
  - YES or 1–Include the application's Selection Data Set in the backup list. The Selection Data Set name will be either the default Automated Critical Data Identification name, or if ABARS is used, the ABARS aggregate's Selection Data Set name. If your current Automated Critical Data Identification applications were defined within sequential files, specifying YES for this option forces the Selection Data Set names to be included in the final backup list. However, if partitioned data set member names were used to contain individual application selection lists, specifying YES for this option forces the partitioned data set name into the backup list for each application, which creates duplicate (overlapped) critical data sets.
  - NO or 0–Do not include the application's Selection Data Set in the backup list.
- INCINFODSN–Specify a value for this option to determine whether the application's information data set is included in the backup list during final verification processing. The information data set contains previous history data and JCL members that were used to create the application, which might be useful for disaster recovery purposes. Using this option is global for all current applications. YES is the default value for this option.
  - YES or 1–Include the application's information data set in the backup list during final verification processing.
  - NO or 0–Do not include the application's information data set in the backup list during final verification processing.

CATLG_LOOKUP_ERROR_ACTION=NONE|NOT_CAT|ERR_LIST

Use this keyword to specify what action you want taken when Automated Critical Data Identification encounters a catalog lookup error.

- NONE–Take no action. The program ends with rc=8; other return codes are ignored. This is the default value for this keyword.
- NOT_CAT–The data set in error appears as not cataloged and processing continues, if CATLG_LOOKUP_ERROR_RC=0 OR 4 was specified.
- ERR_LIST–The data set appears as not cataloged, but its name is included in a separate list in the excluded data sets portion of the Selection Data Set showing the catalog lookup error. The error list is created if CATLG_LOOKUP_ERROR_RC=0 OR 4. Processing stops if RC=8.

CATLG_LOOKUP_ERROR_RC=n

Use this keyword to specify the return code you want supplied when Automated Critical Data Identification encounters a catalog lookup error.

APPL_RECOVERY_PLAN_DSN=*appl_recov_plan_dsname*
This release of Advanced Backup and Recovery for z/OS supports the use of an optional application recovery plan data set. This data set contains one member for each application recovery plan; each member has the same name as the application whose recovery plan it contains. Each recovery plan contains whatever information is deemed necessary to recover that one application.

- If you do not intend to use application recovery plans, leave this token blank.
- If you do want to use application recovery plans, you must create a PDS or PDSE and then specify that data set name for this keyword. There are no restrictions on the attributes that you use for this data set, but it must be ISPF EDIT accessible. A default template (*hlq*.SBKMEXEC(BKMAPPRV)) is used to create the application recovery plan.

  You cannot modify the default template, but you can use the default template to create your own template. The customized template must reside in member RHEAD in *hlq.appl_recov_plan_dsname*.

SDS_NAME_PREFIX=prefix
Use this keyword to specify the default prefix for the Selection Data Set name to be used when defining new Automated Critical Data Identification applications. To construct the SDS name, the SDS name prefix is followed by the application name and SELECT qualifiers. The prefix can have a maximum of 26 characters.

SDS_FORMAT= data mover type
Use this keyword to specify the format of the Selection Data Set contents. Valid format types are ABARS, DSS, DSSL, LIST, DMS, FDR, or CA Disk. If you do not assign a format type to this keyword, it takes the default value (ABARS).

SDS_EVAL_JCL_GDG=Y|N
Use this keyword to specify whether you want GDG entries from JCL data evaluated.

- Y–Evaluate the GDG entries from JCL data. The relative GDS names are converted to absolute names based on the setting for **GDG names in Selection Data Set**. This is the default value for this keyword.
- N–Do not evaluate the GDG entries from JCL data.

Note: You can specify Y for this keyword and for the SDS_EVAL_SMF_GDG keyword.

SDS_EVAL_SMF_GDG=Y|N
Use this keyword to specify whether you want GDG entries from RSP evaluated.

- Y–Evaluate the GDG entries from RSP. The absolute GDS names of the RSP SMF entries are converted to relative names, based on the setting for **GDG names in Selection Data Set**. This is the default value for this keyword.
- N–Do not evaluate the GDG entries from RSP.

Note: You can specify Y for this keyword and for the SDS_EVAL_JCL_GDG keyword.

SDS_GDG_ADD_MISSING=MINIMUM|NO|ALLCATALOGED|FORWARD
Use this keyword to specify whether you want GDG entries other than those already in the database added.

- MINIMUM–All generations from the oldest referenced through the current generation (0 after the application has completed) are added to the database, if they were not already present. If any gaps exist in the range of generations referenced, the relative generations are not correct after a restore (assuming that GDG relative notation is used in JCL DD statements). This is the default value for this keyword.

  Note: The default evaluation of new generations is EXCLUDE, which is consistent with Automated Critical Data Identification including only input data sets. During a rerun, if an application is sensitive to the knowledge that new generations were indeed created, it might be necessary for these new generations to be restored. See the SDS_GDG_MISSING_CATEGORY option for more information about including or allocating new generations.

- NO–Do not add additional GDG entries to the database. When it is restored, the backup produced using this keyword value might cause the application to fail if there are gaps in the GDSs for a GDG referenced by the application. For example, if the only relative generations referenced by the application are 0 and -2, backing up just these two generations causes the -2 generation to be -1 after recovery (assuming that the original -1 generation is not recovered from some other source.
- ALLCATALOGED–All generations found in the catalog are added to the database, if they are not already in the database. Be careful in specifying this option; users frequently overstate the LIMIT of the base catalog entry. New generations are normally excluded from the database unless SDS_GDG_NEWGENS_CATEGORY=INCLUDE or SDS_GDG_NEWGENS_CATEGORY=ALLOCATE.
- FORWARD–Equal numbers of oldest generations referenced and new generations are excluded, forcing a choice to be made for which new generations to include. Specifying FORWARD also forces **ABARS category for missing GDGs=ALLOCATE** to include, but only for certain generations.

  Note: Forward recovery applies to GDGs only. It does not infer that all data sets for a forward recovery will be captured.

SDS_GDG_ONLY_NEWGEN_CATEGORY=INCLUDE|EXCLUDE
Use this keyword to specify whether to include new GDSs in the Selection Data Set. This keyword is meaningful only if SDS_GDG_ADD_MISSING=FORWARD.

- INCLUDE–New GDSs are included in the Selection Data Set. This is the default value for this keyword.
- EXCLUDE–Do not include new GDSs in situations where no existing GDS for the GDG is referenced.

SDS_GDG_MISSING_CATEGORY=INCLUDE|ALLOCATE
Use this keyword to determine the ABARS SDS category for data sets added by the SDS_GDG_ADD_MISSING keyword.

- INCLUDE–Assign the added data sets to the INCLUDE category. This is the default value for this keyword.
- ALLOCATE–Assign the added data sets to the ALLOCATE category. After a restore, if the application is to be rerun exactly as it was run originally, you might save backup time by specifying ALLOCATE for the unreferenced GDGs.

SDS_GDG_NEWGENS_CATEGORY=EXCLUDE|INCLUDE|ALLOCATE
Use this keyword to determine the ABARS SDS category for GDSs found to be new. You can use this keyword to force the category for new GDSs to INCLUDE or ALLOCATE.

- EXCLUDE–Do not include the GDSs found to be new. This is the default value for this keyword.
- INCLUDE–Include the GDSs found to be new.
- ALLOCATE–Assign the new GDSs to the ALLOCATE category.

SDS_GDG_NAMES=BOTH|RELATIVE|ABSOLUTE
Use this keyword to specify how you want generation data set names in the Selection Data Set derived. Names are derived from database entries obtained from a JCL capture, RSP SMF records, or both (if SDS_EVAL_JCL_GDG=Y or SDS_EVAL_SMF_GDG=Y, or both).

If database entries exist from RSP JCL capture and SMF, and you specify either ABSOLUTE or RELATIVE for this keyword, the unacceptable names are excluded.

Absolute GDS names generated from relative database entries are refreshed from the catalog each time BKMAPLEN rebuilds the Selection Data Set or verification is run from ISPF.

For absolute names to be resolved correctly to relative names (or relative names to absolute names), the Selection Data Set must be built completely before any GDG activity alters the status of the GDG at the time the current cycle completed.

- BOTH–Data set names in the Selection Data Set can be absolute (.G0000V00s, for example) or relative (0, -1, and so forth). This is the default value for this keyword.
- RELATIVE–Data set names in the Selection Data Set are to be relative.
- ABSOLUTE–Data set names in the Selection Data Set are to be absolute.

EVALUATE_ALL_DATASETS_AS_CRITICAL=YES|NO

Use this keyword to specify whether the application should be defined for forward recovery or rerun recovery. Specify on of these options:

- YES–Forward recovery. All input and output data sets are classified as critical (with the exception of the data sets you exclude with filters) and the names are written to the Selection Data Set as INCLUDE or ACCOMPANY.
- NO–(Default) Rerun recovery. Only data sets that are used as input to the application (that is, data sets that existed before running the application) are considered critical (required for forward recovery).

DETAILED_COMMENTS_IN_SEL_DATASET=YES|NO

Use this keyword to specify whether you want additional comments generated in the Selection Data Set that indicate why a data set was classified in particular category:

- YES–Generate the additional comments in the Selection Data Set.
- NO–Do not generate the additional comments in the Selection Data Set.

SDS_DELETED_DATASETS=EXCLUDE|INCLUDE

Use this keyword to specify what you want done with a data set when its first reference is DELETE. Deleted data sets are in the URD display, regardless of the value you specify for this keyword. If the data set is uncataloged when the Selection Data Set is built, it is excluded, regardless of which value you specify for this keyword.

- EXCLUDE–Do not include the data set. This is the default value for this keyword.
- INCLUDE–Include the data set. Specifying this value is a good idea if deleted data sets resurface during the application, such as those that might result after a reorganization or rebuilding of the data set. Automated Critical Data Identification has no way of knowing whether the data set contents after reinstatement are the same as its contents at the beginning of the application.

Note: If the first reference of the data set is DISP=(MOD,DELETE) and the file exists and is opened for input, the first reference is INPUT, not DELETE, and the data set is put on the INCLUDE list. A data set with DISP=(MOD,DELETE) or DISP=(OLD,DELETE), where the file does not get opened, is put on the DELETE list.

SDS_RENAMED_OLD_DATASETS=EXCLUDE|INCLUDE

Use this keyword to specify what you want done with a data set when its first reference is RENAME for the existing (OLD) data set. Renamed data sets are in the URD display, regardless of the value you specify for this keyword. If the data set is uncataloged when the Selection Data Set is built, it is excluded, regardless of which value you specify for this keyword.

- EXCLUDE–Do not include the data set. This is the default value for this keyword.
- INCLUDE–Include the data set. Specifying this value is a good idea if renamed data sets resurface during the application, such as those that might result after a reorganization or rebuilding of the data set. Automated Critical Data Identification has no way of knowing whether the data set contents after reinstatement are the same as its contents at the beginning of the application.

SDS_RENAMED_NEW_DATASETS=EXCLUDE|INCLUDE

Use this keyword to specify what you want done with a data set when its first reference is RENAME for the new data set. Renamed data sets are in the URD display, regardless of the value you specify for this keyword. If the data set is uncataloged when the Selection Data Set is built, it is excluded, regardless of which value you specify for this keyword.

- EXCLUDE–Do not include the data set. This is the default value for this keyword.
- INCLUDE–Include the data set. Specifying this value is a good idea if renamed data sets resurface during the application, such as those that might result after a reorganization or rebuilding of the data set. Automated Critical Data Identification has no way of knowing whether the data set contents after reinstatement are the same as its contents at the beginning of the application.

SDS_USE_JCL_DATA=YES|NO
Use this keyword to specify whether you want the data collected by RSP JCL capture used in building the Selection Data Set.

- YES–Use the captured JCL data when building the Selection Data Set. This is the default value for this keyword.
- NO–Do not use the captured JCL data when building the Selection Data Set.

SDS_USE_SMF_DATA=YES|NO
Use this keyword to specify whether you want the data collected by the RSP SMF capture or batch SMF scan features used in building the Selection Data Set.

- YES–Use the captured or scanned SMF data when building the Selection Data Set. This is the default value for this keyword.
- NO–Do not use the captured or scanned SMF data when building the Selection Data Set.

SDS_STOP_SHARE=YES|NO
Use this keyword to specify whether you want the Selection Data Set build programs to stop sharing the Automated Critical Data Identification database during execution. The value you specify for this keyword is ignored if you are using RLS for the Automated Critical Data Identification database.

- YES–The Selection Data Set build programs stop sharing the database during execution.
- NO–The Selection Data Set build programs continue sharing the database during execution.

SDS_FLTR_OVERRIDES_CAT=NO|YES
Use this keyword to specify whether uncataloged entries are included in the Selection Data Set when an INCL_DSN filter has been specified.

- NO–The uncataloged entries are excluded from the Selection Data Set. This is the default value for this keyword.
- YES–Include the uncataloged entries.

SDS_CHG_ALLOC_TO_INCL=NO|YES
Use this keyword to specify whether you want data sets evaluated as ALLOCATE to be reassigned to the INCLUDE category.

- NO–Do not change ALLOCATE data sets to INCLUDE. This is the default value for this keyword.
- YES–Reassign ALLOCATE data sets to the INCLUDE category.

FILTERS_OVERRIDE_ACCOMPANY=YES|NO
Use this optional keyword to specify how you want the disposition of the critical tape data sets determined.

- YES–Disposition of the critical data sets is determined by a filter match, regardless of how the accompany switch is set. However, if a critical data set does not match any filters, its disposition is determined by the value assigned to the accompany switch.
- NO–Disposition of the critical data sets is determined by the value assigned to the accompany switch. This is the default value for this keyword.

:ACDI_RSP section keywords

Use the information in this section to specify appropriate values for the keywords in the :ACDI_RSP section of member BKMINI in the product parameter library. Be aware that some of these keywords can be specified for a specific BKMSTSKR in a multi-BKMSTSKR system.

Specifying keyword values for individual BKMSTSKR tasks

For several of the keywords in the :ACDI_RSP section of BKMINI, you can set values for a specified BKMSTSKR in a multi-BKMSTSKR system. The SMF system IDs are the values of either the SYSID(...) or the TEST(...) parameters passed to the started tasks in the JCL. If you specify an asterisk (*) as the SMF system ID for one of these keywords, Automated Critical Data Identification considers it to be a default for

any BKMSTSKR that does not have a keyword value associated specifically with it. Keywords that do not have SMF system IDs specified for them are ignored.

Consider the following example:

```
LOG = A.B.A (sysa) A.B.B (sysb) A.B.C (sysc) +
      A.B.X (xxx)  A.B.Q (*)
```

If a BKMSTSKR starts on an MVS system whose SMF ID is SYSA, DSN=A.B.A appears for it in the message log file. Note that data set A.B.A shows the SYSA SMF system ID in parentheses.

If a BKMSTSKR with an EXEC PARM set to PARM='TEST(XXX)' starts on any MVS system, DSN=A.B.X appears for it in the message log file. Data set A.B.X shows the XXX SMF system ID in parentheses.

If a BKMSTSKR with an EXEC PARM set to PARM='SYSID(SYSC)' starts on any MVS system, DSN=A.B.C appears for it in the message log file. Data set A.B.C shows the SYSC SMF system ID in parentheses.

If a BKMSTSKR starts on an MVS system whose SMF system ID is TEST, DSN=A.B.Q appears in the message log file. Data set A.B.Q shows the * SMF system ID in parentheses, indicating a default value.

RSP_LOG=*prefix.value*.Vvrr.RSP.MSGLOG(*)*prefix.value*.Vvrr.RSP.MSGLOG(sysid)
Use this keyword to specify the name of the started task RSP log files. It is recommended that you specify one log file that is shared across all your BKMSTSKR started tasks, unless Technical Support has advised you to do otherwise.

The following example shows the keyword specification of an RSP log file to be shared by all the BKMSTSKR in a multi-BKMSTSKR installation:

```
LOG = prefix.value.Vvrr.RSP.MSGLOG (*)
```

The following example shows the keyword specification when you want to use a different RSP log file for each individual BKMSTSKR in a multi-BKMSTSKR installation:

```
LOG = prefix.value.Vvrr.RSP.PROD.MSGLOG (PROD)  +
      prefix.value.Vvrr.RSP.DVLP.MSGLOG (DVLP)  +
      prefix.value.Vvrr.RSP.TEST.MSGLOG (TEST)
```

RSP_AUTOSTART=(YES (*)|(sysid))|(NO (*)|(sysid))
Use this keyword to specify how you want the individual processes within the BKMSTSKR structure started. You must specify either *, indicating that you want the keyword value applied to all systems running RSP, or a SMF system ID to which you want the keyword value applied.

- YES–Start all individual processes within the BKMSTSKR structure.
- NO–Do not start individual processes within the BKMSTSKR structure.

RSP_DSPC_BUFFERS=nnnnn(*|sysid)
Use this keyword to specify the maximum number of buffers RSP can use. The number you specify multiplied by the 288-byte buffer length cannot exceed the maximum data space size, 32 MB. Use the asterisk (*) to specify one value that applies to this keyword across all BKMSTSKR tasks in a multi-BKMSTSKR environment. Alternatively, you can specify a unique value for each instance of BKMSTSKR by using the SMF system IDs in the value specification (*sysid*). The default value for this keyword is 10000 (*)

RSP_USE_U84_EXIT=YES(*|sysid)|NO(*|sysid)
Use this keyword to specify whether you want SMF data from the U84 exit collected. Use the asterisk (*) to specify one value that applies to this keyword across all BKMSTSKRs in a multi-BKMSTSKR environment. Alternatively, you can specify a specific value for each instance of BKMSTSKR by using the SMF system IDs in the value specification (*sysid*).

Note: Use of this exit is required if you are using the application cycle control option.

- YES–Collect JCL data from the U84 exit.
- NO–Do not collect JCL data from the U84 exit.

RSP_USE_U83_EXIT=YES(*|sysid)|NO(*|sysid)
Use this keyword to specify whether you want SMF data from the U83 exit collected. Use the asterisk (*) to specify one value that applies to this keyword across all BKMSTSKRs in a multi-BKMSTSKR environment. Alternatively, you can specify a specific value for each instance of BKMSTSKR by using the SMF system IDs in the value specification (*sysid*).

- YES–Collect JCL data from the U83 exit.
- NO–Do not collect JCL data from the U83 exit.

RSP_USE_UJI_EXIT=YES(*|sysid)|NO(*|sysid)
Use this keyword to specify whether you want JCL data from the UJI exit collected. Use the asterisk (*) to specify one value that applies to this keyword across all instances of BKMSTSKR in a multi-BKMSTSKR environment. Alternatively, you can specify a specific value for each instance of BKMSTSKR by using the SMF system IDs in the value specification (*sysid*).

- YES–Collect JCL data from the UJI exit.
- NO–Do not collect JCL data from the UJI exit.

RSP_START_HDU_TASK=NO|YES
Set this keyword to YES (the default value) if you want data set usage information collected and stored in the Historical Dataset Usage database every time RSP is started. Specifying YES for this keyword ensures that the data in the Historical Dataset Usage database is current. If this keyword is set to NO when RSP is started, no new data is stored in the Historical Dataset Usage database until this keyword is set to YES when RSP is started up.

RSP_SKIP_ALL_STARTEDTASKS=NO|YES
Use this keyword to control processing in the UJI and U83 exit modules. This keyword applies to RSP only; it does not apply to Batch SMFSCAN. RSP must be recycled for this switch to take effect.

- NO–All SMF records from any started task are written to the Automated Critical Data Identification database. This is the recommended setting for this keyword if you intend to collect CICS or other started task data. NO is the default value for this keyword.
- YES–Drop all SMF records from any started task and do not write them to the Automated Critical Data Identification database.

RSP_SKIP_ALL_TSOJOBS=NO|YES
Use this keyword to control processing in the UJI and U83 exit modules. This keyword applies to RSP only; it does not apply to Batch SMFSCAN. RSP must be recycled for this switch to take effect.

- NO–All SMF records from TSO jobs are written to the Automated Critical Data Identification database. NO is the default value for this keyword.
- YES–Drop all SMF records from TSO jobs and do not write them to the Automated Critical Data Identification database.

RSP_JOBFLAT=*prefix.value*.Vvrr.RSP.JOBFLAT
Use this keyword to specify the name of the extract file to be used by RSP U83 exits. This file is created by JCL member BKMRSPSU during Automated Critical Data Identification configuration and is loaded by BKMBLDJB or BKMJRFSH.

RSP_MSGQUEUE=*prefix.value*.Vvrr.RSP.RSPMSGQ
Use this keyword to specify the name of the file that holds RSP messages and statistics. This file is created by JCL member BKMRSPSU during Automated Critical Data Identification configuration.

RSP_OVERFLOW_DSN=*prefix.value*.Vvrr.RSP.OVERFLOW (sysid)
Use this keyword to specify the name of the overflow data set. You must have one overflow data set for each BKMSTSKR running in your environment. Each specification you make for this keyword value must include `(sysid)`, where *sysid* is the SMF system ID where the BKMSTSKR is running. The overflow data sets should not be pre-allocated.

The following is an example value specification for multiple overflow data sets:

```
OVERFLOW_DSN=AMP.PACRIM.DVLP.OVERFLOW (DVLP) +
AMP.PACRIM.TST.OVERFLOW (TST)
AMP.PACRIM.MNT.OVERFLOW (MNT)
```

RSP_OVERFLOW_PARMS=BLKSIZE(prefix.value) UNIT(xxxx SPACE(nn nn) CYLINDERS|TRACKS RACCHECK(YES|NO) VOLUME(volser)

Use this keyword to specify the allocation values for the overflow data sets on each instance of the BKMSTSKD started task. By default, the following allocations are used:

```
BLKSIZE(0) UNIT(SYSALLDA) SPACE(10 2) CYLINDERS +  RACCHECK(YES)
```

Adhering to the TSO ALLOCATE command syntax, you can add other parameters, but the total length you specify cannot exceed 136. The overflow data set is fixed block and has a 300-byte record size.

- If you want to allocate the overflow data set on a specific volume, include the VOLUME(*volser*) parameter, where *volser* is the volume on which you want the overflow data set to reside.
- Specify RACCHECK(YES) if you want RACF checking done.
- You cannot change the LRECL, DDNAME, DSORG, RECFM, or DISP attributes for the overflow data sets.
- The overflow data set should not be pre-allocated.

RSP_DELETE_DETAIL_AT_APPLEND=NO|YES

Use this optional keyword to delete the DSD records after the Selection Data Set has been built successfully. The DSD records are used during verification as well. When the records have been deleted, the last successful cycle was removed an no VERIFYs can be done.

- NO–Do not delete the DSD records after the Selection Data Set has been built. NO is the default value for this keyword.
- YES–Delete the DSD records after the Selection Data Set has been built.

:ACDIOPC_PARMS section keywords

Use the information in this topic to specify appropriate values for the keywords in the :ACDIOPC_PARMS section of member BKMINI in the product parameter library.

EQQMLIB='NULLFILE'

Use this keyword to specify the name of the OPC/ESA message library that contains the OPC/ESA messages issued by the OPC/ESA batch report jobs. You can find the name of this library in the OPC started task that usually resides in the SYS1.PROCLIB.

EQQPARM='NULLFILE'

Use this keyword to specify the name of the OPC/ESA parameter library used by OPC/ESA batch jobs. The OPC/ESA options are controlled by the BATCHOPT member of this library.

For further information about the BATCHOPT parameters, refer to the *OPC/ESA Installation and Customization Guide* (SH19-6718).

EQQAD='NULLFILE'

Use this keyword to specify the name of the OPC VSAM application description data set.

DATE_FORM=MMDDYY|YYMMDD|DDMMYY

Use this keyword to specify the format of the date range in the OPT/TWS INTERNAL DEPENDENCIES Report.

APPLI ID:applid APPL TEXT:appl text VALID:01/14/03-12/31/71 set token value to MMDDYY.

APPLI ID:applid APPL TEXT:appl text VALID:03/01/14-71/12/31 set token value to YYMMDD. This is the default value for this keyword.

APPLI ID:applid APPL TEXT:appl text VALID:14/01/03-31/12/71 set token value to DDMMYY.

BATCHOPT=MemberName|BATCHOPT

Use this keyword to specify the name to be used for the EQQPARM member. The default value for this keyword is BATCHOPT.

:CATSCRUB_OPTIONS section keywords

Use the information in this topic to set appropriate values for the keywords in the :CATSCRUB_OPTIONS section of member BKMINI in the product parameter library. These keywords determine how the CATSCRUB command behaves in batch mode. If you do not intend to use the CATSCRUB command, leave the keywords in this section set to their default values.

NONVSAM_MULTI_VOL_ERR_DISP=KEEP|DELETE
Use this keyword to specify whether you want the catalog record for a non-VSAM data set to be kept in the catalog, regardless of its allocation status.

- KEEP–Retain the non-VSAM data set's catalog record, regardless of its allocation status. This is the default value for this keyword.
- DELETE–Delete the non-VSAM data set's catalog record if the data set is found to be in an incomplete allocation status.

Note: When CATSCRUB deletes a catalog record, it performs the equivalent of a DELETE NOSCRATCH; it does not physically delete any object on the disk volumes.

NONVSAM_MULTI_VOL_ERR_RC=n
Use this keyword to specify the return code you want assigned on the SYSOUT message whenever a non-VSAM data set is found to be in an incomplete allocation status. The default value for this keyword is 4. You can specify any value in the range 0-9999, inclusive.

NONVSAM_MULTI_VOL_ERR_PROC=CONTINUE|END
Use this keyword to specify whether you want CATSCRUB processing to continue in situations where it encounters a non-VSAM data set that is in an incomplete allocation status.

- CONTINUE–CATSCRUB processing continues in situations where a non-VSAM data set in incomplete allocation status is encountered. This is the default value for this keyword.
- END–CATSCRUB processing ends in situations where a non-VSAM data set is in incomplete allocation status.

NONVSAM_MIGRATED_DISP=KEEP|DELETE
Use this keyword to specify whether you want to keep the catalog record for any non-VSAM data set in migrate status.

- KEEP–Keep the non-VSAM data set's catalog record, even when it is in migrate status. This is the default value for this keyword.
- DELETE–Delete the non-VSAM data set's catalog record when it is in migrate status.

Note: When CATSCRUB deletes a catalog record, it performs the equivalent of a DELETE NOSCRATCH; it does not physically delete any object on the disk volumes.

NONVSAM_MIGRATED_RC=n
Use this keyword to specify the return code you want assigned on the SYSOUT message whenever a non-VSAM data set is found to be in migrate status. The default value for this keyword is 4. You can specify any value in the range 0-9999, inclusive.

NONVSAM_MIGRATED_PROC=CONTINUE|END
Use this keyword to specify whether you want CATSCRUB processing to continue in situations where it encounters a non-VSAM data set that is in migrate status.

- CONTINUE–CATSCRUB processing continues in situations where a non-VSAM data set in migrate status is encountered. This is the default value for this keyword.
- END–CATSCRUB processing ends in situations where a non-VSAM data set is in migrate status.

NONVSAM_NOT_FOUND_DISP=KEEP|DELETE
Use this keyword to specify whether you want to keep the catalog record for any non-VSAM data set that no longer exists.

- KEEP–Keep the non-VSAM data set's catalog record, even when the data set no longer exists. This is the default value for this keyword.

- DELETE–Delete the non-VSAM data set's catalog record when the data set no longer exists.

Note: When CATSCRUB deletes a catalog record, it performs the equivalent of a DELETE NOSCRATCH; it does not physically delete any object on the disk volumes.

NONVSAM_NOT_FOUND_RC=n
Use this keyword to specify the return code you want assigned on the SYSOUT message whenever a non-VSAM data set is found to exist no longer. The default value for this keyword is 4. You can specify any value in the range 0-9999, inclusive.

NONVSAM_NOT_FOUND_PROC=CONTINUE|END
Use this keyword to specify whether you want CATSCRUB processing to continue in situations where it encounters a non-VSAM data set that no longer exists.

- CONTINUE–CATSCRUB processing continues in situations where a non-VSAM data set is found to no longer exist. This is the default value for this keyword.
- END–CATSCRUB processing ends in situations where a non-VSAM data set is found to no longer exist.

NONVSAM_TAPE_DISP=KEEP|DELETE
Use this keyword to specify whether you want to keep the catalog record for any non-VSAM data set that exists on tape.

- KEEP–Keep the non-VSAM data set's catalog record, even when the data set exists on tape. This is the default value for this keyword.
- DELETE–Delete the non-VSAM data set's catalog record when the data set exists on tape.

Note: When CATSCRUB deletes a catalog record, it performs the equivalent of a DELETE NOSCRATCH; it does not physically delete any object on the disk volumes.

NONVSAM_TAPE_RC=n
Use this keyword to specify the return code you want assigned on the SYSOUT message whenever a non-VSAM data set is found to reside physically on tape. The default value for this keyword is 4. You can specify any value in the range 0-9999, inclusive.

NONVSAM_TAPE_PROC=CONTINUE|END
Use this keyword to specify whether you want CATSCRUB processing to continue in situations where it encounters a non-VSAM data set that physically resides on tape.

- CONTINUE–CATSCRUB processing continues in situations where a non-VSAM data set is found to reside physically on tape. This is the default value for this keyword.
- END–CATSCRUB processing ends in situations where a non-VSAM data set is found to reside physically on tape.

NONVSAM_ALIAS_NO_REALNAME_DISP=KEEP|DELETE-ALIAS
Use this keyword to specify whether you want to keep the catalog record for any non-VSAM alias in situations where the real data set does not exist on the specified volumes for the command.

- KEEP–Keep the catalog record for the non-VSAM alias, even when the real data set does not exist on the specified volumes. This is the default value for this keyword.
- DELETE-ALIAS–Delete the catalog record for the non-VSAM alias when the real data set does not exist on the specified volumes.

Note: When CATSCRUB deletes a catalog record, it performs the equivalent of a DELETE NOSCRATCH; it does not physically delete any object on the disk volumes.

NONVSAM_ALIAS_NO_REALNAME_RC=n
Use this keyword to specify the return code you want assigned on the SYSOUT message whenever the real data set for a non-VSAM alias does not physically exist. The default value for this keyword is 4. You can specify any value in the range 0-9999, inclusive.

NONVSAM_ALIAS_NO_REALNAME_PROC=CONTINUE|END
Use this keyword to specify whether you want CATSCRUB processing to continue in situations where the real data set does not physically exist for a non-VSAM alias.

- CONTINUE–CATSCRUB processing continues in situations where the data set for a non-VSAM alias does not physically exist. This is the default value for this keyword.
- END–CATSCRUB processing ends in situations where a data set for a non-VSAM alias does not physically exist.

VSAMSPHERE_MULTI_VOL_ERR_DISP=KEEP|DELETE
Use this keyword to specify whether you want to keep the cluster sphere record for a VSAM cluster in situations where the cluster is in migrate status.

- KEEP–Keep the cluster sphere record in situations where the VSAM cluster is in migrate status. This is the default value for this keyword.
- DELETE–Delete the cluster sphere record in situations where the VSAM cluster is in migrate status.

Note: When CATSCRUB deletes a catalog record, it performs the equivalent of a DELETE NOSCRATCH; it does not physically delete any object on the disk volumes.

VSAMSPHERE_MULTI_VOL_ERR_RC=n
Use this keyword to specify the return code you want assigned on the SYSOUT message whenever the cluster sphere record for a VSAM cluster is in the migrate status. The default value for this keyword is 4. You can specify any value in the range 0-9999, inclusive.

VSAMSPHERE_MULTI_VOL_ERR_PROC=CONTINUE|END
Use this keyword to specify whether you want CATSCRUB processing to continue when it encounters a VSAM data set in an incomplete allocation status.

- CONTINUE–CATSCRUB processing continues when it encounters a VSAM data set in an incomplete allocation status. This is the default value for this keyword.
- END–CATSCRUB processing ends on the first occurrence of a multivolume VSAM data set in an incomplete allocation status.

VSAMSPHERE_MIGRATED_DISP=KEEP|DELETE
Use this keyword to specify whether you want to keep the cluster sphere record for a VSAM cluster in situations where the cluster is in migrate status.

- KEEP–Keep the cluster sphere record in situations where the VSAM cluster is in migrate status. This is the default value for this keyword.
- DELETE–Delete the cluster sphere record in situations where the VSAM cluster is in migrate status.

Note: When CATSCRUB deletes a catalog record, it performs the equivalent of a DELETE NOSCRATCH; it does not physically delete any object on the disk volumes.

VSAMSPHERE_MIGRATED_RC=n
Use this keyword to specify the return code you want assigned on the SYSOUT message whenever the cluster sphere record for a VSAM cluster is found to be in the migrate status. The default value for this keyword is 4. You can specify any value in the range 0-9999, inclusive.

VSAMSPHERE_MIGRATED_PROC=CONTINUE|END
Use this keyword to specify whether you want CATSCRUB processing to continue when it encounters a VSAM data set in migrate status.

- CONTINUE–CATSCRUB processing continues when it encounters a VSAM data set in migrate status. This is the default value for this keyword.
- END–CATSCRUB processing ends when it encounters a VSAM data set in migrate status.

VSAMSPHERE_NOT_FOUND_DISP=KEEP|DELETE
Use this keyword to specify whether you want to keep the catalog record for a VSAM data set in situations where the VSAM data set no longer exists.

- KEEP–Keep the catalog record in situations where the VSAM data set no longer exists. This is the default value for this keyword.
- DELETE–Delete the catalog record in situations where the VSAM data set no longer exists.

VSAMSPHERE_NOT_FOUND_RC=n
Use this keyword to specify the return code you want assigned on the SYSOUT message whenever a VSAM data set is found to no longer exist. The default value for this keyword is 4. You can specify any value in the range 0-9999, inclusive.

VSAMSPHERE_NOT FOUND_PROC=CONTINUE|END
Use this keyword to specify whether you want CATSCRUB processing to continue in situations where it cannot find the VSAM data set.

- CONTINUE–CATSCRUB processing continues when it encounters a situation where the VSAM data set cannot be found. This is the default value for this keyword.
- END–CATSCRUB processing ends when it encounters a situation where the VSAM data set cannot be found.

GDS_NOT_FOUND_DISP=KEEP|DELETE
Use this keyword to specify whether you want to keep the generation data set within the GDG sphere, even when the data set does not physically exist. Consider your value specification carefully for this keyword, as it can cause a change in the relative number of generation data sets within a GDG, which can in turn produce an unintended result during subsequent processing with relative generation specifications in the job streams.

- KEEP–Retain the generation data set within the GDG sphere, even when the data set does not physically exist. Keeping the GDS entry in these situations can result in an error when the GDG is processed subsequently, although it does result in a processing error if this specific generation in error is never actually accessed. An example of this situation would be the GDS being retained in the GDG sphere record to maintain proper relativity of generations and is not intended for processing.
- DELETE–Delete the generation data set within the GDG sphere in situations where the data set does not physically exist on the volumes to be matched by CATSCRUB.

GDS_NOT_FOUND_RC=n
Use this keyword to specify the return code you want assigned on the SYSOUT message whenever a generation data set is found to no longer exist. The default value for this keyword is 4. You can specify any value in the range 0-9999, inclusive.

GDS_NOT FOUND_PROC=CONTINUE|END
Use this keyword to specify whether you want CATSCRUB processing to continue in situations where the generation data set does not physically exist.

- CONTINUE–CATSCRUB processing continues when it encounters a situation where the generation data set does not physically exist. This is the default value for this keyword.
- END–CATSCRUB processing ends when it encounters a situation where the generation data set does not physically exist.

GDS_MULTI_VOL_ERR_DISP=KEEP|DELETE
Use this keyword to specify whether you want to keep the generation data set within the GDG sphere, even when the validity of its allocation status is in question. Consider your value specification carefully for this keyword, as it can cause a change in the relative number of generation data sets within a GDG, which can in turn produce an unintended result during subsequent processing with relative generation specifications in the job streams.

- KEEP–Retain the generation data set within the GDG sphere, even when the data set's allocation status is in question. Keeping the GDS entry in these situations can result in an error when the GDG is processed subsequently, although it does result in a processing error if this specific generation in error is never actually accessed. An example of this situation would be the GDS being retained in the GDG sphere record to maintain proper relativity of generations and is not intended for processing.
- DELETE–Delete the generation data set within the GDG sphere in situations where the data set's allocation status is in question.

GDS_MULTI_VOL_ERR_RC=n
Use this keyword to specify the return code you want assigned on the SYSOUT message whenever a generation data set is found to be in an incomplete allocation status. The default value for this keyword is 4. You can specify any value in the range 0-9999, inclusive.

GDS_MULTI_VOL_ERR_PROC=CONTINUE|END
Use this keyword to specify whether you want CATSCRUB processing to continue in situations where the generation data set is in an incomplete allocation status.

- CONTINUE–CATSCRUB processing continues when it encounters a situation where the generation data set is in an incomplete allocation status. This is the default value for this keyword.
- END–CATSCRUB processing ends when it encounters a situation where the generation data set is in an incomplete allocation status.

GDS_MIGRATED_DISP=KEEP|DELETE
Use this keyword to specify whether the generation data set is kept within the GDG sphere if it is found to be in migrate status.

Note: If you do not explicitly specify values for *all* of the GDS_MIGRATED-DASD_DISP, GDS_MIGRATED-DASD_RC, GDS_MIGRATED-DASD_PROC, GDS_MIGRATED-TAPE_DISP, GDS_MIGRATED-TAPE_RC, and GDS_MIGRATED-TAPE_PROC device-specific keywords, the values you specify for the GDS_MIGRATED_DISP, GDS_MIGRATED_RC, and GDS_MIGRATED_PROC keywords serve as the default values for the device-specific keywords.

- KEEP–Retain the generation data set within the GDG sphere, even when the data set is in migrate status. Keeping the GDS entry in these situations can result in an error when the GDG is processed subsequently, although it does result in a processing error if this specific generation in error is never actually accessed. An example of this situation would be the GDS being retained in the GDG sphere record to maintain proper relativity of generations and is not intended for processing.
- DELETE–Delete the generation data set within the GDG sphere in situations where the data set is in migrate status.

Note: When CATSCRUB deletes a catalog record, it performs the equivalent of a DELETE NOSCRATCH; it does not physically delete any object on the disk volumes.

GDS_MIGRATED_RC=n
Use this keyword to specify the return code you want assigned to the SYSOUT message whenever a generation data set is found to be in the migrate status. You can specify any value in the range 0-9999, inclusive.

Note: If you do not explicitly specify values for *all* of the GDS_MIGRATED-DASD_DISP, GDS_MIGRATED-DASD_RC, GDS_MIGRATED-DASD_PROC, GDS_MIGRATED-TAPE_DISP, GDS_MIGRATED-TAPE_RC, and GDS_MIGRATED-TAPE_PROC device-specific keywords, the values you specify for the GDS_MIGRATED_DISP, GDS_MIGRATED_RC, and GDS_MIGRATED_PROC keywords serve as the default values for the device-specific keywords.

Note: When CATSCRUB deletes a catalog record, it performs the equivalent of a DELETE NOSCRATCH; it does not physically delete any object on the disk volumes.

GDS_MIGRATED_PROC=CONTINUE|END
Use this keyword to specify whether you want CATSCRUB to continue processing when it encounters a generation data set in migrate status.

Note: If you do not explicitly specify values for *all* of the GDS_MIGRATED-DASD_DISP, GDS_MIGRATED-DASD_RC, GDS_MIGRATED-DASD_PROC, GDS_MIGRATED-TAPE_DISP, GDS_MIGRATED-TAPE_RC, and GDS_MIGRATED-TAPE_PROC device-specific keywords, the values you specify for the GDS_MIGRATED_DISP, GDS_MIGRATED_RC, and GDS_MIGRATED_PROC keywords serve as the default values for the device-specific keywords.

- CONTINUE–Continue processing when CATSCRUB encounters a generation data set in migrate status.
- END–End processing when CATSCRUB encounters a generation data set in migrate status.

GDS_TAPE_DISP=KEEP|DELETE
Use this keyword to specify whether you want to keep the generation data set within the GDG sphere, even when the data set resides physically on tape. Consider your value specification carefully for this keyword, as it can cause a change in the relative number of generation data sets within a GDG,

which can in turn produce an unintended result during subsequent processing with relative generation specifications in the job streams.

- KEEP–Retain the generation data set within the GDG sphere, even when the data set resides physically on tape. Keeping the GDS entry in these situations can result in an error when the GDG is processed subsequently and the tape cannot be located, although no processing error occurs if the generation in error is never actually accessed. An example of this situation would be the GDS being retained in the GDG sphere record to maintain proper relativity of generations and is not intended for processing.
- DELETE–Delete the generation data set within the GDG sphere in situations where the data set resides physically on tape.

Note: When CATSCRUB deletes a catalog record, it performs the equivalent of a DELETE NOSCRATCH; it does not physically delete any object on the disk volumes.

GDS_TAPE_RC=n
Use this keyword to specify the return code you want assigned on the SYSOUT message whenever a generation data set is found to reside physically on tape. The default value for this keyword is 4. You can specify any value in the range 0-9999, inclusive.

GDS_TAPE_PROC=CONTINUE|END
Use this keyword to specify whether you want CATSCRUB processing to continue in situations where the generation data set resides physically on tape.

- CONTINUE–CATSCRUB processing continues when it encounters a situation where the generation data set resides physically on tape. This is the default value for this keyword.
- END–CATSCRUB processing ends when it encounters a situation where the generation data set resides physically on tape.

GDSBASE_NO_ACTIVE_GENS_DISP=KEEP|DELETE
Use this keyword to specify whether you want to keep the GDG sphere record in situations where it is empty of generation data sets.

- KEEP–Retain the GDG sphere record, even when the record is empty of generation data sets.
- DELETE–Delete the GDG sphere record in situations where the record is empty of generation data sets.

Note: When CATSCRUB deletes a catalog record, it performs the equivalent of a DELETE NOSCRATCH; it does not physically delete any object on the disk volumes.

GDSBASE_NO_ACTIVE_GENS_RC=n
Use this keyword to specify the return code you want assigned on the SYSOUT message issued whenever a GDG sphere record is identified as having no active generations. The default value for this keyword is 4. You can specify any value in the range 0-9999, inclusive.

GDSBASE_NO_ACTIVE_GENS_PROC=CONTINUE|END
Use this keyword to specify whether you want CATSCRUB processing to continue when it encounters a GDG sphere record that is empty of generations or when other CATSCRUB processing deletes all generations of the GDG.

- CONTINUE–CATSCRUB processing continues when it encounters a GDG sphere record that is empty of generations or when other CATSCRUB processing deletes all generations of the GDG. This is the default value for this keyword.
- END–CATSCRUB processing ends when it encounters a GDG sphere record that is empty of generations or when other CATSCRUB processing deletes all generations of the GDG.

OBJECT_ACCESS_METHOD_DISP=KEEP|DELETE
Use this keyword to specify whether you want to keep the OAM entry.

- KEEP–Retain the OAM entry. This is the default value for this keyword.
- DELETE–Delete the OAM entry.

Note: When CATSCRUB deletes a catalog record, it performs the equivalent of a DELETE NOSCRATCH; it does not physically delete any object on the disk volumes.

OBJECT_ACCESS_METHOD_RC=n
Use this keyword to specify the return code you want assigned on the SYSOUT message issued for each OAM entry processed. The default value for this keyword is 4. You can specify any value in the range 0-9999, inclusive.

OBJECT_ACCESS_METHOD_PROC=CONTINUE|END
Use this keyword to specify whether you want CATSCRUB processing to continue when it processes an OAM entry.

- CONTINUE–CATSCRUB processing continues when it processes an OAM entry. This is the default value for this keyword.
- END–CATSCRUB processing ends when it processes an OAM entry.

FATAL_CATALOG_ERR_PROC=CONTINUE|END
Use this keyword to specify whether you want CATSCRUB processing to continue when it encounters a fatal catalog error.

- CONTINUE–CATSCRUB processing continues when it encounters a fatal catalog error.
- END–CATSCRUB processing ends when it encounters a fatal catalog error. This is the default value for this keyword.

:SI040_VALUES section keywords

Use the information in this topic to specify appropriate values for the :SI040_VALUES section of the BKMINI member in the product parameter library, which is contained in module BKM01VSI. It is recommended that you change the values for these keywords only at the instruction of Technical Support.

The values for *VSAM-CLSUTER-NAME=RLS can be changed if you are using RLS (VSAM record level sharing) at your installation. If you have defined the HSM control data sets to use RLS, you might want to modify the keywords in the :SI040_VALUES section.

Using RLS can improve performance of Advanced Backup and Recovery for z/OS. Users can choose to access product VSAM data sets through RLS. It is highly recommended that you use RLS for all VSAM flat files.

BKM.INVDB.LOCKDSN=lockdatasetname
Use this keyword to identify the lock data set. You must modify the keyword itself as well as specifying a value for the keyword.

1. Replace BKM.INVDB.LOCKDSN with *clustername*.LOCKDSN, where *clustername* is the cluster name of the Advanced Backup and Recovery for z/OS Inventory Data Set (IDS).
2. Replace *lockdatasetname* with the name of a small data set to be used as a permanent lock switch. You can specify any valid data set name.

Note: Because the lock data set is allocated dynamically, you can use the LOCKDSN.ATTRIB keyword to specify the allocation parameters for it.

LOCKDSN.ATTRIB=[STORCLAS] [UNIT] [VOL] ...
Use this keyword to specify the allocation parameters you want used when the lock data set is allocated dynamically. You can specify any allocation parameters that adhere to TSO ALLOCATE syntax rules.

Consider the following example:

```
BKM.INVDB.LOCKDSN = BKM.LOCK.THIS.FILE
LOCKDSN.ATTRIB = STORCLAS(MYSMS)
```

In this example, the REORG utility will be used for the Advanced Backup and Recovery for z/OS Inventory Data Set (IDS) BKM.INVDB. The data set to be allocated dynamically and used as an in-progress lock switch is BKM.LOCK.THIS.FILE, allocated using storage class MYSMS. The REORG utility does not reorganize the IDS unless the lock data set has been defined.

***VSAM-CLUSTER-NAME.MEDIUM|*VSAM-CLUSTER-NAME.HIGH|*VSAM-CLUSTER-NAME.LOW=**
Change the values for this keyword at the direction of Technical Support only. The keywords and values described here are valid for HIGH, MEDIUM, and LOW.

- TTW–TIMES TO WAIT describes the number of times BKM01VSI waits before issuing a RC 94. Your specification must be an integer. The default values are as follows:
  - HIGH 25
  - MEDIUM 100
  - LOW 100
- IA–INTERVAL ACTIVE describes the time period BKM01VSI remains active before checking for other requesters. Specify the appropriate value to two decimal places, without including the decimal point. For example, to specify a time period of 25 seconds, your entry would be `IA(2500)`; for a time period of 2.5 seconds, you would specify `IA(250)`. The default values are as follows:
  - HIGH 30.00 seconds
  - MEDIUM 8.00 seconds
  - LOW 1.50 seconds
- ITW–INTERVAL TO WAIT describes the time period BKM01VSI waits after giving up control of the file. Specify the appropriate value to two decimal places, without including the decimal point. For example, to specify a time period of 2.5 seconds, your entry would be `ITW(250)`; for a wait interval of 25 seconds, your value specification would be `ITW(2500)`. The default values are as follows:
  - HIGH 2.00 seconds
  - MEDIUM 1.00 second
  - LOW 1.00 second
- ITP–INTERVAL TO PAUSE describes the time period BKM01VSI remains dormant after giving up control of the file before requesting it again. Specify the appropriate value to two decimal places, without including the decimal point. For example, to specify a time period of 2.5 seconds, your entry would be `ITP(250)`; for a wait interval of 25 seconds, your value specification would be `ITP(2500)`. The default values are as follows:
  - HIGH 0.00 seconds
  - MEDIUM 0.00 seconds
  - LOW 0.00 seconds

Use this keyword alone to override globally the BKM01VSI default values. Use data set names to override the global overrides for specific data sets.

Consider the following examples:

```
HIGH = TTW( 11) IA(2222) ITW(333) TIP(0400)
MEDIUM = TTW(555) ITW(777) ITP(0800)
LOW = TTW(999) IA( 111) ITP(0300)
A.B.C.MEDIUM = TTW( 0) IA(1000) ITW(111)
EH.AMPVOLRA.DAD.MEDIUM = TTW( 0) IA(1000) TIW(111) ITP(10)
EM.MCGHEE.DATAFILE.LOW = TTW(99) IA( 4111)
BP.RUFUS.DATABASE.HIGH = TTW(999) IA( 111)
```

***VSAM-CLUSTER-NAME.BUFFERS=**
Change the value specifications for this keyword at the direction of Technical Support only.

- DNSR – Specifies the number of data component buffers when registered for Non Shared Resource (NSR) buffering. (Registration value 0.) The default value is 3.
- INSR – specifies the number of index component buffers when registered for Non Shared Resource (NSR) buffering. (Registration value 0.) The default value is 12.
- DLSR – specifies the number of data component buffers when registered for Local Shared Resource (LSR) buffering. (Registration value 4.)
  - If the data component is 5 MB or less, the entire data component is buffered.

- If the data component is greater than 5 MB two control areas of buffers are allocated.

- ILSR – specifies the number of index component buffers when registered for Local Shared Resource (LSR) buffering. (Registration value 4.)

  Enough buffers for the entire index component will be allocated. Use this token alone to globally override BKM01VSI default values.

Use data set names to override the global overrides for specific data sets.

Consider the following examples:

```
BUFFERS = DNSR(100) INSR(200) DLSR(300) ILSR(400)
A.B.C.BUFFERS = DNSR(2222) INSR(3333)
EH.AMPVOLRA.DAD.BUFFERS = ILSR(15) DLSR(51)
BP.RUFUS.DATABASE.BUFFERS = DLSR(2222) ILSR(3333)
```

***VSAM-CLUSTER-NAME= RLS**

Configure this keyword if RLS is employed for the DFSMShsm MCDS or BCDS (or both), or BKMSTSKA is using a VSAM flat file. Specify RLS to request that BKM01VSI use Record Level Sharing for this data set. This requires that your installation's coupling facility is operable, and the SYSVSAM address space is active. The RLS option is mutually exclusive with the *VSAM-CLUSTER-NAME =NORNL option.

1. Delete the asterisk from the *VSAM-CLUSTER-NAME = RLS keyword to activate it.
2. Replace `VSAM-CLUSTER-NAME = RLS` with `YOUR.HSM.MCDS.NAME = RLS` or BCDS name.
3. If multiple DFSMShsm MCDS or BCDS need to be specified, insert additional lines.

```
HSM.MCDS = RLS
HSM.MCDS2 = RLS
HSM.BCDS = RLS
HSM.BCDS2 = RLS
```

The following example shows the specification of the HSM control data sets and BKMSTSKA VSAM flat file using RLS:

```
PROD.MCDS.#10F2 = RLS
PROD.MCDS.#20F2 = RLSPROD.CBTI.VSAMFF = RLS
```

***VSAM-CLUSTER-NAME.RLS_TIMEOUT=25**

Do not change the value of this keyword unless you are directed to do so by Technical Support.

Use this keyword to specify the time, in seconds, that VSAM waits while trying to obtain the RLS lock for a file. When this value is exceeded, BKM01VSI gives a return code of 8, with the response block containing 8/0/22.

Specify an integer value for this keyword.

Use this keyword alone to globally override BKM01VSI default values. Use data set names to override the global overrides of specific data sets.

In the following example, a value of 6 minutes is being specified for the keyword:

```
RLS_TIMEOUT=360
X.Y.Z.RLS_TIMEOUT = 25
```

***VSAM-CLUSTER-NAME=NORNL**

Do not implement this keyword without guidance from Technical Support. Improper use of this keyword can cause serious unrecoverable damage to the data set and can produce unpredictable results from Advanced Backup and Recovery for z/OS.

You cannot use this keyword if you have implemented the *VSAM-CLUSTER-NAME=RLS keyword.

Use this keyword to request the BKM01VSI should bypass GRS RNL processing and bypass involvement by non-IBM resource serialization products when issuing ENQs using QNAME=SIS2000 for this data set name. This keyword is intended to address very specific performance and environmental situations only.

:JCL_PROC_PARMS section keywords

Do not change any of the keyword specifications in this section of BKMINI.

These keywords are reserved for use by product development.

:JCL_DEFAULTS section keywords

Use the information in this topic to specify appropriate values for the keywords in the :JCL_DEFAULTS section of the BKMINI member in the product parameter library. These keyword values provide the default JCL settings for the Automated Critical Data Identification feature.

ACDIDSRS=ACDI.DSN.OUTFILE
Use this keyword to specify the name of the default output file used for the BKMDSRS report.

ACDIOVLP=ACDI.OVERLAP.OUTFILE
Use this keyword to specify the name of the default output file used for the BKMOVLP report.

ESPJOBS=ACDI.ESP.JOBS
Use this keyword to specify the data set name of the ESP LSAR (List Scheduled Activity Report) that is used to extract data from a scheduled activity data set (SADGEN) file to produce a standard scheduled activity report. Until you specify the report name for this keyword, Advanced Backup and Recovery for z/OS cannot run ESP job collection.

Note: The LSAR report must have the following DCB attributes:

- RECFM=FB
- LRECL=133
- BLKSIZE=27930

JOBSSPCE=spacevalue|(CYL(10,10),RLSE)
Use this keyword to specify the space value to be used during job collection for ZEKEJOBS and temporary files. The default value for this keyword is (CYL,(10,10),RLSE).

ZEKEOUTP=ZEKEUTL.SYSPRINT.FILE
Use this keyword to specify the name of the data set created by ZEKEUTL, DD name SYSPRINT, for ASG-Zeke V4.20. This file is deleted by job ASPJOBZI. The default value for this keyword is ZEKEUTL.SYSPRINT.FILE.

ZEKEOUTF=ZEKEUTL.AMPZEKE.FILE
Use this keyword to specify the name of the data set created by ZEKEUTL, DD name AMPZEKE, for ASG-Zeke V4.20. This file is deleted after ZEKEJOBS has been created successfully. The default value for this keyword is ZEKEUTL.AMPZEKE.FILE.

ZEKEJOBS==ACDI.ZEKE.JOBS
Use this keyword to specify the name of the permanent data set created by the Automated Critical Data Identification interface to ASG-Zeke. This file contains job information for all ASG-Zeke jobs. The default value for this keyword is ACDI.ZEKE.JOBS.

ZEKESPCE=spacevalue|(CYL(10,10),RLSE)
Use this keyword to specify the space value to be used for ZEKEOUTP and ZEKEOUTF. These files are deleted after ZEKEJOBS completes successfully. The default value for this keyword is (CYL,(10,10),RLSE).

Appendix C. SMP/E configuration notes

This appendix contains reference information about the SMP/E DDDEFs, Global Zone settings, Product CSI Cluster, Target Zone settings, and DLIB Zone settings.

About the SMP/E DDDEFs

This topic provides reference information about the Target Library DDDEFs, Distribution Library DDDEFs, MVS System Library DDDEFs, SMP/E Log DDDEFS, and SMP/E Library DDDEFs.

DDDEF summary

The following table explains the zone designations used in the tables for the specific DDDEF types.

Zone designation	DDDEF definition and ownership
GBL	DDDEF is defined in and owned by the Global Zone.
gz	DDDEF is defined in and owned by the Global Zone.
TARG	DDDEF is defined in and owned by the Target Zone.
tz	DDDEF is defined in and owned by the Target Zone.
DLIB	DDDEF is defined in and owned by the DLIB Zone.
dz	DDDEF is defined in and owned by the DLIB Zone.

Target Library DDDEFs

DD NAME	Zone			Definition
TBKMASMP	-	TARG	-	PRODUCT.SBKMASMP
TBKMEXEC	-	TARG	-	PRODUCT.SBKMEXEC
TBKMJCL	-	TARG	-	PRODUCT.SBKMJCL
TBKMLOAD	-	TARG	-	PRODUCT.SBKMLOAD – (APF Load Library)
TBKMMSGS	-	TARG	-	PRODUCT.SBKMMSGS
TBKMPARM	-	TARG	-	PRODUCT.SBKMPARM
TBKMPLIB	-	TARG	-	PRODUCT.SBKMPENU
TBKMRSMP	-	TARG	-	PRODUCT SBKMRSMP
TBKMSLIB	-	TARG	-	PRODUCT.SBKMSKEL
TEXTDOC	-	TARG	-	SMP/E Apply Processing - (not a Product Library)

Distribution Library DDDEFs

DD name	Zone			Definition
DBKMASMP	-	tz	DLIB	DLIB–PRODUCT DBKMASMP
DBKMEXEC	-	tz	DLIB	DLIB–PRODUCT.DBKMEXEC
DBKMJCL	-	tz	DLIB	DLIB–PRODUCT DBKMJCL
DBKMMSGS	-	tz	DLIB	DLIB–PRODUCT.DBKMMSGS

DD name	Zone			Definition
DBKMMOD	-	tz	DLIB	DLIB–PRODUCT.DBKMMOD
DBKMPARM	-	tz	DLIB	DLIB–PRODUCT DBKMPARM
DBKMRSMP	-	tz	DLIB	DLIB–PRODUCT.DBKMRSMP
DBKMSLIB	-	tz	DLIB	DLIB–PRODUCT.DBKMSLIB
DBKMPLIB	-	tz	DLIB	DLIB–PRODUCT DBKMPLIB
DEXTDOC	-	tz	DLIB	DLIB–System Dependencies

MVS System Library DDDEFs

DD name	Zone			Definition
#CSSLIB	-	tz	-	SYS1.CSSLIB–(input reference only)
#LINKLIB	-	tz	-	SYS1.LINKLIB–(input reference only)
#MACLIB	-	tz	-	SYS1.MACLIB–(input reference only)
#MODGEN	-	tz	-	SYS1.MODGEN–(input reference only)
SYSLIB	-	tz	-	MACLIB concatenation

SMP/E Log DDDEFs

DD name	Zone			Definition
SMPLOG	gz	tz	dz	<CSI>.SMPLOG–(owned by CSI of Zone)
SMPLOGA	gz	tz	dz	<CSI>.SMPLOGA–(owned by CSI of Zone)

SMP/E Library DDDEFs

DD name	Zone			Definition
SMPHOLD	GBL	-	-	DUMMY (default allocation)
SMPLTS	-	TARG	dz	Apply/Accept Processing
SMPMTS	-	TARG	dz	Apply/Accept Processing
SMPPTS	GBL	tz	dz	Receive Processing
SMPSCDS	-	TARG	dz	Apply/Accept Processing
SMPSTS	-	TARG	dz	Apply/Accept Processing
SMPTLIB	GBL	-	-	Receive/Apply/Accept Processing

Global Zone settings

You can use the SMP/E **UCLIN** command or the IBM SMP/E ISPF dialog to modify the Global Zone options for Advanced Backup and Recovery for z/OS.

Global Zone data sets

The component names for the Global Zone CSI are slightly different from its cluster name defaults. The Component Names follow the Global Zone's data set naming convention.

The SMPLOG and SMPLOGA data sets are dedicated to the Global Zone.

The SMPPTS library is allocated as a PDSE, which is the IBM recommendation.

Dynamically created SMPTLIB data sets do not follow the Global Zone's data set naming convention. Instead, they have a prefix of SMPEHLQ?.SMPTLIB.

GLOBALZONE entry

The SREL must specify SREL(MSC1). All product maintenance installed using SMP/E uses the identity established by this required setting.

The ZONEINDEX subentries are added as new related products and releases are installed. Each zone subentry references a second-level CSI for the product or release being installed.

Global Zone OPTIONS entry

The NOREJECT option is specified so that when SYSMODs are restored from the DLIB Zone back into the Target Zone, they are *not* automatically purged from the SMPPTS and the Global Zone. This is the recommended setting.

The NOPURGE option is not specified. Beware of double negatives. By not specifying NOPURGE, the end effect is that when SYSMODs are ACCEPTed into the DLIB zones, the SYSMODs are automatically purged from the SMPPTS and the Global Zone. This is the recommended setting.

The DSPREFIX and DSSPACE options are not specified because they are overridden by the DDDEF(SMPTLIB) entry in the Global Zone.

The PEMAX option is initially omitted, allowing the default value of PEMAX(2000). If a future product requires a value higher than the default, then it will be documented with the install instructions.

Global Zone UTILITY entries

The UTILITY entry for LKED (the linkage editor) includes the parameters DCBS and LIST(STMT).

The DCBS parameter causes the linkage editor to not change the DCB attributes of the SYSLMOD (target load library) data set. LIST(STMT) significantly reduces the amount of SYSPRINT produced by the linkage editor.

Global Zone DDDEF entries

DDDEF entries for SMPLOG and SMPLOGA reference the log data sets dedicated to the Global Zone.

Dynamically created SMPTLIB data sets do not follow the Global Zone's data set naming convention. Instead, they have a prefix of SMPEHLQ?.SMPTLIB.

DSNTYPE(PDS) is explicitly specified to ensure that a PDSE is not allocated. The IBM utility IEBCOPY cannot copy load modules from tape into a PDSE. The SPACE attributes are initially set at TRACKS(150,15) DIR(88). If a future product requires higher values, then they will be documented with the install instructions.

The DDDEF for SMPHOLD is defined for a dummy data set. This is defined for convenience during RECEIVE processing. HOLDDATA is shipped inline with the PTF maintenance only.

Product CSI Cluster

The Product CSI Cluster contains both the Target and DLIB zones for a particular product and version level (FMID).

The SMPLOG and SMPLOGA data set pair is dedicated for use by all zones residing within this CSI. The SMPLOG data sets are considered to be owned by the CSI, and not necessarily by the zones contained within that CSI.

Target Zone settings

This topic provides information about a subset of Target Zone values and options that are pertinent for Advanced Backup and Recovery for z/OS. You can modify these values and options using the SMP/E **UCLIN** command or through the IBM SMP/E ISPF dialog. Not all of the Target Zone options and entries are included in this topic.

Target Zone data sets

The Target Zone uses the SMPLOG and SMPLOGA data sets associated with the current product CSI, within which it resides.

The LOAD library referenced by DDDEF(TBKMLOAD) should be APF authorized.

The target library referenced by DDDEF(TEXTDOC) is not considered to be a product library. It is intended to serve as a repository for SMP/E managed members for specific SYSMODs that represent external dependencies.

The SMPMTS and SMPSTS libraries are currently not used by Advanced Backup and Recovery for z/OS. However, SMP/E requires their availability.

The SMPLTS data set can be allocated with conservative size attributes. Advanced Backup and Recovery for z/OS involves SMPLTS link-edit activity only when necessary.

TARGETZONE entry

The SREL must specify SREL(MSC1). All maintenance that was installed using SMP/E uses this identity. This setting is required.

The RELATED subentry references this Target Zone's corresponding DLIB Zone that coexists within the same product CSI.

Target Zone DDDEF entries

This topic provides information about the Target Zone SMP DDDEF entries, the Target Zone MVS System DDDEF entries, Target Zone Advanced Backup and Recovery for z/OS DDDEF entries, Target Zone DLIB DDDEF entries, and the Target Zone GLOBAL DDDEF entries.

Target Zone SMP DDDEF entries

DDDEF entries for SMPLOG and SMPLOGA reference the log data sets dedicated to the current product CSI.

Target Zone MVS System DDDEF entries

DDDEF(#MACLIB) refers to data set SYS1.MACLIB on the current MVS system, and is used for the Assembler SYSLIB concatenation.

DDDEF(#MODGEN) refers to data set SYS1.MODGEN on the current MVS system, and is used for the Assembler SYSLIB concatenation.

DDDEF(#LINKLIB) refers to data set SYS1.LINKLIB on the current MVS system, and is used as input for the link-edit processing of certain load modules.

DDDEF(#CSSLIB) refers to data set SYS1.CSSLIB on the current MVS system, and is used as input for the link-edit processing of certain load modules.

Target Zone Advanced Backup and Recovery for z/OS DDDEF entries

DDDEF(TBKMLOAD) allocates to the product LOAD library that is APF authorized.

DDDEF(TBKMPARM) allocates to the product PARMLIB library.

All other DDDEF entries allocate to their respective product-specific libraries.

DDDEF(TEXTDOC) is a target library, but is not considered part of the product library set.

Target Zone DLIB DDDEF entries

All DLIB DDDEF entries are defined in the Target Zone to support RESTORE processing.

Target Zone GLOBAL DDDEF entries

DDDEF(SMPPTS) is required for APPLY processing.

DDDEF entry for SMPTLIB is not required if the TLIB libraries were cataloged at RECEIVE time.

DLIB Zone settings

This topic provides information about a subset of the DLIB Zone values and options for Advanced Backup and Recovery for z/OS. You can modify these values and options using the SMP/E **UCLIN** command or through the IBM SMP/E ISPF dialog. This topic does not include descriptions of all the DLIB Zone options and values.

DLIB Zone data sets

The DLIB Zone uses the SMPLOG and SMPLOGA data sets associated with the current Product CSI, within which it resides.

DLIBZONE entry

The SREL must specify SREL(MSC1). All maintenance installed using SMP/E uses this identity. This setting is required.

The ACCJCLIN subentry should be specified. The ACCJCLIN parameter causes RESTORE processing to ignore SYSMOD chaining problems with the SMPSCDS data set for LMOD JCLIN controls. (SYSMOD chaining in the SMPSCDS can become corrupted due to normal use of the APPLY REDO, UCLIN, and JCLIN commands.) Use of the ACCJCLIN attribute is highly recommended.

The RELATED subentry references this DLIB Zone's corresponding Target Zone that coexists within the same product CSI. It is assumed that ACCEPT processing for this DLIB zone will be done through only this one target zone.

DLIB Zone DDDEF entries

This topic provides information about the DLIB Zone Distribution DDDEF entries, the DLIB Zone Target DDDEF entries for product libraries, the DLIB Zone SMP DDDEF entries, DLIB Zone MVS System DDDEF entries, DLIB Zone SMP/E Target DDDEF entries, and DLIB Zone GLOBAL DDDEF entries.

DLIB Zone Distribution DDDEF entries

It is recommended that all DLIB data sets be allocated on SMS managed volumes as DSNTYPE=LIBRARY (PDSE) data sets.

DLIB Zone Target DDDEF entries – product libraries

The Advanced Backup and Recovery for z/OS product libraries are not required in the DLIB zone. They are not referenced by ACCEPT processing.

DLIB Zone SMP DDDEF entries

DDDEF entries for SMPLOG and SMPLOGA reference the log data sets that are dedicated to the current Product CSI.

DLIB Zone SMP/E Target DDDEF entries

DDDEF entries for the libraries SMPMTS SMPSTS SMPLTS SMPSCDS from the RELATED target zone must be defined in the DLIB zone to support ACCEPT processing.

DLIB Zone GLOBAL DDDEF entries

DDDEF(SMPPTS) required for ACCEPT processing.

DDDEF entry for SMPTLIB is not required if the TLIB libraries were cataloged at RECEIVE time.

Appendix D. Extended ACS masking characters in filtering pattern masks

Many Advanced Backup and Recovery for z/OS commands allow you to specify filtering masks that use the extended ACS masking characters described in this topic.

Mask	Hex	Description
*	5C	Represents 0–*n* characters. You can use this character before an item or after an item (or both) to designate a wildcard character position. **Note:** You cannot use this character in the middle position of a single item except for data set names. If you need to represent 0-*n* characters in the middle portion of a single item that is not a data set name, use the % character. Consider *PROD as an example. An item is selected if the last four characters are PROD, regardless of how many characters precede PROD or what those preceding characters are. Consider *ZREM* as another example. In this case, an item is selected if the string ZREM occurs at any position in that item's name.
%	6C	Represents a single-character placeholder value that can be any alphanumeric or special character. You can use % at any position in the mask. You can use % more than once within a mask. In the example CRM%%ER6, an item is selected if it has an 8-character length, the first three characters are CRM, any two characters occupy the fourth and fifth positions, and the last three characters in its name are ER6.
<	4C	Represents a single-character placeholder value that can be any *alphabetic* character. You can use < at any position in the mask. You can use < more than once within a mask. In the example CR<<ER*, an item is selected if the first two characters of its name are CR, any two alphabetic characters occupy the third and fourth positions, ER occupies the fifth and sixth positions, and 0-*n* characters of any type in any subsequent positions.
>	6E	Represents a single-character placeholder value that can be any *numeric* character. You can use > at any position in the mask. You can use > more than once within a mask. In the example CR<<ER*, an item is selected if the first two characters of its name are CR, any two numeric characters occupy the third and fourth positions, ER occupies the fifth and sixth positions, and 0-*n* characters of any type in any subsequent positions.

Mask	Hex	Description
**	5C5C	Represents a partially qualified data set name. ** allows compatibility with standard ACS and DFSMSdss filtering masks. You can the * or ** characters at any position within the mask to represent a wildcard node. In th example CRMF*.*VER.**, a data set is selected if it has two or more qualifiers, the first qualifier contains the string CRMF in positions 1-4, and the final three positions of the second qualifier contain the string VER. **Note:** You can use any combination of *, **, %, <, and > characters within the mask that you specify.
!	5A	Represents a single-character placeholder value that must be a national character. According to the US EBCDIC code set, national characters are @, #, and $. You can use ! at any position within a filtering mask.

Appendix E. Batch processing

Automated Critical Data Identification batch job submission sequence

Job collection and batch SMF Scan functions are performed by submitting batch jobs. At an appropriate point during these data collection steps, you submit the batch job. As you submit a batch job, Automated Critical Data Identification displays a series of panels that allow you to modify the batch job before you submit it for processing.

Job statement verification

The first panel displayed during the batch job submission process is a job statement verification panel. For example:

```
         REVIEW and/or CHANGE JOB/JCL Statements as Required:
                Use ENTER To Submit JOB; or PF3 To Abort

     Randomize Job Name to Eight Characters ? ==> YES  ( YES / NO )

 //MSIS04TZ  JOB   (,101),'I. M. USER',CLASS=Z,REGION=0M,
 //  NOTIFY=MSIS04T,MSGLEVEL=(1,0),MSGCLASS=X
 //* NEW BATCH JOB
 //*
```

Figure 7. Job statement verification

The information on this panel is obtained from your TSO ISPF 0.2 option, which defines a default job statement for batch jobs. If needed, change the fields on this panel to match your installation's requirements for batch jobs.

Job submission preferences

The next panel, displayed after the job card information has been provided, is a job submission preferences panel. This panel allows you to either immediately submit the batch job, or to edit the generated JCL prior to submitting it for processing. For example:

```
Command ==>

     Current Application:  CA7MAXJB
    ACDI AUTOBUILD Phase:  JOB COLLECTION

     SELECT All The Options You Wish.

        _  EDIT JCL Prior To Job Submit
        _  SUBMIT JOB For Execution

  Make Selections and ENTER ; Use PF3 To Terminate.
```

Figure 8. Job submission preferences

Note the following items on this panel:

Current® Application
The name of the application for which data collection is being performed.

AUTOBUILD Phase
The name of the primary data collection step (for example, job collection, or SMF Scan).

EDIT JCL Prior to Submit
ISPF Edit displays the JCL that has been generated for the batch job. You can view and optionally modify the JCL before submitting it using the TSO **SUBMIT** command. Select this option by typing an S in this field.

SUBMIT JOB For Execution
The job is submitted immediately. Select this option by typing an S in this field.

When the batch job has completed, a number of pending messages are sent. You can display these messages using option **5 (STATUS)** or by pressing F12 on the **Application AutoBuild Options** menu.

Critical Backup Tracking and Inventory batch processing

This information describes the batch processing options that are available with Critical Backup Tracking and Inventory.

Critical Backup Tracking and Inventory offers several batch processing options:

- Batch jobs—Sample batch job JCL is available in the Advanced Backup and Recovery for z/OS JCL library. You copy these jobs, modify them as needed, and submit them for processing.
- Advanced Backup and Recovery for z/OS — You can use the BKMBPROC member of the Advanced Backup and Recovery for z/OS library to perform specific batch processing. Batch reports are submitted in this manner.

Batch jobs

This information describes the batch jobs available in the product library.

BKMEXPIR

Schedule the BKMEXPIR job on a regular basis to remove old job name cycles for Critical Backup Tracking and Inventory tracked backups and expired DFSMShsm AUTODUMP versions from the Advanced Backup and Recovery for z/OS Inventory Data Set.

Note: You can set token EXPIRE_BAD_RC (in the CBTI_DUMP_BACKUP_PROCESS_OPTIONS section of the BKMINI member in the product parameter library) to serve as a threshold return code value for all backups tracked by Critical Backup Tracking and Inventory. Any backup job that terminates with a return code equal to or greater than the threshold is expired.

Purpose

You must execute the BKMEXPIR job to expire and remove from the Advanced Backup and Recovery for z/OS Inventory Data Set any backups or DFSMShsm AUTODUMPs that are no longer valid. If a backup job contains multiple backups, the backup job name will not be removed from the Advanced Backup and Recovery for z/OS Inventory Data Set until all backups in the job are no longer valid.

In addition to removing old backups, BKMEXPIR removes old application cycle backups when using the Automated Critical Data Identification Backup End Options function.

BKMEXPIR removes old application cycle backups as defined by the **Backup Cycles to Retain** parameter (on the Automated Critical Data Identification **Application SETUP Options** panel).

Example

The BKMEXPIR JCL is shown in the example:

```
//BKMEXPIR JOB ,'BKMBATCH',CLASS=A                    <<< SEE #1
// JCLLIB ORDER=prefix.value..SBKMJCL                          <<< SEE #2
//*******************************************************************
//* 1. CHANGE JOB STATEMENT TO LOCAL STANDARDS
//* 2. CHANGE ALL prefix.value TO LOCAL STANDARDS
//*******************************************************************
//*******************************************************************
//* THIS JOB IS USED TO EXPIRE OUTDATED BACKUP                      *
//* CYCLES FROM  THE IDS                                            *
//*******************************************************************
//BKM      EXEC BKMBPROC
//BKMBATCH.SYSIN  DD  *
  BKMEXPIRE
//
```

BKMADUMP

DFSMShsm AUTODUMPs are not automatically tracked by Critical Backup Tracking and Inventory. To update the Advanced Backup and Recovery for z/OS Inventory Data Set with DFSMShsm AUTODUMPs, schedule the BKMADUMP batch job.

Purpose

The BKMADUMP batch job updates the DFSMShsm AUTODUMP inventory information in the Advanced Backup and Recovery for z/OS Inventory Data Set.

The DFSMShsm AUTODUMP information is retrieved from the DFSMShsm BCDS. The BKMADUMP job must run on an LPAR with DFSMShsm active. Otherwise, the BKMADUMP job ends without updating the DFSMShsm AUTODUMP information in the IDS.

Schedule the BKMADUMP job at a time when the DFSMShsm AUTODUMP process should have completed. Trigger it based on DFSMShsm `AUTODUMP complete` messages, or run it as needed.

Note: Use the BKMEXPIR job to remove expired DFSMShsm AUTODUMP versions from the IDS.

Example

```
/*
//AUTODUMP JOB ,'BKM HSM AUTODUMP',CLASS=A      <<< SEE #1
// JCLLIB ORDER=prefix.value.SBKMJCL                          <<< SEE #2
//*******************************************************************
//* HSM AUTODUMP                                                    *
//* SAMPLE JOB TO UPDATE THE IDS WITH ANY NEW AUTODUMP CYCLES       *
//*******************************************************************
//*******************************************************************
//* 1. CHANGE JOB STATEMENT TO LOCAL STANDARDS                      *
//* 2. CHANGE ALL prefix.value TO LOCAL STANDARDS                *
//*******************************************************************
//HSMBKMA  EXEC BKMBPROC
//BKMBATCH.SYSIN DD *
 HSMADUPD
/*
```

Batch reports

Batch reports are generated in batch mode using the BKMBPROC member of the Advanced Backup and Recovery for z/OS JCL library.

Generating batch reports

You submit batch reports using REPORT statements as input to the BKMBPROC procedure. Any number of REPORT statements can be included in a BKMBPROC invocation.

Note: Note:

- Only one report type can be specified with each REPORT statement.

- You can continue multiple lines using either a plus sign (+) or a minus sign (-) and by leaving at least one blank after the parameter value. Continued lines can begin in any position.

Example

This example shows a request for multiple reports: an Overlap by Data Set report (OVERLAPD) and a Data Set List by Data Set Name, Date report (DSNLIST):

```
//STEPNAME EXEC BKMBPROC
//BKMBATCH.SYSIN DD *
  REPORT OVERLAPD-CBTI DS(dsname1)
  REPORT DSNLIST-CBTI DS(dsname1)
```

OVERLAPA-CBTI (Overlap by Application report)

Use OVERLAPA-CBTI to submit a report request for the Overlap by Backup Event report in batch.

Required parameters

OVERLAPA-CBTI
Create the overlap by event or jobname report.

APPL ***applname or mask***
Specifies the application name or application name masks to obtain a data set list which is then used to determine if other applications are tracking the same data sets. If the same data set is found in other applications the report will be created.

Example

In this example, only one application is searched for overlaps.

```
//STEPNAME EXEC BKMBPROC
//BKMBATCH.SYSIN DD *
  REPORT OVERLAPA-CBTI APPL(applname)
```

OVERLAPD-CBTI (Overlap by Data Set report)

OVERLAPD-CBTI submits a report request for the Overlap by Data Set report in batch.

Required parameters

OVERLAPD-CBTI
Parameter to create Overlap by Data Set report.

DS ***dsn***
Specifies the names or name masks of the data sets to be searched for to create the overlap by data set report.

Example

In this example, only one data set is searched for overlaps.

```
//STEPNAME EXEC BKMBPROC
//BKMBATCH.SYSIN DD *
  REPORT OVERLAPD-CBTI DS(dsname1)
```

A second method is to specify more than one data set name. In this example, multiple overlap by data set searches will be performed. Overlaps will be searched for 'dsname1' and any data set matching the mask 'dsname2*'.

```
//STEPNAME EXEC BKMBPROC
//BKMBATCH.SYSIN DD *
  REPORT OVERLAPD-CBTI DS(dsname1 dsname2*)
```

OVERLAPE-CBTI (Overlap by Backup Event report)

Use OVERLAPE-CBTI to submit a report request for the Overlap by Backup Event report in batch.

Required parameters

OVERLAPE-CBTI
Create the overlap by event or job name report.

EV ***dsn***
Specifies the names or name masks of the aggregate events or job names to be searched for to create the overlap by event or job name report.

Example

The following example, only one event or job name will be included in the report.

```
//STEPNAME EXEC BKMBPROC
//BKMBATCH.SYSIN DD *
  REPORT OVERLAPE-CBTI EV(DS1)
```

OVERLAPJOBGRP (Overlaps by Job Group List report)

Use OVERLAPJOBGRP to submit a report request for the Overlaps by Job Group List report in batch.

Required parameters

OVERLAPJOBGRP
Create an Overlaps by Job Group report.

JOB ***jobname***
Specifies the names or name masks of the aggregate events or job names to determine which job name is tracked by more than one job group.

Example

In the following example one job name is checked to see if more than one job group is tracking it.

```
//STEPNAME EXEC BKMPROC
//BKMBATCH.SYSIN DD *
  REPORT OVERLAPJOBGRP JOB(jobname1)
```

DSNLIST-CBTI (Data Set List by Data Set Name, Date report)

Specifies DSNLIST-CBTI to submit a report request for the Data Set List by Data Set Name, Date report in batch.

Required parameters

DSNLIST-CBTI
Create the data set lists report type.

DS ***dsn***
Specifies the names or name masks of the data sets to be listed in the data set list report.

Optional parameters

DATE

fromdate fromdatemask
Specifies the starting date or mask of the data sets or events to be listed in the data set list report.

todate todatemask
Specifies the ending date or mask of the data sets or events to be listed in the data set list report. For relative date masking use * for today, and *-nnn for today minus a number of days, up to 999. Relative date masking is permitted. For example, you can specify an asterisk (*) to indicate the current date, or you could specify *-5 for five days before the current date (up to *-999).

JOB ***jobname***
Specifies any number of job names or masks of job names to be listed in the data set list report.

EXCLUDE RC
Specify RC to exclude event Return code >= Entered Value

For example,

- 4 - show only events with RC < 4
- 8 - show only events with RC < 8

Note: The HSM backup events will not be displayed if you opt for this field.

TYPE
Specifies a utility type to search for data sets in:

A
Data sets backed up by ABARS.

C
Data sets backed up by ICF Catalog Management utility.

D
Data sets backed up by DFSMSdss.

F
Data sets backed up by FDR.

G
Data sets backed up by IEBGENER, ICEGENER, IEBCOPY, or SYNCGENR.

H
Data sets backed up by HSM Incremental, ARCINBAK, and HSM Auto dump.

I
Data sets backed up by IDCAMS REPRO or EXPORT.

M
Data sets backed up by CFCAMS.

K
Data sets backed up by ADARUN.

O
Data sets backed up by ICETOOL.

S
Data sets backed up by ICEMAN, SORT, or SYNCSORT.

V
Data sets backed up by CA-FAVER.

Z
Data sets backed up by VSAM Assist.

Example

In this example, only one data set is included in the report and the backup type defaults to all backup types.

```
//STEPNAME EXEC BKMBPROC
//BKMBATCH.SYSIN DD *
  REPORT DSNLIST-CBTI DS(dsname1)
```

A second method is to include more than one job name in the REPORT DSNLIST-CBTI statement. In this example, any data set matching dsname* in jobnames jobname1 or jobname2* type DSS, between today and 2001/02/29, will be displayed.

```
//STEPNAME EXEC BKMBPROC
//BKMBATCH.SYSIN DD *
  REPORT DSNLIST-CBTI DS(dsname*)               -
                      JOB(jobname1 jobname2*)   -
                      TYPE(D)                   -
                      DATE(*,2001/02/29)
```

```
//STEPNAME EXEC BKMBPROC
//BKMBATCH.SYSIN DD *
  REPORT DSNLIST-CBTI
                       DS(dsname*)                          -
                      RCD(8  )                              -
                      TYPE(ACDFGHIMKORSTUVWZ)               -
```

DSNLIST-O (Data Set List by Jobname, Data Set Name, Date with Outputs report)

Specify DSNLIST-O to submit a report request for the Data Set List by Jobname, Data Set Name, Date with Outputs report in batch.

Required parameters

DSNLIST-O
Creates the Data Set List by Job, Data Set Name, Date with Outputs report.

JOB ***jobname***
Specifies the job names or job name masks to display input data sets by job name with their associated output data sets.

Example

In this example, only one job name is included in the report.

```
//STEPNAME EXEC BKMPROC
//BKMBATCH.SYSIN DD *
  REPORT DSNLIST-O JOB(jobname1)
```

FVDLIST (Full Volume Dump List report)

Specify FVDLIST to submit a report request for the Full Volume Dump List report in batch.

Required parameters

FVDLIST
Create the Full Volume Dump lists report.

VOL ***volume***
Specifies the volume names or volume name masks of the full volume dumps to be listed.

Optional parameters

DATE

fromdate fromdatemask
Specifies the starting date or mask of the data sets or events to be listed in the data set list report.

todate todatemask
Specifies the ending date or mask of the data set(s) or event(s) to be listed in the data set list report. For relative date masking use * for today, and *-nnn for today minus a number of days, up

to 999. Relative date masking is permitted. For example, you can specify an asterisk (*) to indicate the current date, or you can specify *-5 to indicate five days before the current date (up to *-999).

Examples

In this example, only one volume is included in the report.

```
//STEPNAME EXEC BKMBPROC
//BKMBATCH.SYSIN DD *
  REPORT FVDLIST VOL(volname1)
```

A second method is to specify more than one volume name. In this example, any volume matching 'volname*' between today and 2001/02/29 will be displayed.

```
//STEPNAME EXEC BKMBPROC
//BKMBATCH.SYSIN DD *
  REPORT FVDLIST VOL(volname*)            -
              DATE(*,2001/02/29)
```

JOBLIST (Jobname List report)

Specify JOBLIST to submit a report request for the Jobname List report in batch.

Required parameters

JOBLIST
Create the Job List report.

JOB ***jobname***
Mutually exclusive with JOBGRP. Either JOBGRP or JOB must be specified. Specifies the job names or job name masks of the job names to be listed in the Jobname List report.

JOBGRP ***jobname***
Mutually exclusive with JOBGRP. Either JOBGRP or JOB must be specified. Specifies the job group name of the job names to be listed in the Jobname List report. JOBGRP must specify an absolute name (up to eight characters in length).

Optional parameters

DATE

fromdate fromdatemask
Specifies the starting date or mask of the job names to be listed in the Jobname List report.

todate todatemask
Specifies the ending date or mask of the job names to be listed in the Jobname List report. For relative date masking use * for today, and *-nnn for today minus a number of days, up to 999. Relative date masking is permitted. For example, you can specify an asterisk (*) to indicate the current date, or you can specify *-5 to indicate five days before the current date (up to *-999).

DATETYPE
Valid values are:

- S — (Default) Specify S to use Start Date when comparing to a specified From Date or To Date.
- E — Specify E to use End Date when comparing to a specified From Date or To Date.

TIME

fromtime
fromtime requires a *fromdate* be specified when searching for job names. If *todate* is not specified, then only job names matching the date and time specified will be reported. Or specify a To Date or Time to report on a range of job names by date or time.

totime
totime requires a *todate* be specified searching for job names. If *fromdate* is not specified, then only job names matching the date and time specified will be reported. Or specify a To Date or Time to report on a range of job names by date or time.

GEN*nn*
Specifies a job name generation number to report only those job names matching the generation specified.

Examples

In this example, only one job name is included in the report. If found, the job name will be displayed.

```
//STEPNAME EXEC BKMBPROC
//BKMBATCH.SYSIN DD *
  REPORT JOBLIST JOB(jobname1)
```

A second method is to specify more than one job name. In this example, any job name matching 'jobname*' between today and 2001/02/29 will be displayed.

```
//STEPNAME EXEC BKMBPROC
//BKMBATCH.SYSIN DD *
  REPORT JOBLIST JOB(jobname*)                -
               DATE(*,2001/02/29)
```

NOJOBGRP (Jobnames Not Tracked By a Job Group report)

Use NOJOBGRP to submit a report request for the Jobnames Not Tracked By a Job Group report in batch.

Required parameters

NOJOBGRP
Create the Jobnames Not Tracked By a Job Group report.

JOB *jobname*
Specifies the job names or job name masks to determine if the matching job name is being tracked by a job group.

Example

In this example, only one job name is searched.

```
//STEPNAME EXEC BKMBPROC
//BKMBATCH.SYSIN DD *
  REPORT NOJOBGRP JOB(jobname1)
```

NOBACKUP (What Is Not Backed Up report)

Use report NOBACKUP to find out which data sets are not backed up in your environment.

Required parameters

This report will help ensure that you are not missing critical data sets. The NOBACKUP report compares data sets on your volumes or in your ICF user catalogs (or both), to the Advanced Backup and Recovery for z/OS Inventory Data Set, and optionally to the DFSMShsm BCDS. Critical Backup Tracking and Inventory will identify data sets that are not recorded in either and thus either have not been tracked yet, or do not have a backup. The NOBACKUP report automatically excludes data sets identified in the Advanced Backup and Recovery for z/OS Inventory Data Set as a backup output file.

Note: You are responsible for the maintenance of the ICF catalogs. If data set entries exist in the catalog that are either no longer on disk, or are cataloged to nonexistent volumes, it is considered a catalog maintenance issue. Similarly, if there are disk entries with no associated catalog entry, it is considered

a DASD maintenance issue. The What Isn't Backed Up Report identifies these entries, but it does not eliminate them.

NOBACKUP
Create a report of what is not backed up.

TARGET
Identifies what the data is compared to: the Advanced Backup and Recovery for z/OS Inventory Data Set, the DFSMShsm BCDS, or both. It will compare to all data sets if the optional SOURCE parameter has not been specified.

IDS
(Default) The source will be compared to the Advanced Backup and Recovery for z/OS Inventory Data Set.

BCDS
The source will be compared to the DFSMShsm BCDS.

SORTYBYNAME |SORTBYVOL
SORTBYNAME and SORTBYVOL are mutually exclusive.

SORTYBYNAME
(Default) Specify this parameter to sort the report by data set name.

SORTBYVOL
Specify this parameter to sort the report by volume.

Optional parameters

SOURCE
Identifies selected data to compare to; either by data set name, by data sets on volsers, or by user catalog. A SOURCE parameter is recommended to prevent all data sets on the system from being compared. Without a SOURCE parameter, a report will be created that is too large to use easily.

Note: If SOURCE is not specified, TARGET will be compared to all data sets.

INCLDSN|EXCLDSN
INCLDSN and EXCLDSN allow you to specify data set names to be included or excluded in the compare. Specifies fully qualified data set names, or masks.

Note: INCLDSN and EXCLDSN can be used in conjunction with INCLUCAT or EXCLUCAT and INCLVOL or EXCLVOL.

INCLUCAT|EXCLUCAT
INCLUCAT and EXCLUCAT allow you to specify user catalog names to be included or excluded in the compare. Specifies fully qualified catalog names, or masks. If this option is used, the user catalog is used as the source and there is no volume lookup.

Note: INCLUCAT and EXCLUCAT can be used in conjunction with INCLDSN or EXCLDSN and INCLVOL or EXCLVOL.

INCLVOL|EXCLVOL
INCLVOL and EXCLVOL allow you to specify volume names to be included or excluded in the compare. Specifies fully qualified volume names, or masks. If this option is used, the VTOC is used as the source and there is no catalog lookup.

Note: INCLVOL and EXCLVOL can be used in conjunction with INCLDSN or EXCLDSN and INCLUCAT or EXCLUCAT.

Examples

Consider an example in which all K* data sets (excluding KR* data sets) on volumes that begin with T or S are found. Data set backups found in the Advanced Backup and Recovery for z/OS Inventory Data Set and DFSMShsm BCDS are compared and a report sorted by data set name is generated.

```
//RPTNOBKP EXEC PGM=BKM50100,REGION=0M
//STEPLIB  DD   DISP=SHR,DSN=BKM.PROD.LOAD
//INI      DD   DISP=SHR,DSN=BKM.PROD.PARMLIB(BKMINI)
//BRMIDS   DD   DISP=SHR,DSN=BKM.PROD.INVDB
//SYSPRINT DD   SYSOUT=*
//REPORT   DD   SYSOUT=*
//SYSABEND DD   SYSOUT=*
//SNAP     DD   SYSOUT=*
//SYSIN    DD   *
  REPORT NOBACKUP                 -
         SOURCE(INCLDSN(K*)       -
                EXCLDSN(KR*)      -
                INCLVOL(T* S*))   -
         TARGET(IDS BCDS)         -
         SORTBYNAME
//
```

The following figure shows the output from example one

```
******************************** TOP OF DATA ************************
*  WHAT ISN'T BACKED UP                                              *
*                                                                    *
*  NO BACKUP FOUND FOR THE FOLLOWING:                *
*  NOTE: "+" = MULTI-VOLUME DATASET                  *
*****************************************************
  DATASET NAME                                    VOLSER
-----------------------------------------------------
  KAS.C.TEST1                                     TSO003
  KAS.C.TEST2                                     TSO001
  KAS.C.TEST3                                     TSO001
  KAS.C.TEST4                                     TSO004
  TOTAL DATASETS FOUND:              4
```

Now consider an example in which all KAS.C.** data sets anywhere on the system are found, except for KAS.C.TEST1, and excluding any data sets found on volser TSO003.

```
//RPTNOBKP EXEC PGM=BKM50100,REGION=0M
//STEPLIB  DD   DISP=SHR,DSN=BKM.PROD.LOAD
//INI      DD   DISP=SHR,DSN=BKM.PROD.PARMLIB(BKMINI)
//BRMIDS   DD   DISP=SHR,DSN=BKM.PROD.INVDB
//SYSPRINT DD   SYSOUT=*
//REPORT   DD   SYSOUT=*
//SYSABEND DD   SYSOUT=*
//SNAP     DD   SYSOUT=*
//SYSIN    DD   *
REPORT NOBACKUP                  -
       SOURCE(INCLDSN(KAS.C.**)  -
              EXCLDSN(KAS.C.TEST1)  -
              EXCLVOL(TSO003))    -
       TARGET(IDS BCDS)
//
```

The following figure shows the output from example two:

```
*****************************************************
*  WHAT ISN'T BACKED UP
*  NO BACKUP FOUND FOR THE FOLLOWING:                *
*  NOTE: "+" = MULTI-VOLUME DATASET                  *
*****************************************************
  DATASET NAME                                    VOLSER
-----------------------------------------------------
  KAS.C.TEST2                                     TSO001
  KAS.C.TEST3                                     TSO001
  KAS.C.TEST4                                     TSO004
  TOTAL DATASETS FOUND:              3
```

In a final example, all D* data sets in any user catalog that matches the user catalog name mask, USERCAT.**, except for USERCAT.UCAT1, are found.

```
//RPTNOBKP EXEC PGM=BKM50100,REGION=0M
//STEPLIB  DD   DISP=SHR,DSN=BKM.PROD.LOAD
//INI      DD   DISP=SHR,DSN=BKM.PROD.PARMLIB(BKMINI)
//BRMIDS   DD   DISP=SHR,DSN=BKM.PROD.INVDB
//SYSPRINT DD   SYSOUT=*
//REPORT   DD   SYSOUT=*
//SYSABEND DD   SYSOUT=*
//SNAP     DD   SYSOUT=*
//SYSIN    DD   *
REPORT NOBACKUP                        -
       SOURCE(INCLUCAT(USERCAT.**)     -
              INCLDSN(D*.**)           -
              EXCLUCAT(USERCAT.UCAT1))-
       TARGET(IDS BCDS)                -
//
```

The following figure shows the output from example three:

```
*****************************************************
* WHAT ISN'T BACKED UP                              *
*  NO BACKUP FOUND FOR THE FOLLOWING:               *
*  NOTE: "+" = MULTI-VOLUME DATASET                 *
*****************************************************
  DATASET NAME                                VOLSER
-----------------------------------------------------
  DCF140.ADCFASM                              Z4DIS1
  DCF140.ADCFIMAG                             Z4DIS1
  DCF140.ADCFMAC                              Z4DIS1
  DCF140.ADCFSAMP                             Z4DIS1
  DCF140.AEDFDOW1                             Z4DIS1
  TOTAL DATASETS FOUND:              5
```

ACDI2BKM-NB (Critical in Automated Critical Data Identification with No Critical Backup Tracking and Inventory Backup Record report)

Specify ACDI2BKM-NB to submit a report request for the Critical in Automated Critical Data Identification with No Critical Backup Tracking and Inventory Backup Record report in batch.

Required parameters

ACDI2BKM-NB
Creates the Critical in Automated Critical Data Identification with No Critical Backup Tracking and Inventory Backup Record report.

APPL ***application name***
Specify the Automated Critical Data Identification application name for which a BKUPEND was performed.

Optional parameters

INCLHSM
Specify this parameter to include the DFSMShsm incremental backups in the search.

INCLPHY
Specify this parameter to include full volume and physical by data set dumps in the search.

EXCL-ALLOCATE
Specify this parameter to bypass ALLOCATE data sets in the search.

EXCL-ACCOMPANY
Specify this parameter to bypass ACCOMPANY data sets in the search.

DS *dsn*
The Critical Backup Tracking and Inventory backup inventory status for the application specified is limited to the data set name or mask identified by this parameter.

Example

The following example shows how to initiate the Critical in Automated Critical Data Identification with No Critical Backup Tracking and Inventory Backup Record reports in batch mode using procedure BKMBPROC.

```
//STEPNAME EXEC BKMBPROC
//BKMBATCH.SYSIN DD *
  REPORT ACDI2BKM-NB APPL(TEST)
```

ACDI2BKM-B (Critical in Automated Critical Data Identification with Critical Backup Tracking and Inventory Backup Record report)

Specify the ACDI2BKM-B parameter to submit a report request for the Critical in Automated Critical Data Identification with Critical Backup Tracking and Inventory Backup Record report in batch.

Required parameters

This report describes the critical data sets found in Automated Critical Data Identification for which there is a matching backup record in Critical Backup Tracking and Inventory. The report is listed by Automated Critical Data Identification application and displays the input data set name backed up, output backup data set name, job name that created the backup, date, time, type of backup, volser of the output data set, and tape location of the output data set (if one exists).

Note: Critical Backup Tracking and Inventory automatically recognizes the tape management systems CA1 and RMM. No JCL or BKMINI changes are required.

ACDI2BKM-B
Creates the Critical in Automated Critical Data Identification with Critical Backup Tracking and Inventory Backup Record report.

APPL *application name*
Specifies the Automated Critical Data Identification application name for which a BKUPEND was performed.

Optional parameters

INCLHSM
Specify this parameter to include the DFSMShsm incremental backups in the search.

INCLPHY
Specify this parameter to include full volume and physical by data set dumps in the search.

EXCL-ALLOCATE
Specify this parameter to bypass ALLOCATE data sets in the search.

EXCL-ACCOMPANY
Specify this parameter to bypass ACCOMPANY data sets in the search.

DS *dsn*
The Critical Backup Tracking and Inventory backup inventory status for the application specified will be limited to the data set name or mask identified by this parameter.

Example

The following example shows how to specify a Critical in Automated Critical Data Identification with Critical Backup Tracking and Inventory Backup Record report in batch mode using procedure BKMBPROC.

```
//STEPNAME EXEC BKMBPROC
//BKMBATCH.SYSIN DD *
  REPORT ACDI2BKM-B APPL(TEST)
```

BACKUP-END (Backup End Summary report)

Specify BACKUP-END to submit a report request for the Backup End Summary report in batch. This report produces the application data set backup status found at the time BKUPEND processing was performed.

Required parameters

BACKUP-END
Creates the Backup End Summary report.

APPL ***application name***
Specifies the Automated Critical Data Identification application name to include in the report.

CYCLE ***n***
Specifies the BKUPEND processing relative cycle number on which to report, where *n* is an integer in the range 0 to -9999.

Example

In this example, a BACKUP-END summary report is produced for the current BKUPEND processing cycle for the ACDI application TEST.

```
//STEPNAME EXEC BKMBPROC
//BKMBATCH.SYSIN DD *
  REPORT BACKUP-END APPL(TEST) CYCLE(0)
```

CBTI-TAPEPULL (Tape Pull report)

Specify these options to submit a report request for the Tape Pull report in batch.

Required parameters

CBTI-TAPEPULL
Creates the Tape Pull report.

JOB ***jobname***
Mutually exclusive with JOBGRP. Either JOBGROUP or JOB must be specified. Specify the job names or job name masks to be listed in the Tape Pull report.

JOBGROUP ***groupname***
Mutually exclusive with JOB. Either JOBGROUP or JOB must be specified. Specify the job group name be listed in the Tape Pull report. JOBGROUP must specify an absolute name (up to 8 characters in length).

Optional parameters

DATE

fromdate fromdatemask
Specifies the starting date or mask of the data sets or events to be listed in the Tape Pull report.

todate todatemask
Specifies the ending date or mask of the data sets or events to be listed in the Tape Pull report. For relative date masking specify * for today, and *-nnn for today minus a number of days, up to

999. Relative date masking is permitted. For example, you can specify an asterisk (*) to indicate the current date, or you could specify *-5 for five days before the current date (up to *-999).

GEN

fromgen
Specifies the starting relative generation (nnnn) of the events to be listed in the tape pull report.

togen
Specifies the ending relative generation of the events to be listed in the tape pull report.

ODS |ODSN |INCLODSN
Specifying this parameter causes tape data set names that match this to be included in processing. The net result is to cause all tape data set names to be excluded except those that match the tape data set name or tape data set name mask.

OVOL |INCLOVOL
Specifying this parameter causes tape volume serials that match this to be included in processing. The net effect is to cause all volumes to be excluded except those that match the volume serials or volume serials mask.

INCLLOC
Specifying this parameter causes tape whose location matches this location or location mask to be included in processing. The net effect is exclude all locations except those that match the location or location mask.

EXCLLOC
Specifying this parameter causes tape whose location matches this location or location mask to be excluded from processing.

EXCLJOB
Specifying this parameter causes job names that match the job name or job name masks to be excluded from processing.

EXCLODSN
Specifying this parameter causes tape data set names that match tape data set names or tape data set name masks to be excluded from processing.

EXCLOVOL
Specifying this parameter causes tape volume serials that these volume serials or volume serials masks this to be excluded from processing.

INCLPHYDUMP
Specify Y for the report to include tapes that are physical dumps. The default value for this parameter is N.

COPY
To have the secondary (or other) copy of a backup used instead of the primary backup, specify this parameter with the number of the backup copy you want used. The default is 1.

BADBKUPRC
This parameter limits the backups searched to only those that had a return code less than the number you specify (*nn*). The default is 8.

SORTBYNAME
This parameter is mutually exclusive with SORTBYVOL. Specifying this parameter sorts the tape pull list by data set name.

SORTBYVOL
This parameter is mutually exclusive with SORTBYNAME. Specifying this parameter sorts the tape pull list by volume.

VOLUMEONLY
This causes the tape pull report to display volume numbers only.

Examples

In this example, only one job name is included in the report. If found, the volumes assigned to that job name will be displayed sorted by volume.

```
//STEPNAME EXEC BKMBPROC
//BKMBATCH.SYSIN DD *
  REPORT CBTI-TAPEPULL JOB(jobname1)
           SORTBYVOL
```

In this example, volumes from any job name matching jobname1 or jobname2* from generation 0 between today and 2001/2/29 is displayed. If a matching job name is found, the list is sorted by data set name.

```
//STEPNAME EXEC BKMBPROC
//BKMBATCH.SYSIN DD *
  REPORT CBTI-TAPEPULL JOB(jobname1 jobname2*)
                GEN(0)            -
               DATE(*,2001/02/29) -
         SORTBYNAME               -
```

BACKUP LIST-CBTI (Backup List by Data Set Name, Date report)

Specifies BACKUP LIST-CBTI to submit a report request for the Backup List by Data Set Name, Date report in batch.

Required parameters

BACKUP LIST-
Create the Backup Lists report type.

DS ***dsn***
Specifies the names or name masks of the data sets to be listed in the Backup List report.

Optional parameters

DATE

fromdate fromdatemask
Specifies the starting date or mask of the data sets or events to be listed in the Backup List report.

todate todatemask
Specifies the ending date or mask of the data sets or events to be listed in the data set list report. For relative date masking use * for today, and *-nnn for today minus a number of days, up to 999. Relative date masking is permitted. For example, you can specify an asterisk (*) to indicate the current date, or you could specify *-5 for five days before the current date (up to *-999).

JOB ***jobname***
Specifies any number of job names or masks of job names to be listed in the Backup List report.

TYPE
Specifies a utility type to search for data sets in:

A
Data sets backed up by ADARUN.

C
Data sets backed up by ACM or CR+.

D
Data sets backed up DSS.

F
Data sets backed up by FDR.

G
Data sets backed up by IEBGENER, ICEGENER, IEBCOPY, or SYNCGENR.

I
Data sets backed up by IDCAMS.

M
Data sets backed up by CFCAMS.

O
Data sets backed up by ICETOOL.

R
Data sets backed up by Fast Replication Services.

S
Data sets backed up by ICEMAN, SORT, or SYNCSORT.

T
Data sets backed up by TAPECOPY.

U
Data sets backed up by User Defined Backup programs.

V
Data sets backed up by CA-FAVER.

W
Data sets backed up by CA-Disk.

Z
Data sets backed up by VSAM Assist.

BKMBPROC syntax

This information describes BKMBPROC syntax.

BUILD-BIFDSN and REFRESH-Critical Backup Tracking and Inventory

Any time you add new job names to Critical Backup Tracking and Inventory through ISPF, you must rebuild the contents of the batch intercept file with the current information stored in the Advanced Backup and Recovery for z/OS Inventory Data Set and then refresh the started task so that a new copy of the intercept filters can be loaded into storage.

Purpose

The BUILD-BIFDSN and REFRESH-CBTI parameters enable you to rebuild the contents of the batch intercept file with the current information stored in the Advanced Backup and Recovery for z/OS Inventory Data Set and then refresh the started task. These statements are executed on all participating systems. The original filters are valid until the batch job has completed.

Example

```
//STEPNAME EXEC BKMBPROC
//BKMBATCH.SYSIN DD *
  BUILD-BIFDSN
  REFRESH-CBTI
```

LOADIDS

Specify LOADIDS to load, in batch, a previously unloaded copy of theAdvanced Backup and Recovery for z/OS Inventory Data Set .

Purpose

LOADIDS loads a previously unloaded copy of the Advanced Backup and Recovery for z/OS Inventory Data Set. The default behavior is to load all data previously extracted from the Advanced Backup and Recovery for z/OS Inventory Data Set.

Note: "All data" refers to all aggregate records from ABARS Manager and all job records from Critical Backup Tracking and Inventory. Optional parameters can be used to selectively load previously extracted Advanced Backup and Recovery for z/OS Inventory Data Set data.

CAUTION: If selected loads are required, for performance reasons it is recommended that the selection be made during the UNLOADIDS rather than unloading everything and then selectively loading.

`LOADIDSINFILE(`*dsn*`)`

Required parameters

LOADIDS
(Used with ABARS Manager and Critical Backup Tracking and Inventory)

If this parameter is specified without the optional parameters AG (available with ABARS Manager) or JOB, all records contained in the Advanced Backup and Recovery for z/OS Inventory Data Set for both ABARS Manager and Critical Backup Tracking and Inventory will be loaded.

To load ABARS Manager or Critical Backup Tracking and Inventory records separately or selectively, the AG or JOB parameters are required.

Note: AG is an ABARS Manager only parameter.

INFILE(*dsn*)
Specifies the data set name (*dsn*) of the file containing previously unloaded Advanced Backup and Recovery for z/OS Inventory Data Set records. Do not specify quotes around the data set name.

Optional parameters

JOB(*|jobname1 |jobmask1*)
(Critical Backup Tracking and Inventory only) Specifies jobs to be loaded. This parameter is required if you want to load Critical Backup Tracking and Inventory data either selectively (that is, load only the jobs you are interested in) or separately from ABARS Manager. Specifying an asterisk (*) causes all jobs to be loaded.

The relationship between AG and JOB is:

- If both AG and JOB parameters are omitted, all data is loaded.
- If only one parameter, AG, or JOB is specified, only the data specified will be loaded.

EXCLJOB(*jobname1|jobmask1*)
(Critical Backup Tracking and Inventory only) This parameter is only valid with the JOB parameter. EXCLJOB specifies jobs to exclude from the load. You can specify job names or job name masks.

GENS(*nn*)
Specifies which generations are to be loaded. *nn* specifies the number of generations to load. Specifying an asterisk (*) causes all generations to be loaded. If specified, the generations are loaded in the order from newest to oldest. Omitting the GENS parameter automatically loads all generations. Only one GENS parameter per LOADIDS statement is allowed. This parameter is shared by ABARS Manager and Critical Backup Tracking and Inventory.

REPLACE
Replaces current entries. Only one REPLACE parameter per LOADIDS is allowed. This parameter is shared by ABARS Manager and Critical Backup Tracking and Inventory.

Examples

Consider an example in which all data previously extracted from the Advanced Backup and Recovery for z/OS Inventory Data Set is loaded from the specified data set into the Advanced Backup and Recovery for z/OS Inventory Data Set.

```
//STEPNAME  EXEC  BKMBPROC
//BKMBATCH.SYSIN  DD *
    LOADIDS INFILE(dsn)
```

Now consider an example in which one generation of all jobs, except for those starting with PA*, are loaded from the data set into the Advanced Backup and Recovery for z/OS Inventory Data Set.

```
//STEPNAME  EXEC  BKMBPROC
//BKMBATCH.SYSIN  DD *
    LOADIDS INFILE(PROD.BKM.INVDB.UNLOAD.IDS) -
           JOB(*) EXCLJOB(PA*) GENS(1)
```

In this example, three generation of all aggregates, and three generations of Critical Backup Tracking and Inventory tracked jobs beginning with PAY*, except for PAY1, are loaded from the data sets into the Advanced Backup and Recovery for z/OS Inventory Data Set.

```
//STEPNAME  EXEC  BKMBPROC
//BKMBATCH.SYSIN  DD *
    LOADIDS INFILE(PROD.BKM.INVDB.UNLOAD.IDS) -
           AG(*) JOB(PAY*) EXCLJOB(PAY1) GENS(3)
```

UNLOADIDS

Specify UNLOADIDS to perform an unload, in batch of the data from the Advanced Backup and Recovery for z/OS Inventory Data Set.

Purpose

UNLOADIDS unloads data from the Advanced Backup and Recovery for z/OS Inventory Data Set.

Note: The default behavior is to unload all data from the Advanced Backup and Recovery for z/OS Inventory Data Set. "All data" refers to all aggregate records from ABARS Manager and all job records from Critical Backup Tracking and Inventory. Optional parameters can be used to selectively unload data.

CAUTION: If selected loads are required, for performance reasons it is recommended that the selection be made during the UNLOADIDS rather than unloading everything and then selectively loading.

Using UNLOADIDS to remove ABARS Manager entries incurred during a trial

If you want to remove ABARS Manager entries from the Advanced Backup and Recovery for z/OS Inventory Data Set you can do so by following these steps:

1. Specify UNLOADIDS to extract the Critical Backup Tracking and Inventory entries.
2. Delete the Advanced Backup and Recovery for z/OS Inventory Data Set.
3. Define a new Advanced Backup and Recovery for z/OS Inventory Data Set.
4. Specify LOADIDS to load the previously unloaded Critical Backup Tracking and Inventory entries.

UNLOADIDS

Required parameters

UNLOADIDS

(ABARS Manager and Critical Backup Tracking and Inventory.)

Specify UNLOADIDS to unload all ABARS Manager aggregates and all Critical Backup Tracking and Inventory jobs.

If this parameter is specified without the optional parameters AG (available with ABARS Manager) or JOB, all records contained in the Advanced Backup and Recovery for z/OS Inventory Data Set for both ABARS Manager and Critical Backup Tracking and Inventory will be unloaded.

To unload ABARS Manager or Critical Backup Tracking and Inventory records separately or selectively, the AG or JOB parameters are required.

Note: AG is an ABARS Manager only parameter.

Optional parameters

JOB(*|*jobname1* |*jobmask1*)
Critical Backup Tracking and Inventory only. Specifies jobs to be unloaded. The JOB parameter is required if you want to unload Critical Backup Tracking and Inventory data either selectively (that is, unload only the jobs you are interested in) or separately from ABARS Manager. The AG parameter is an ABARS Manager ONLY parameter. Specifying an asterisk (*) causes all jobs to be unloaded.

The relationship between AG and JOB is:

- If both AG and JOB parameters are omitted, all data is loaded.
- If only one parameter, AG, or JOB is specified, only the data specified will be loaded.

EXCLJOB(*jobname1* |*jobmask1*)
(Critical Backup Tracking and Inventory only.) EXCLJOB is valid only with the JOB parameter. EXCLJOB specifies jobs to exclude from the unload. You can specify job names or job masks.

GENS(*nn*)
Specifies which generations are to be unloaded. *nn* specifies the number of generations to unload. Specifying an asterisk (*) causes all generations to be unloaded. If specified, the generations are unloaded in the order newest to oldest. Omitting the GENS parameter will automatically unload all generations. Only one GENS parameter per UNLOADIDS statement is allowed.

OUTFILE(*dsn*)
Specifies the output data set name *dsn* to unload Advanced Backup and Recovery for z/OS Inventory Data Set records. If the data set name specified in the OUTFILE parameter does not exist, it will be dynamically allocated using the following specifications: `SPACE=(CYL,(1,1)),UNIT=SYSALLDA, RECFM=VBS,LRECL=32756,BLKSIZE=32760`

You can change the default allocation values by modifying the following settings in the :PROCESSING_OPTIONS section of the BKMINI PARMLIB member:

```
UNLOADIDS_SPACE = 'CYLINDERS SPACE(1 2)' UNLOADIDS_UNIT = SYSALLDA
```

If OUTFILE parameter is not specified, all data in the Advanced Backup and Recovery for z/OS Inventory Data Set will be unloaded, but only a report will be created.

This parameter is shared by ABARS Manager and Critical Backup Tracking and Inventory.

Note: Quotes around the data set name are not permitted. Only one OUTFILE parameter per UNLOADIDS statement is allowed.

Examples

In this example, all data will be unloaded from the Advanced Backup and Recovery for z/OS Inventory Data Set into the specified data set. This includes both ABARS Manager and Critical Backup Tracking and Inventory records.

```
//STEPNAME  EXEC  BKMBPROC
//BKMBATCH.SYSIN  DD *
     UNLOADIDS OUTFILE(dsn)
```

In this example, one generation of all jobs, except for those starting with PA*, will be unloaded from the data set into the Advanced Backup and Recovery for z/OS Inventory Data Set.

```
//STEPNAME  EXEC  BKMBPROC
//BKMBATCH.SYSIN  DD *
    UNLOADIDS OUTFILE(PROD.BKM.INVDB.UNLOAD.IDS) -
           JOB(PAY*) EXCLJOB(PAY1) GENS(3)
```

Using ABARS Manager in batch mode

You can use ABARS Manager batch commands to perform **ABACKUP**, **ARECOVER**, native ABARS **ARECOVER**, aggregate management, SDSL parameter conversion, loading the activity log, loading and unloading the Inventory Data Set (IDS), and reporting.

Viewing Advanced Backup and Recovery for z/OS messages

This topic tells you how to look up Advanced Backup and Recovery for z/OS messages.

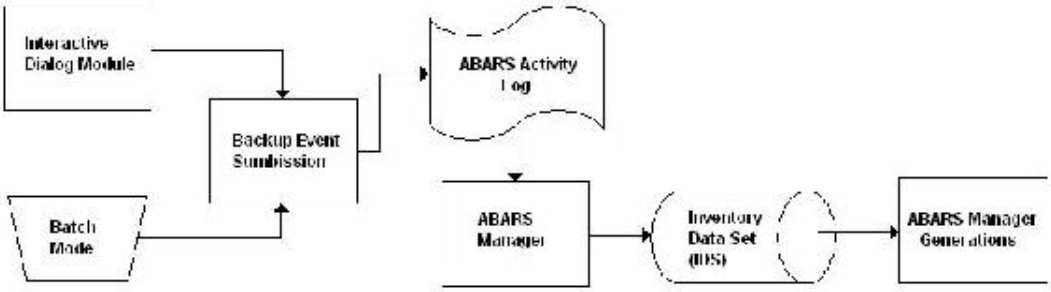
You can look up message text, explanations, and recommended user responses three ways:

- Look up the message by number in the IBM Tivoli Advanced Backup and Recovery for z/OS: Messages Reference.
- Browse the appropriate member in the Advanced Backup and Recovery for z/OS messages library
- Select Menu from any Advanced Backup and Recovery for z/OS panel, then select the Messages option to browse the messages library.

Using the ABACKUP command in batch mode

This topic explains the ABACKUP batch process and provides detailed information about running ABACKUP in batch mode.

The following figure shows the sequence of events for ABACKUP batch processing.



ABARS Manager interfaces with DFSMShsm to initiate the ABARS ABACKUP and locate and process the resulting activity log. Information from the activity log is used to update the Inventory Data Set (IDS) for each aggregate's ABACKUP

Submitting the ABACKUP command

This topic provides information about submitting **ABACKUP** batch commands.

Use the **ABACKUP** command to submit an ABARS aggregate backup in batch. You can generate and submit **ABACKUP** batch commands interactively using the online dialog panels, or you can provide the command syntax as SYSIN to the BKMBPROC member of the Advanced Backup and Recovery for z/OS JCL library.

ABACKUP command syntax

This topic shows the **ABACKUP** command syntax, provides parameter descriptions, and shows examples of its usage.

The **ABACKUP** command runs in batch mode to create a backup copy of one or more aggregates.

ABACKUP AG(*aggname1* | *aggnamemask1* | *logaggname1 ...aggnamen* | *aggnamemaskn*) EXECUTE | VERIFY | VEREXE [CCERROR(ABEND | CONTINUE | WTOR)]

[CCOFF(ALL | D | I)]

[FILTEROUTPUTDATASET(dsn)]

[INCREMENTAL]

[LIST(SKIPPED)]

[MOVE]

[NOTIFY(userid)]

[OPTIMIZE(1 | 2 | 3 | 4)]

[PROCESSONLY(LEVEL0 | MIGRATIONLEVEL1 | MIGRATIONLEVEL2 | USERTAPE)]

[RESETCB]

[RECALL-MIGRATED-VSAM(Y | N)]

[RECALL-FAILED(E | W)]

[SKIP(PPRC | XRC)]

[STACK | NOSTACK]

[TERMMSGS(NO | YES)]

[UNIT(3400-3 | 3400-4 | 3400-5 | 3400-6 | 3400-9 | 3480 | 3480X | 3490 | 3590-1)]

Required parameters

The following parameters are required:

AG (*aggname1 | aggnamemask1 | logaggname1...aggnamen | aggnamemaskn*)
Use parentheses to specify multiple blank delimited aggregate names, aggregate name masks, or logical aggregate names. You can process a maximum of 256 aggregates with one **ABACKUP** command. You cannot use a mask to specify logical aggregate names.

For concurrent execution, there is no limit on the number of aggregate names included in the **ABACKUP** command.

There is no default value for this parameter.

EXECUTE | VERIFY | VEREXE
One (and only one) of these options must be specified:

- EXECUTE schedules an **ABACKUP**. No VERIFY step is performed.
- VERIFY submits a VERIFY without performing an EXECUTE step.
- VEREXE submits a VERIFY step first; if VERIFY succeeds, an EXECUTE is submitted.

There is no default value for this parameter.

Optional parameters

The following parameters are optional:

CCERROR(ABEND | CONTINUE | WTOR)
Use this parameter to specify what action you want ABARS Manager to take in the event that a concurrent copy fails.

Note: If backups are running concurrently, any unaffected backups will continue normally. The ABEND occurs only after all concurrent processing has completed.

- ABEND cancels backups for this aggregate and causes the job to terminate abnormally as soon as possible.
- CONTINUE causes ABARS Manager to ignore the failure and continue processing.

- WTOR asks the console operator to respond with a course of action.

There is no default value for this parameter.

CCOFF(*ALL | D | I*)
Use this parameter to override the management class specifications when using concurrent copy.

- ALL disables concurrent copy when backing up the instruction data set and LEVEL0 data to the D file.
- D disables concurrent copy when backing up LEVEL0 data to the D file.
- I disables concurrent copy when backing up the instruction data set.

FILTEROUTPUTDATASET
Use this parameter to specify the name of an output data set to which the lists of data sets selected from INCLUDE/EXCLUDE, ALLOCATE/ALLOCATEEXCLUDE, ACCOMPANY/ACCOMPANYEXCLUDE filtering are written. The data set name can be 1 to 44 characters in length. If the data set specified is NOT cataloged, a new data set is created. Quotes are optional.

The short form of this parameter is FOD.

INCREMENTAL
Use this parameter to specify that this is an incremental ABARS backup.

LIST(SKIPPED)
This parameter cannot be used unless the SKIP parameter has been specified. Use the LIST parameter to specify that you want all data sets skipped during **ABACKUP** processing to be identified in the **ABACKUP** activity log.

MOVE
Use this parameter to specify what you want done with all selected data sets after a successful execution of the aggregate backup: delete them, uncatalog them, or both delete and uncatalog them.

If you specify this parameter when backing up an application, all data sets in the application are deleted after a successful backup. These application data sets are affected in the following manner:

- ACCOMPANY data sets are uncataloged.
- ALLOCATE data sets are deleted from DASD (except ICF user catalogs in the ALLOCATE list).
- DASD data sets are deleted.
- MIGRATED data sets are deleted.
- TAPE data sets are uncataloged.

NOTIFY(*userID*)
If you want a DFSMShsm messages sent to a secondary user ID, use this keyword to specify that user ID.

OPTIMIZE(1|2|3|4)
Use this parameter to override the **SETSYS ABARSOPTIMIZE** command. The OPTIMIZE parameter is passed to and used by DFSMSdss when it backs up level 0 DASD data sets. Valid values for this parameter are as follows:

- 1 causes DFSMSdss to read one track at a time.
- 2 causes DFSMSdss to read two tracks at a time.
- 3 causes DFSMSdss to read five tracks at a time. This is the default value for OPTIMIZE if you did not specify **SETSYS ABARSOPTIMIZE**.
- 4 causes DFSMSdss to read one cylinder at a time.

PROCESSONLY(LEVEL0|MIGRATIONLEVEL1|MIGRATIONLEVEL2|USERTAPE)
Use this parameter to direct DFSMShsm to process only the specified data set types. You can specify the data set types in any order and in any combination.

- LEVEL0 causes DFSMShsm to process DASD data sets.
- MIGRATIONLEVEL1 causes DFSMShsm to process ML1 data sets.
- MIGRATIONLEVEL2 causes DFSMShsm to process ML2 data sets.

- USERTAPE causes DFSMShsm to process tape data sets.

There is no default value for this parameter.

RESETCB
Use this parameter to have ADRDSSU reset the data set change bit in the VTOC after the data set has been successfully backed up. By default, the change bit is not reset. You cannot use this parameter if you submit **ABACKUP** from an ISPF panel.

Note: Generally, ABARS is used for disaster recovery while DFSMShsm Incremental backups are taken for local recovery. The RESETCB keyword resets the change bit after the data set is successfully backed up in an **ABACKUP**. DFSMShsm Incremental backup processing does not select these data sets for backup because the change bit flag was turned off.

RECALL-MIGRATED-VSAM(Y|N)
Use this parameter to specify whether you want ABARS Manager to recall VSAM data sets before submitting the **ABACKUP** to ABARS. The default value for this parameter is N.

RECALL-FAILED(E|W)
Use this parameter to specify how you want Advanced Backup and Recovery for z/OS to treat failures that occur during a recall of a VSAM data set.

- E causes processing to terminate.
- W causes processing to continue.

The default value for this parameter is E.

SKIP(PPRC|XRC|PPRC XRC)
Use this parameter to specify how you want data sets residing on PRC or XRC (or both) volumes skipped during **ABACKUP** processing.

- PPRC skips data sets that reside entirely on PPRC primary volumes.
- XRC skips data sets that reside entirely on XRC primary volumes.
- PPRC and XRC skips data sets that reside entirely on either PPRC or XRC primary volumes.

There is no default value for this parameter.

STACK|NOSTACK
Use this parameter to override the SETSYS ABARSTAPES setting.

- STACK causes DFSMShsm to stack the **ABACKUP** output files onto the fewest possible number of tape volumes during the **ABACKUP**.
- NOSTACK causes DFSMShsm to not stack the output files during the **ABACKUP**.

STACK is the default situation.

TERMMSGS(NO|YES)
Use this parameter to specify how you want HSM messages handled.

- NO causes HSM messages to appear in the ABARS Manager processing log, but they are not sent to the user's terminal.
- YES sends HSM messages to the user's terminal.

The default value for this parameter is YES.

UNIT(3400-3|3400-4|3400-5|3400-6|3400-9|3480|3480X|3490|3590-1)
Use this parameter to specify the type of output tape unit to which you want the aggregate backup data written. If you do not specify a unit type for this parameter, the default becomes the **UNIT** value specified by the **SETSYS ABARSUNITNAME** command.

Examples

The following examples show how to initiate **ABACKUP** aggregate events in batch mode using procedure BKMBPROC .

Consider an example in which only one aggregate is backed up. The condition code of this backup can be checked in the next step using normal step condition code checking. The VEREXE parameter checks for an **ABACKUP** VERIFY failure. If so, then the **ABACKUP** EXECUTE is aborted.

```
//STEPNAME EXEC BKMBPROC
//BKMBATCH.SYSIN DD *
  ABACKUP AG(aggname1) VEREXE
```

Another method is to code the same JCL but have more than one aggregate name in the **ABACKUP** command.

```
//STEPNAME EXEC BKMBPROC
//BKMBATCH.SYSIN DD *
  ABACKUP AG(aggname1 aggname2) EXE -
          NOTIFY(UserXYZ)
```

In this case, multiple aggregate backups are executed. Each is submitted as a separate DFSMShsm subtask and controlled by the submitting JCL. The highest return code produced from any of the executes causes the job to end with that return code.

The output for these **ABACKUP**s can be found under the jobname submitted. Individual SYSOUTs are created for each aggregate, identified by the aggregate name.

In this example, all ABARS Manager messages are sent to 'UserXYZ', if that user is currently logged on to the system.

In the following command examples, VERIFY is used in the constructed **ABACKUP** command given to HSM.

```
//STEPNAM1 EXEC BKMBPROC
//BKMBATCH.SYSIN DD *
  ABACKUP AG(aggname1) VER

//STEPNAM2 EXEC BKMBPROC,COND=(0,LT,STEPNAM1)
//BKMBATCH.SYSIN DD *
  ABACKUP AG(aggname2) VER
```

The example above single threads the **ABACKUP** requests, depending on the condition code returned from the previous step. This is particularly convenient when you need multiple aggregate **ABACKUP** commands in the same job and require condition code checking, The condition code of each of these **ABACKUP** VERIFYs is written to the job's SYSOUT.

About concurrent copy

DFSMShsm issues messages to the console regarding the status of the DFSMSdss Concurrent Copy operation. Message ARC6402I is issued when the Concurrent Copy session initialization is complete. Message ARC6402I indicates the number of data sets for which Concurrent Copy initialization was successful or unsuccessful.

If all data sets to be backed up by DFSMSdss are eligible for Concurrent Copy, message ARC6402I is issued as soon as Concurrent Copy initialization is complete. At this point, for sites that do not have AUTO operations, a REXX EXEC can be used to submit a job to act as a trigger for the scheduler.

When the **CCTRIGGER** parameter in the BKMINI member of the product parameter library is set to Y, and the management class causes Concurrent Copy, ABARS Manager intercepts the messages during DFSMSdss processing and automatically issues the REXX EXEC at 'commit'.

A sample REXX EXEC is provided in member BKMCCTRG of the product parameter library. You can modify this any way you like. It is just a sample. If you want to create separate job names for separate aggregates, you must add REXX logic. Both the aggregate name and a status parameter are passed to the REXX EXEC. The BKMABARS procedure, which HSM uses to launch an ABARS **ABACKUP**, has a SYSEXEC DD from which the REXX can be launched.

If any data set is not eligible for Concurrent Copy, message ARC6402I is issued only when the entire DFSMSdss backup is physically complete, which could be several hours later. ABARS Manager provides

several **ABACKUP** parameters to enhance this process. If a failure occurs during Concurrent Copy processing and these **ABACKUP** commands have been provided, the following occurs:

- Continue – Concurrent Copy failure is to be ignored and continue processing.
- WTOR – A message will inform the console operator of the problem and the appropriate course of action.
- ABEND – The task is to stop the backup for this aggregate and the job is to ABEND as soon as possible.

Because the use of Concurrent Copy for an aggregate backup is controlled by the SMS management class, if you need to change the Concurrent Copy options for an aggregate, you must change the management class definition for the aggregate and reactivate the updated SMS configuration, unless you are using ABARS Manager. ABARS Manager provides additional flexibility in the use of Concurrent Copy. You can temporarily and dynamically disable the use of Concurrent Copy for an entire aggregate or for only the backup of the activity log and instruction data set.

Overriding SYSOUT to a data set

This topic provides a procedure for overriding SYSOUT to a data set.

To override SYSOUT to a data set, code the following, where BKMBATCH.AGGNAME is the PROCSTEP, followed by the actual aggregate name and the DCB information shown must be coded.

```
//BKMBATCH.AGGNAME DD DSN=PROD.AGGNAME.OUT,DISP=(,CATLG),
// DCB=(RECFM=FB,LRECL=134,BLKSIZE=0),
// SPACE=(CYL,(1,1))
```

Using the ARECOVER command

This topic shows and describes the **ARECOVER** batch process.

The following figure shows the **ARECOVER** batch process:

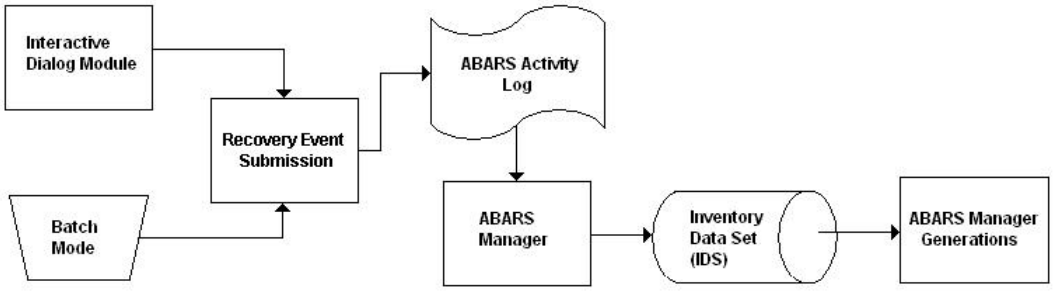


Figure 9. ARECOVER batch process

ABARS Manager interfaces with DFSMShsm to initiate the ABARS **ARECOVER** and to locate and process the resulting activity log. Information from the activity log is used to update the Inventory Data Set (IDS) for each aggregate's **ARECOVER**.

Submitting the ARECOVER command

Use the **ARECOVER** command to submit an ABARS aggregate recovery in batch. You can generate and submit **ARECOVER** commands interactively using the online dialog panels, or you can provide the command syntax as SYSIN to the BKMBPROC member of the Advanced Backup and Recovery for z/OS JCL library.

You cannot run **ABACKUP** and **ARECOVER** simultaneously against an aggregate.

ABARS Manager input parameters can pass either single or multiple **ARECOVER** commands to ABARS.

- When multiple **ARECOVER** commands are submitted in the same batch job, they are single threaded.
- When multiple aggregates are submitted in the same **ARECOVER** command, they are executed in parallel and controlled by the submitting JCL.

ARECOVER command syntax

This topic shows the **ARECOVER** command syntax, provides parameter descriptions, and shows examples of its usage.

The **ARECOVER** command runs in batch mode to restore lost files and data from backup copies.

ARECOVER AG(*aggname1* | *aggnamemask1* | *logaggname1 ...aggnamen* | *aggnamemaskn*) EXECUTE | VEREXE | VERIFY

[ACTIVITY]

[AGGENQ (NO | YES)]

[COPYSET(nn)]

[DATASETCONFLICT(BYPASS | RENAMESOURCE(level) | RENAMETARGET(level) REPLACE |)]

[EXCLUDE(dsn1 | dsnmask1 ...dsnn | dsnmaskn)]

[FINALRENAMEACTION(EXECUTE | FILE)]

[FULLRENAMEMASKS([(old_mask1, new_mask1)] ... [(old_maskn, new_maskn)])]

[GDGREPLACE]

[GDGGEN0ONLY]

[GEN(g) | GEN(g, ig) | GEN(g, ig, ONLY)]

[INCLUDE(dsn1 | dsnmask1 ...dsnn | dsnmaskn)]

[INSTRUCTION]

[MENTITY(modeldsn)]

[MIGRATEDDATA(ML1 | ML2 | SOURCELEVEL)] [ML1 | ML2 | PRIMARY | TAPE]

[NOBACKUPMIGRATED]

[NOTIFY(userid)]

[OUTPUTRENAMEDSN(&HSMUID.&HOSTID.&AGGNAME.&DATE.&TIME.&USERID | dsn)]

[PERCENTUTILIZED(nnn)]

[PREVENT_BACKLEVEL(N | Y)]

[RECOVERNEWNAMEALL(level)]

[RECOVERNEWNAMELEVEL([(olevel1, nlevel1)] ...[(olevel30, nlevel30)])]

[RESTART]

[STACK | NOSTACK]

[TARGETUNIT(3400-3 | 3400-4 | 3400-5 | 3400-6 | 3400-9 | 3480 | 3480X | 3490 | 3590-1)]

[TERMMSGS(YES | NO)]

[TGTGDS(ACTIVE | DEFERRED | ROLLEDOFF | SOURCE | NONE)]

[UNIT(3400-3 | 3400-4 | 3400-5 | 3400-6 | 3400-9 | 3480 | 3480X | 3490 | 3590-1)]

[VOLCOUNT(ANY)]

[XMIT]

Required parameters

The following parameters are required:

AG (*aggname1 | aggnamemask1 | logaggname1...aggnamen | aggnamemaskn*)
Use parentheses to specify multiple blank delimited aggregate names, aggregate name masks, or logical aggregate names. A maximum of 256 aggregates can be processed with one **ARECOVER** command. You cannot use a mask to specify logical aggregate names.

EXECUTE | VEREXE | VERIFY | VEREXE|VERIFY
One (and only one) of these options must be specified:

- EXECUTE schedules an ABACKUP. No VERIFY step is performed.
- VERIFY submits a VERIFY without performing an EXECUTE step.
- VEREXE submits a VERIFY step first; if VERIFY succeeds, an EXECUTE is submitted.

There is no default value for this parameter.

Optional parameters

The following are optional parameters:

ACTIVITY
Specify the recovery of the activity log data set, if one exists on the output tape for the specified aggregate.

AGGENQ(YES|NO)
Indicate how you want ABARS Manager to perform serialization on the aggregate:

- Specify NO if you do not want ABARS Manager to perform serialization on the aggregate name so a data set could be restored from an aggregate while a backup of the same aggregate is running.
- Specify YES to prevent the restoration of one or more data sets from an aggregate group while a backup of the same aggregate is executing.

The default value for this parameter is YES.

COPYSET(*nn*)
Use this parameter to specify which copyset to recover, where *nn* is an integer in the range 1-15, inclusive. There is no default value for this parameter.

DATASETCONFLICT(BYPASS | RENAMESOURCE(*level*)| RENAMETARGET(*level*) REPLACE |)
Use this parameter to specify how you want data set naming conflicts resolved.

- BYPASS skips data sets being recovered, if a data set already exists (is cataloged) with the same name.
- RENAMESOURCE renames data sets being recovered, assigning them the high level qualifier specified by *level*.
- RENAMETARGET renames the existing data sets at the recovery site, assigning them the high level qualifier specified by *level*. The data sets being recovered retain their original names.
- REPLACE deletes the existing (cataloged) data set and replaces it with the data set being recovered.

There is no default value for this parameter.

EXCLUDE(*dsn1* | *dsnmask1 ...dsnn* | *dsnmaskn*)
Use this parameter to specify a list of data set names or masks that identify data sets you want excluded from the recovery. You can use this parameter in conjunction with the INCLUDE, PRIMARY, ML1, ML2 and TAPE parameters.

FINALRENAMEACTION(EXECUTE | FILE)
Use this parameter to specify how you want ABARS Manager to handle renames:

- Specify EXECUTE if you want ABARS Manager to create a file containing ALTER control statements and dynamically invoke IDCAMS to perform the renames. This can be set as the installation default in parmlib member BKMINI.
- Specify FILE if you want ABARS Manager to create a file containing ALTER control statements that you can view and execute. The installation default can be changed in parmlib member BKMINI

The default value for this parameter is FILE.

FULLRENAMEMASKS([(*old_mask1, new_mask1*)] ... [(*old_maskn, new_maskn*)])
Use this parameter to specify pairs of old name and new name masks. Separate each pair from the others with a comma or a blank space.

Example:

```
FULLRENAMEMASK(                                 +
  EDP9.LAB.**,EDP9.LAB.+NOBACKUP.**             +
  EDP9.TEST*.AIX*.**,EDP9.+LAB.+NOBACKUP.**     +
  EDP9.TEST*.**,EDP9.+LAB.+NOBACKUP.**)
```

The rules for the *oldmask* are identical to those that can be used for the **INCLUDE** parameter in the DFSMSdss DUMP and RESTORE control statements:

- A single asterisk indicates any single qualifier or any group of characters within a qualifier.
- Only one double asterisk may be used in the mask and indicates zero or more qualifiers.
- A percent sign (%) indicates any single character.

The rules for the *newmask* are the same as *oldmask* except partial qualifiers (such as 'ab*') that are allowed in the *oldmask* are not allowed in *newmask*.

GDGREPLACE(Y | N)
Specify Y if you want the GDG base catalog entries to be updated without replacing existing data sets — in a single restore.

Y functions only if no ABARS conflict resolution is specified. In other words, GDG data sets (with no ABARS conflict resolution specified) are treated the same as non-GDG data sets. This option does not allow replacement of existing GDG data sets.

In native ABARS, DATASETCONFLICT(REPLACE) must be specified to force GDG data sets to restore, even though they might not exist in the target catalog. This is true because, for each GDG data set to be restored, ABARS treats the need to update the GDG base catalog entry as a conflict. This might be undesirable if you do not want existing data sets to be replaced.

Using the ABARS CR exit, the ABARS Manager GDGREPLACE option allows GDG base catalog entries to be updated, but considers existing data sets conflicts.

You might find it helpful to know that active GDG data sets (non rolled-off) do not have separate catalog entries (as an ISPF 3.4 display might lead you to believe). Active GDG data sets for a given base are all represented in a single catalog entry, therefore each data set in a GDG that is restored causes the base entry to be updated.

GDGGEN0ONLY(Y | N)
Specify Y if you want the GDG base catalog entries to be updated without replacing existing data sets — in a single restore.

Restores only the most current generation (generation 0) of any generation data set found in the ABACKUP.

Note: For Incremental ABARS, be aware that when a base and its associated incremental backups are restored, only the most current generation (generation 0) is restored *regardless* of how many generations exist in the combined base and associated incremental backups.

GEN[(g) | (g, ig) | (g, ig, only)]
Specify which generation you want recovered.

- g–Non-incremental only. Specify the relative base generation to recover.
- g, ig–Incremental only, where *g* is the relative base generation to recover (BI) and *ig* is the relative incremental generation on the base from which to recover. All incremental generations from ig to its associated base are recovered. For example, GEN(0,0) recovers from base generation 0/incremental generation 0 and incremental generations to the associated base.
- g,ig,only–Incremental only, where *g* is the relative base generation to recover (BI) and *ig* is the relative incremental generation on the base from which to recover. Only the specified *g,ig* is recovered. For example, GEN(0,-3,ONLY) recovers base generation 0/incremental generation -3 only.

INCLUDE
Specify a list of data set names or masks to include in the recovery.

You can use this parameter in conjunction with the EXCLUDE, PRIMARY, ML1, ML2, and TAPE parameters.

INSTRUCTION
Specify that you want the instruction data set recovered, if one exists on the output tape for the specified aggregate group at the backup site.

MENTITY
Specify the data set name of the predefined RACF model. The model is used to define discrete profiles to RACF for data sets to be recovered.

MIGRATEDDATA (ML1|ML2|SOURCELEVEL)
Specify the level to which you want to recover the migrated data sets.

- ML1–Recover all migrated data sets to ML1.
- ML2–Recover all migrated data sets to ML2.
- SOURCELEVEL–Recover all migrated data sets to their original migration level.

[ML1|ML2|(PRIMARY|PRI)|TAPE]
Specify which data set types you want DFSMShsm to process. You can specify data set types in any order and in any combination of types.

- ML1–Recover ML1 data sets.
- ML2–Recover ML2 data sets.
- PRIMARY–Recover DASD data sets.
- TAPE–Recover tape data sets.

NOBACKUPMIGRATED
Turn off automatic backup for migrated data sets.

NOTIFY
Specify a secondary user ID to which you want DFSMShsm messages sent.

OUTPUTRENAMEDSN(&HSMUID.&HOSTID.&AGGNAME.&DATE.&TIME.&USERID| *dsn*)
Specify the name of the ALTER output data set that will contain the IDCAMS ALTER (rename) control statements. The default is &HSMUID.&HOSTID.&AGGNAME.&DATE.&TIME.&USERID. You can specify a *dsn* value that is fully qualified or contains any of the following symbolic values:

- &HSMUID–The authorized HSM-Userid value (see SETSYS options).
- &HOSTID–The HSM Host-ID (see ARCCMDXX in HSM PARMLIB).
- &AGGNAME–The name of the aggregate being recovered.
- &DATE–The current date in D*yyyyddd* format.
- &TIME–The current time in format T*hhmmss* format.
- &USERID–The user ID under which the **ARECOVER** will be submitted.

PERCENTUTILIZED
Specify an integer between 1 and 100 to allow DFSMSdss to fill the L0 volumes in the associated ARPOOL up to 100% (100) for this aggregate.

PREVENT_BACKLEVEL (N|Y)
Specify whether you want to prevent a data set from being back-leveled during recovery (assuming PREVENT_BACKLEVEL has been set up.

- N–Do not use the PREVENT_BACKLEVEL parameter during recovery.
- Y–Use the PREVENT_BACKLEVEL parameter during recovery.

RECOVERNEWNAMEALL
Specify the high level qualifier to which you want all the data sets renamed.

RECOVERNEWNAMELEVEL (*olevel*,*nlevel*)
Specify pairs of *olevel* and *nlevel* values to rename all recovered data sets that match the *olevel* qualifier to the specified *nlevel* qualifier. You can specify a maximum of *olevel*, *nlevel* pairs.

- *olevel*–The original high level qualifier.
- *nlevel*–The high level qualifier to which you want the recovered data sets renamed.

For example:

```
RECOVERNEWNAMELEVEL(DEM0AGG,EDP9          +
                    EDP,EDP9)
```

STACK |NOSTACK
Use this parameter to override the SETSYS ABARSTAPES setting.

- STACK causes DFSMShsm to stack the ABACKUP output files onto the fewest possible number of tape volumes during the **ARECOVER**.
- NOSTACK causes DFSMShsm to not stack the output files during the **ARECOVER**.

STACK is the default situation.

TARGETUNIT*unit-type*
Specify the tape unit type for receiving data sets recovered to tape devices. Valid unit-type values are 3400-3, 3400-4, 3400-5, 3400-6, 3400-9, 3480, 3480X, 3490, 3590-1.

TERMMSGS(NO|YES)
Use this parameter to specify how you want HSM messages handled.

- NO causes HSM messages to appear in the ABARS Manager processing log, but they are not sent to the user's terminal.
- YES sends HSM messages to the user's terminal.

The default value for this parameter is YES.

TGTDS (ACTIVE|DEFERRED|ROLLEDOFF|SOURCE)
Specify an override option for the **SETSYS ARECOVERTGTDS** command.

- ACTIVE–Assign the ACTIVE status to the target data set.
- DEFERRED–Assign the DEFERRED status to the target data set.
- ROLLEDOFF–Assign the ROLLEDOFF status to the target data set.
- SOURCE–Assign the target data set the same status as that of the source data set.

UNIT *unit-type*
Specify the tape unit type for receiving data sets recovered to tape devices. Valid unit-type values are 3400-3, 3400-4, 3400-5, 3400-6, 3400-9, 3480, 3480X, 3490, 3590-1.

VOLCOUNT (ANY)
Specify an override to the SETSYS ABARSVOLCOUNT command, and allow DFSMSdss to allocate as many volumes as required, up to a maximum of 59 for all L0 data sets

XMIT
Specify this keyword to cause **ARECOVER** to search the catalog for the control and data files, and, if found, use the unit and volser from the catalog when allocating these data sets.

FullRename restrictions

ABARS Manager Data Set Rename Extensions are subject to limitations, some of which are based on DSS restrictions and some that are due to restrictions in native ABARS processing.

These are the data set rename limitations:

- ICF catalogs cannot be renamed.
- Data sets in the ACCOMPANY list cannot be renamed (because they are tape).
- Tape data sets specified in the INCLUDE list cannot be renamed.

- Migrated data sets are recalled to rename them.
- Migrated VSAM data sets cannot be renamed because they are known to DFSMShsm as non-VSAM by the cluster name. The data and index names are unknown.
- Each qualifier in the filter fully replaces a qualifier in the original DSN.
- The FullRename function is accomplished after the ARECOVER has completed by using IDCAMS ALTER statements. If you try to rename a data set to one that is already cataloged, the rename will fail.
- Because ABARS Manager data set rename processing renames data sets only after they have been ARECOVERed, the RECOVERNEWNAMEALL or RECOVERNEWNAMELEVEL parameters may be necessary in order to restore the data sets using a temporary, or intermediate, high level qualifier.
- No check is made to see that the same data set name is not created twice by the renaming criteria the user specifies. If two or more data sets are to be renamed to the same output name, only the first renaming will be successful.
- The output mask cannot have a partially qualified node. For example, PROD.A*.** is not allowed.

User catalog and SMS restrictions

If you are renaming a previously recovered data set by changing its high level qualifier value, the final renamed data set could be re-cataloged in a different user catalog other than the original (old) data set name. In this case, the rename function will be successful, but the user data set will not be accessible from the original catalog. In addition, certain installation SMS restrictions may prevent a complete renaming of data set entries. With VSAM, the rename fails if the **Rename to DSN/Mask** will be in a catalog other than the catalog of the **Original DSN/Mask**.

Tips for using FullRename

You might find the following advice helpful:

- The FullRename function is accomplished after the ARECOVER has completed by using IDCAMS ALTER statements. Therefore, the data sets need to be restored first. You can use either ARECOVER with the REPLACE command, or ABARS RECOVERNEWNAMEALL or RECOVERNEWNAMELEVEL except for VSAM files.
- Ensure the target data sets do not exist on the system unless they are GDG data sets and the RENAME_GDG_DELETE or RENAME_GDG_SCRUB parameters are set to Y in the product PARMLIB member BKMINI.
- Read all the rules and ensure all data is appropriately planned for to ensure a successful FullRename.

Rules for the Original DSN/Mask filter

The Original DSN/Mask filter mask is used to select the data sets against which to apply the Rename to DSN/Mask filter mask. Valid filter characters for the Original DSN/Mask filter mask are as follows:

*	A single asterisk represents exactly one DSN qualifier of any value. Example: *. or .*. or .* . A single asterisk combined with valid DSN characters or with % means '0 to nn characters of any value.'
**	A double asterisk represents 0 to nn DSN qualifiers of any value. Example: **. or .**. or .** A double asterisk can be used *once* in a mask; for example, **.abcd.** would not be valid.
%	A percent sign represents one nonblank character.
!	An exclamation point represents one national character, such as @ # $.
<	A less-than sign represents one nonnumeric character, national symbols included.
>	A greater-than sign represents one numeric character.

Note: Advanced Backup and Recovery for z/OS codes uses the USA EBCDIC characters.

For example, a filter of **.PAYROLL*.%%%%%23*.*.DATA would match a data set named TLQ050.PAYROLL.CYCLE23.YEAREND.DATA.

Rules for the Rename to DSN/Mask filter

The **Rename to DSN/Mask** (new name) filter mask is used to rename the data sets selected by the **Original DSN/Mask** (old name) filter mask. The allowable filter characters are as follows:

*	A single asterisk represents exactly one DSN qualifier of any value. You cannot use a single asterisk to represent a partial qualifier in a new name mask. For example, aaa.*bb.** is not a valid mask.
**	A double asterisk represents 0 to *nn* DSN qualifiers of any value. A double asterisk can be used *once* in a mask; for example, **.abcd.** would not be valid.
%	A percent sign represents one nonblank character.
+cccccc	A plus sign followed by 1 to 8 characters means 'insert this new qualifier.'
-	A minus sign means 'remove this qualifier from the new name.'

For example, if the original DSN/mask is BKMI.LAB9.DEMO1.SDS.ORIG, and the rename DSN/mask is BKMI.+TEST.XX%%.-.**, the renamed data set would be BKMI.TEST.XXB9.SDS.ORIG.

Mask examples

Consider an example in which we rename a single, fully qualified data set: FULLRENAMEMASKS(AN.OLD.DATA.SET,A.NEW.DATA.SET)

```
Original DSN/Mask     ==> AN.OLD.DATA.SET
Rename to DSN/Mask    ==> A.NEW.DATA.SET
```

Results:

```
Old Data Set Name    AN.OLD.DATA.SET
New Data Set Name    A.NEW.DATA.SET
```

Now consider an example in which we rename any three-qualifier DSN with a middle level qualifier of PAY to a new name that begins with BKMI:

```
FULLRENAMEMASKS(*.PAY.*,BKMI.PAY.*)

Original DSN/Mask     ==> *.PAY.*
Rename to DSN/Mask    ==> BKMI.PAY.*
```

Results:

```
NONVSAM DS: BKM.PAY.TEST1
             >> MATCHING FILTER:  *.PAY.*
             >> NEW NAME FILTER:  BKMI.PAY.*
             >> NEW NAME       :  BKMI.PAY.TEST1
NONVSAM DS: BKM.PAY.TEST2
             >> MATCHING FILTER:  *.PAY.*
             >> NEW NAME FILTER:  BKMI.PAY.*
             >> NEW NAME       :  BKMI.PAY.TEST2
NONVSAM DS: BKM.PAY.TEST3
             >> MATCHING FILTER:  *.PAY.*
             >> NEW NAME FILTER:  BKMI.PAY.*
             >> NEW NAME       :  BKMI.PAY.TEST3
NONVSAM DS: BKM.PAY.TEST4
             >> MATCHING FILTER:  *.PAY.*
             >> NEW NAME FILTER:  BKMI.PAY.*
             >> NEW NAME       :  BKMI.PAY.TEST4
```

In this example, we rename all data sets beginning with BKMI and ending with MAY to end with JUNE:

```
FULLRENAMEMASKS(BKMI.**.MAY,**.JUNE)
```

```
Original DSN/Mask          =>  BKMI.**.MAY
Rename to DSN/Mask        =>  **.JUNE
```

Results:

```
NONVSAM DS: BKMI.QUAL1.A.MAY
             >> MATCHING FILTER:  BKMI.**.MAY
             >> NEW NAME FILTER:  **.JUNE
             >> NEW NAME       :  BKMI.QUAL1.A.JUNE
NONVSAM DS: BKMI.QUAL1.MAY
             >> MATCHING FILTER:  BKMI.**.MAY
             >> NEW NAME FILTER:  **.JUNE
             >> NEW NAME       :  BKMI.QUAL1.JUNE
NONVSAM DS: BKMI.QUAL1.QUAL2.MAY
             >> MATCHING FILTER:  BKMI.**.MAY
             >> NEW NAME FILTER:  **.JUNE
             >> NEW NAME       :  BKMI.QUAL1.QUAL2.JUNE
NONVSAM DS: BKMI.QUAL2.MAY
             >> MATCHING FILTER:  BKMI.**.MAY
             >> NEW NAME FILTER:  **.JUNE
             >> NEW NAME       :  BKMI.QUAL2.JUNE
```

In the next example, we rename any three-qualifier data set with BKM in the first three characters of the first qualifier and DATA as the low level qualifier to have IVP as the high level qualifier and DATA2 as the last qualifier:

```
FULLRENAMEMASKS(BKM*.*.DATA,IVP.*.DATA2)

Original DSN/Mask          =>  BKM*.*.DATA
Rename to DSN/Mask        =>  IVP.*.DATA2
```

Results:

```
NONVSAM DS: BKMI.GEN1.DATA
             >> MATCHING FILTER:  BKM*.*.DATA
             >> NEW NAME FILTER:  IVP.*.DATA2
             >> NEW NAME       :  IVP.GEN1.DATA2
NONVSAM DS: BKMI.GEN2.DATA
             >> MATCHING FILTER:  BKM*.*.DATA
             >> NEW NAME FILTER:  IVP.*.DATA2
             >> NEW NAME       :  IVP.GEN2.DATA2
NONVSAM DS: BKMI.GEN3.DATA
             >> MATCHING FILTER:  BKM*.*.DATA
             >> NEW NAME FILTER:  IVP.*.DATA2
             >> NEW NAME       :  IVP.GEN3.DATA2
NONVSAM DS: BKMI.GEN4.DATA
             >> MATCHING FILTER:  BKM*.*.DATA
             >> NEW NAME FILTER:  IVP.*.DATA2
             >> NEW NAME       :  IVP.GEN4.DATA2
```

This example shows the removal of qualifier ABC.DEF from the new data set names.

```
FULLRENAMEMASKS(BKMI.D.ABC.DEF.TEST,BKMI.D.-.-.TEST)

Original DSN/Mask          =>  BKMI.D.ABC.DEF.TEST
Rename to DSN/Mask        =>  BKMI.D.-.-.TEST
```

Results:

```
NONVSAM DS: BKMI.D.ABC.DEF.TEST
             >> MATCHING FILTER:  BKMI.D.*.*.TEST
             >> NEW NAME FILTER:  BKMI.D.-.-.TEST
             >> NEW NAME       :  BKMI.D.TEST
```

Now consider two examples in which we add a qualifier of NEW to each data set.

```
FULLRENAMEMASKS(BKMI.PAY.JUNE,BKMI.+NEW.PAY.**)

Original DSN/Mask          =>  BKMI.PAY.JUNE
Rename to DSN/Mask        =>  BKMI.+NEW.PAY.**
```

Results:

```
NONVSAM DS:  BKMI.PAY.JUNE
              >> MATCHING FILTER:  BKMI.PAY.JUNE
              >> NEW NAME FILTER:  BKMI.+NEW.PAY.**
              >> NEW NAME       :  BKMI.NEW.PAY.JUNE
```

```
FULLRENAMEMASKS(BKMI.%.APRIL,BKMI.%.+NEW.APRIL)

Original DSN/Mask          =>  BKMI.%.APRIL
Rename to DSN/Mask        =>  BKMI.%.+NEW.APRIL
```

Results:

```
NONVSAM DS:  BKMI.B.APRIL
              >> MATCHING FILTER:  BKMI.%.APRIL
              >> NEW NAME FILTER:  BKMI.%.+NEW.APRIL
              >> NEW NAME       :  BKMI.B.NEW.APRIL
```

This final example shows what happens when you specify an ambiguous filter for the Rename to DSN/Mask:

```
FULLRENAMEMASKS(OLD.DATA.SET,NEW.**.SET.**)

Original DSN/Mask          =>  OLD.DATA.SET
Rename to DSN/Mask        =>  NEW.**.SET.**
```

Results:

```
BKM11313E NEW NAME MASK IS AMBIGUOUS :  NEW.**.SET.**
BKM11313E IDCAMS RENAME NOT DONE - ERRORS OR WARNINGS FOUND
```

Using FullRename for VSAM clusters

If an ICF VSAM cluster is renamed, all components of the cluster that match the mask for the original data set name will be renamed using the cluster's mask for the new data set name.

When using FullRename masks for VSAM data sets, note the following:

- If the **Rename to DSN/Mask** for a VSAM cluster or the **Rename to DSN/Mask** for any of its components already exists, no alters will be generated for the cluster or its components.
- New names for any AIXs cannot be resolved from the new name of the base cluster. They must be filtered (**Original DSN/Mask, Rename to DSN/Mask**) in the same way as the base cluster, and their components will be renamed in the same way.

Examples

When you rename a VSAM data set, the new name for all components, excluding any AIXs, will be the same as the cluster name, with the last node of the component appended.

Consider the following example, in which a VSAM sphere is renamed:

```
Cluster:    A.VSAM.DATASET
Data   :        A.VSAM.DATASET.DATA
Index  :        A.VSAM.DATASET.INDX
Path   :        A.VSAM.DATASET.PATH1
```

If, however, the last node of the new name for the cluster is CL or CLUS, the new name for the cluster, minus the last node, will be used to rename the components.

Using the VSAM data set above, if the cluster has a new name of A.TEST.VSAM.DATASET.CL, the component new names will be as follows:

```
Cluster:    A.TEST.VSAM.DATASET.CL
Data   :        A.TEST.VSAM.DATASET.DATA
Index  :        A.TEST.VSAM.DATASET.INDX
Path   :        A.TEST.VSAM.DATASET.PATH1
```

Now consider an example in which a single fully-qualified VSAM cluster is renamed:

```
FULLRENAMEMASKS(BKMI.XAA.PROD.AUG,PROD.XAA.PROD.AUG)

Original DSN/Mask          =>  BKMI.XAA.PROD.AUG
Rename to DSN/Mask        =>  PROD.XAA.PROD.AUG
```

Results:

```
VSAM SPHERE:  BKMI.XAA.PROD.AUG
              >> MATCHING FILTER:  BKMI.XAA.PROD.AUG
              >> NEW NAME FILTER:  PROD.XAA.PROD.AUG
              >> NEW NAME       :  PROD.XAA.PROD.AUG
  DATA :      BKMI.XAA.PROD.AUG.DATA
              >> NEW NAME       :  PROD.XAA.PROD.AUG.DATA
  INDEX:      BKMI.XAA.PROD.AUG.INDEX
              >> NEW NAME       :  PROD.XAA.PROD.AUG.INDEX
```

In this example, a single VSAM cluster ending with CL is renamed by removing the XAC qualifier:

```
FULLRENAMEMASKS(BKMI.XAC.PROD.AUG.CL,BKMI.-.PROD.AUG.CL)

Original DSN/Mask          =>  BKMI.XAC.PROD.AUG.CL
Rename to DSN/Mask        =>  BKMI.-.PROD.AUG.CL
```

Results:

```
  VSAM SPHERE: BKMI.XAC.PROD.AUG.CL
               >> MATCHING FILTER:  BKMI.XAC.PROD.AUG.CL
               >> NEW NAME FILTER:  BKMI.-.PROD.AUG.CL
               >> NEW NAME       :  BKMI.PROD.AUG.CL
   DATA :      BKMI.XAC.PROD.AUG.CL.DATA
               >> NEW NAME       :  BKMI.PROD.AUG.DATA
   INDEX:      BKMI.XAC.PROD.AUG.CL.INDEX
               >> NEW NAME       :  BKMI.PROD.AUG.INDEX
```

Now consider an example in which the NEW qualifier is added to the new name for a VSAM cluster:

```
FULLRENAMEMASKS(BKMI.XAB.PROD.AUG,BKMI.+NEW.XAB.PROD.AUG)

Original DSN/Mask          =>  BKMI.XAB.PROD.AUG
Rename to DSN/Mask        =>  BKMI.+NEW.XAB.PROD.AUG)
```

Results:

```
VSAM SPHERE:  BKMI.XAB.PROD.AUG
              >> MATCHING FILTER:  BKMI.XAB.PROD.AUG
              >> NEW NAME FILTER:  BKMI.+NEW.XAB.PROD.AUG
              >> NEW NAME       :  BKMI.NEW.XAB.PROD.AUG
  DATA :      BKMI.XAB.PROD.AUG.DATA
              >> NEW NAME       :  BKMI.NEW.XAB.PROD.AUG.DATA
  INDEX:      BKMI.XAB.PROD.AUG.INDEX
              >> NEW NAME       :  BKMI.NEW.XAB.PROD.AUG.INDEX
```

In this next example, all data sets having a high level qualifier of BKM are renamed to have PROD as their high level qualifier. This renaming includes both VSAM and non-VSAM data sets:

```
FULLRENAMEMASKS(BKM.**,PROD.**)

Original DSN/Mask        =>  BKM.**
Rename to DSN/Mask    =>  PROD.**
```

Results:

```
NONVSAM DS:  BKM.GEN1.DATA
              >> MATCHING FILTER:  BKM.**
              >> NEW NAME FILTER:  PROD.**
              >> NEW NAME       :  PROD.GEN1.DATA
NONVSAM DS:  BKM.PAY.TEST4
              >> MATCHING FILTER:  BKM.**
              >> NEW NAME FILTER:  PROD.**
              >> NEW NAME       :  PROD.PAY.TEST4
VSAM SPHERE:  BKM.PROD.AUG.CL
              >> MATCHING FILTER:  BKM.**
              >> NEW NAME FILTER:  PROD.**
              >> NEW NAME       :  PROD.PROD.AUG.CL
```

```
  DATA :      BKM.PROD.AUG.DATA
              >> NEW NAME       :  PROD.PROD.AUG.DATA
  INDEX:      BKM.PROD.AUG.INDEX
              >> NEW NAME       :  PROD.PROD.AUG.INDEX
VSAM SPHERE:  BKM.XAA.PROD.AUG
              >> MATCHING FILTER:  BKM.**
              >> NEW NAME FILTER:  PROD.**
              >> NEW NAME       :  PROD.XAA.PROD.AUG
  DATA :      BKM.XAA.PROD.AUG.DATA
              >> NEW NAME       :  PROD.XAA.PROD.AUG.DATA
  INDEX:      BKM.XAA.PROD.AUG.INDEX
              >> NEW NAME       :  PROD.XAA.PROD.AUG.INDEX
```

In the following example, all data sets with a high level qualifier of BKM are renamed to have high level qualifiers of PROD.NEW. Both VSAM and non-VSAM data sets are included in this renaming:

```
FULLRENAMEMASKS(BKM.**,PROD.+NEW.**)

Original DSN/Mask        =>  BKM.**
Rename to DSN/Mask        =>  PROD.+NEW**
```

Results:

```
NONVSAM DS:  BKM.GEN1.DATA
              >> MATCHING FILTER:  BKM.**
              >> NEW NAME FILTER:  PROD.+NEW.**
              >> NEW NAME       :  PROD.NEW.GEN1.DATA
NONVSAM DS:  BKM.PAY.TEST4
              >> MATCHING FILTER:  BKM.**
              >> NEW NAME FILTER:  PROD.+NEW.**
              >> NEW NAME       :  PROD.NEW.PAY.TEST4
VSAM SPHERE:  BKM.PROD.AUG.CL
              >> MATCHING FILTER:  BKM.**
              >> NEW NAME FILTER:  PROD.+NEW.**
              >> NEW NAME       :  PROD.NEW.PROD.AUG.CL
  DATA :      BKM.PROD.AUG.DATA
              >> NEW NAME       :  PROD.NEW.PROD.AUG.DATA
  INDEX:      BKM.PROD.AUG.INDEX
              >> NEW NAME       :  PROD.NEW.PROD.AUG.INDEX
VSAM SPHERE:  BKM.XAA.PROD.AUG
              >> MATCHING FILTER:  BKM.**
              >> NEW NAME FILTER:  PROD.+NEW.**
              >> NEW NAME       :  PROD.NEW.XAA.PROD.AUG
  DATA :      BKM.XAA.PROD.AUG.DATA
              >> NEW NAME       :  PROD.NEW.XAA.PROD.AUG.DATA
  INDEX:      BKM.XAA.PROD.AUG.INDEX
              >> NEW NAME       :  PROD.NEW.XAA.PROD.AUG.INDEX
```

The following is an example of a batch ARECOVER using the FULLRENAME feature. Data set names within the parameter **FULLRENAMEMASKS** can be separated by a space or a comma. The example shows a space between the **Original DSN/Mask** filter and the **Rename to DSN/Mask** filter.

```
//STEPNAME  EXEC  BKMBPROC
//BKMBATCH.SYSIN  DD  *
    ARECOVER AG(IVPSTD)                                          -
             GEN(0)                                              -
             EXEC                                                -
             TERMMSGS(YES)                                       -
             NOSTACK                                             -
             INCLUDE(                                            -
                     BKMI.A.TEST1                                -
                     BKMI.B.TEST2                                -
                     BKMI.C.TEST3                                -
                     BKMI.D.MINUS.MINUS.TEST4                    -
                     )                                           -
             FULLRENAMEMASKS(                                    -
              BKMI.A.TEST1      BKMI.+NEW.A.**                   -
              BKMI.%.TEST2      BKMI.%.+NEW.TEST2                -
              BKMI.C.TEST*      BKMI.X.**                        -
              BKMI.*.TEST4      BKMI.D.-.-.TEST4                 -
                              )                                  -
     FINALRENAMEACTION(EXEC)                                     -
     OUTPUTRENAMEDSN(                                            -
       &HSUMUID.&HOSTID.&AGGNAME.&DATE.&TIME.&USERID             -
                             )
```

About PREVENT_BACKLEVEL

Prevent BackLevel is intended to prevent data sets backed up by more than one ABARS aggregate (application) from backleveling another more current data set during an **ARECOVER**. Typically, it is used at disaster recovery sites only. This feature works only if you have overlapped data sets (data sets backed up by more than one ABARS aggregate).

At the disaster recovery site, a file containing a list of all data sets that were backed up in multiple selected aggregates (overlaps) will be used. This file is created at the disaster recovery site using the procedures described below. As **ARECOVER** commands are performed, this file is continuously checked to determine if the version of a data set being recovered is the most current version of that data set. If not, the recovery for that version is bypassed. The file needs to be deleted and re-created with the most current overlap information for each disaster recovery exercise.

Overlapped data sets and backleveling

If a data set is backed up by more than one aggregate or application, it is considered to be an *overlapped* data set. When it comes to recovering applications, if a data set is shared by several applications, an application can easily backlevel a shared data set. While it is best to identify overlaps in advance and exclude the data sets from the applications that don't own the data sets, this is not always possible.

Prevent BackLevel is an option that ensures application data sets are not back-leveled by older copies of data sets being restored. When a data set is to be restored, the data set version is compared to selected backup copies (overlaps) that have been extracted from the IDS and stored in the Prevent BackLevel data set. If this copy is not the most current copy of a data set, it is skipped and restored during the ARECOVER for the application containing the most current copy.

BKLEVEL parameter in BKMINI member of the parameter library

BKLEVEL can be specified as a default for all aggregates in the product parameter library member BKMINI, or at the aggregate level, and is supported in both batch and online submissions.

For the **BKLEVEL** parameter to work, a VSAM data set called the backlevel data set must be created and updated with a list of the overlapped data set names and associated aggregates prior to recovering any aggregates. When the **PREVENT_BACKLEVEL** parameters are specified for an ABARS Manager ARECOVER COMMAND, this VSAM backlevel data set is used as input during the ARECOVER process to compare the data set to be restored against all the overlaps to determine if this version is the most current. If the data set is not the most current copy, it is skipped. The following assumptions underlie the use of the **PREVENT_BACKLEVEL** parameter:

- All current aggregates are recovered.
- No applications are started until all aggregates are recovered.

Setting up Prevent BackLevel

To set up Prevent BackLevel:

1. Specify the backlevel data set name for the BKLEVEL token in the Advanced Backup and Recovery for z/OS parameter library member BKMINI.
2. Specify the backlevel data set name for the BACKLEVEL_DSN token and specify Y for the PREVENT_BACKLEVEL token in the Advanced Backup and Recovery for z/OS parameter library member BKMINI. The backlevel data set will be created when you execute step 3.
3. Define the VSAM backlevel data set specified above, create the Overlap by Event report, and load the backlevel data set by executing JCL member BKMBLVLA. Be sure to update the backlevel data set name in the JCL member to match the VSAM data set name you specified for the **BKLEVEL** parameter.

   The member deletes and defines the back-level data set. The PREVENT_BACKLEVEL keyword specifically instructs the OVERLAPE report processor to add data to the back-level data set. Consider adding the PREVENT_BACKLEVEL keyword to other JCL where REPORT OVERLAPE is invoked. Plan on deleting and redefining the back-level data set regularly. The COMPGEN keyword controls the

applicability of the aggregates considered as potentially regressive for restoring data sets, therefore the OVERLAPE report repopulates the new back-level data set appropriately.

Using Prevent BackLevel during an ARECOVER

The **PREVENT_BACKLEVEL** parameter is supported for the **ARECOVER** command both in batch and online. If the **PREVENT_BACKLEVEL** parameter is set to Y in your product parameter library member BKMINI, this parameter is added automatically to your online ARECOVER. If this parameter is set to N, and you want to issue this command for a specific aggregate(s), you must update the online ARECOVER **PREVENT_BACKLEVEL** setting to Y. If you submit your ARECOVERs in batch, the **PREVENT_BACKLEVEL** parameter must be added to the ARECOVER syntax.

Consider an example of a batch ARECOVER using the PREVENT_BACKLEVEL parameter. Two changes must occur:

- The symbolic for the backlevel data set is required in the batch ARECOVER JCL.
- A parameter of **PREVENT_BACKLEVEL** must be added to the SYSIN statements.

Example:

```
//*=====================================================
 //*  ARECOVER - Using PREVENT BACKLEVEL
 //*=====================================================
 // BKM EXEC BkMBPROC,
 //       LOAD1='prefix.value.SBKMLOAD',
 //       PARMLIB='prefix.value.SBKMPARM,
 //       DAD='DAD.DADHHATT.DAD',
 //       IDS='BKMX.PROD.INVDB'
 //* BACKLEVEL DSN OVERRIDE
 //BKMBATCH.BKLEVEL DD DISP=SHR, DSN='BKM.BACKLVL.DSN'
 //BKMBATCH.SYSIN DD *
 ARECOVER AG(addname)  +
                      GEN(0)  +
                      EXE  +
                      TERMMSGS(YES)  +
                      PREVENT_BACKLEVEL  +
                      UNIT(3490)
```

ARECOVER command examples

These examples show you how to initiate **ARECOVER** aggregate events in batch mode, using procedure BKMBPROC.

```
//STEPNAME  EXEC  BKMBPROC
//BKMBATCH.SYSIN  DD *
      ARECOVER  AG(aggname1) EXEC        -
                DATASETCONFLICT(REPLACE) -
                PRI
```

In this example, only primary data sets are recovered from aggregate *aggname1* using the REPLACE option. Because a generation value was not specified, the current generation (gen (0)) is used. By specifying PRI only, any ML1, ML2, or user tape data sets are not processed.

```
//STEPNAME  EXEC BKMBPROC
//BKMBATCH.SYSIN  DD *
      ARECOVER AG(aggname1)                -
               DATASETCONFLICT(REPLACE)    -
               INCLUDE( SYS2.JCL PROD.INV ) -
               GEN(-1)  VEREXE
```

In this second example, multiple aggregates are recovered using first VERIFY, then EXECUTE with the REPLACE option. The generation to be recovered is (-1). The two data sets to be recovered are SYS2.JCL and PROD.INV.

```
//STEPNAME  EXEC BKMBPROC
//BKMBATCH.SYSIN  DD *
  ARECOVER AG(aggname1 aggname2 aggname3 aggname4 -
              aggname5 aggname6 aggname7 aggname8 -
              aggname9 aggname10)                -
```

```
              PRI  EXEC                          -
              DATASETCONFLICT(REPLACE)
```

In this ARECOVER example, 10 aggregates (*aggname1* to *aggname10*) are recovered using Generation 0 (default), but only those primary data sets in each aggregate are recovered. Nine additional subtasks are created (for *aggname2–aggname10*), each executing in a separate DFSMShsm address space.

Using native ABARS ARECOVER

Because disaster recovery is critical for any organization, Advanced Backup and Recovery for z/OS provides two native ABARS **ARECOVER** options. You can write or submit native **ARECOVER** commands with very little ABARS Manager processing. These **ARECOVER** options are for disaster recovery use only.

ABARS Manager associates an aggregate with the C file name and volsers, with either of the following two **ARECOVER** options.

- Writes the native **ARECOVER** command for each aggregate into a separate partitioned data set member. That member can then be executed by TSO program IKJEFT01.
- Submits the native **ARECOVER** command for each aggregate directly to ABARS.

To use native ABARS **ARECOVER** processing, update JCL member BKMBPROC with either the **NATIVE,FILE** or **NATIVE,EXEC** parameters, described below.

Note: ABARS Manager features not provided by native ABARS cannot be used during native mode, such as Selective Data Set Restore, FullRename, GDG Replace, Prevent BackLevel.

When an **ARECOVER** has been submitted through ISPF, ABARS Manager finds the 'C' file and volsers for the specified aggregate and writes the native **ARECOVER** command into a PDS data set. The PDS is specified by parameter **NATIVE_MODE_PDS** in PARMLIB member BKMINI. The aggregate is not submitted to ABARS. No pre- or postprocessing occurs other than finding and writing the native ABARS ARECOVER with the 'C' file name and volsers to the PDS. No Advanced Backup and Recovery for z/OS Inventory Data Set (IDS) updates are done. Data required in the IDS can be placed there by loading the **ARECOVER** activity log.

Here is an example of the change made to BKMBPROC:

```
//BKMBATCH EXEC PGM=BKMBMAIN,PARM='BKMFUNC/NATIVE,FILE'
```

Here is an example TSO batch job:

```
//JOBCARD  JOB ,'TSOBTCH',CLASS=A,MSGCLASS=X,NOTIFY=PROD
//STEP1   EXEC  PGM=IKJEFT01
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD DSN=BKMX.IVP.NATIVE.PO.TEST(PROD01),
//            DISP=SHR
```

To use the WRITE option, type a PDS data set name in parameter **NATIVE_MODE_PDS**. This parameter can be found in the Advanced Backup and Recovery for z/OS PARMLIB member BKMINI. The PDS data set you specify will be used in writing the native **ARECOVER** commands.

The following rules apply when you are updating BKMINI:

- This option will write the HSEND **ARECOVER** command into the PDS data set as specified by NATIVE_MODE_PDS. No submission to ABARS occurs.
- A PDS data set is the only option, sequential files are not allowed.
- The PDS data set must already exist as LRECL=80.
- The PDS data set must be allocated large enough, and with enough directory blocks, to support all the aggregates.
- The aggregate name must be the PDS member name — no exceptions.
- The PDS data set may or may not have existing members in it. If members are in it, and they match an aggregate name, they are overwritten.

- When you are selecting and submitting aggregates through the ISPF panels, you must use the same job name because more than one member of a PDS cannot be updated at any one time.

When you use the NATIVE,EXEC option and an **ARECOVER** has been submitted through ISPF, ABARS Manager will find the C file and volsers for the specified aggregate and submit the native **ARECOVER** command directly to ABARS. No pre- or postprocessing occurs other than finding and sending the C file name and volsers to ABARS. No IDS updates are done. Data required in the IDS can be placed there by loading the **ARECOVER** activity log.

The following is an example of the appropriate BKMBPROC change:

```
//BKMBATCH EXEC PGM=BKMBMAIN,PARM=`BKMFUNC/NATIVE,EXEC
```

About the AGGMGMT command

You can use the **AGGMGMT** command in batch mode to add or delete ABARS Manager aggregates, set the number of backup or recover generations, and enable or modify the backup options.

If you add aggregates by mask, this option provides great flexibility by allowing aggregate settings (other than the defaults) to be set selectively.

If an aggregate matches the aggregate name or mask specified and is in the SMS address space, but is not in the Inventory Data Set (IDS), it is added to the IDS with the settings specified in the batch job. If you do not specify settings, the aggregates are added using the default settings from the Advanced Backup and Recovery for z/OS product parameter library member BKMINI. The **Backup Retain** value is set to the same value specified in SMS for that aggregate. The **Recover Retain** value will be set to the default value of 1.

By default, each aggregate is assigned the Backup Options specified in the **Backup Options** section in Advanced Backup and Recovery for z/OS product parameter library member BKMINI. These backup options prevent the **ABACKUP** from ending abnormally and issue the return code specified. You can change the values for these options by using the ISPF panels or by using the batch command **AGGMGMT**.

If you add aggregates by mask, this batch option is a quick alternative to adding aggregates manually, setting backup or recover generations, or changing the BKMINI backup options. Aggregate names, aggregate masks, or a combination of both can be used.

Submitting the AGGMGMT command

The **AGGMGMT** command is used to add aggregates to the Advanced Backup and Recovery for z/OS Inventory Data Set (IDS), change the number of backup or recover generations to retain, or change backup options in batch.

Submit the **AGGMGMT** command as SYSIN to the BKMBPROC member of the Advanced Backup and Recovery for z/OS JCL library.

AGGMGMT command syntax

This topic shows the **AGGMGMT** command syntax, provides parameter descriptions, and shows examples of its usage.

You can use either the + symbol or the - symbol as a line continuation character. Make certain that you leave at least one blank space at the end of the line, between the last parameter and the continuation symbol.

AGGMGMT(*aggname1*|*aggnamemask1* ...*aggnamen*|*aggnamemaskn*)

Optional parameters:

[BACKUP_GENS_GRPS_TO_RETAIN(nnnn)]

[CHG_ALC_DSN_B32760_TO_INC(N0|N4|N8|Y0|Y4)]

[CHG_ALC_OF_UN_DSORG_TO_INC(N0|N4|N8|Y0|Y4)]

[CHG_ALIAS_TO_TRUENAME(N0|N4|N8|Y0|Y4)]

[CHG_CATLG_TO_ALC(N0|N4|N8|Y0|Y4)]

[CHG_DASD_DSN_ACC_TO_INC(N0|N4|N8|Y0|Y4)]

[CHG_MIGR_DSN_ACC_TO_INC(N0|N4|N8|Y0|Y4)]

[CHG_TAPE_DSN_ALC_TO_INC(N0|N4|N8|Y0|Y4)]

[FUZZY_CAT_STATUS_RC(0,4,8)]

[INCR(Y|N)]

[MOVE_MULTI_CAT_DSN_TO_INC(N8|Y0|Y4)]

[RECOVERY_GENS_TO_RETAIN(*nnnn*)]

[REMOVE_CAT_ERROR_DSNS(N0|N4|N8|Y0|Y4)]

[REM_DCOL_DSN_CAT_DIFF_VOL(N0|N4|N8|Y0|Y4)]

[REMOVE_GDGB_REF_IF_NOT_ALC(N0|N4|N8|Y0|Y4)]

[REMOVE_ORPHANED_ALIASES(N0|N4|N8|Y0|Y4)]

[REMOVE_UNCAT_DSNS(N0|N4|N8|Y0|Y4)]

[DELETE]

Parameters

AGGMGMTAGG (*aggname*|*aggnamemask1 ...aggnamen*|*aggnamemaskn*)
Within the parentheses, you can specify multiple blank delimited aggregate names, or multiple blank delimited aggregate mask names, or a combination of aggregate and mask names.

If an aggregate matches the aggregate name or aggregate name mask and is in the SMS address space, but is not in the Inventory Data Set (IDS), it is added to the IDS with the settings specified in the batch job. If no settings are specified, the aggregate is added using the default settings from the PARMLIB member BKMINI. The **Backup Retain** value is set to the same value specified in ISMF/SMS for that aggregate.

[BACKUP_GENS_GRPS_TO_RETAIN(nnnn)]
Specify a numeric value between 1 and 9999 to indicate how many backup generation groups you want retained. By default, this value is that specified in ISMF/SMS for the aggregate.

[CHG_ALC_DSN_B32760_TO_INC(N0|N4|N8|Y0|Y4)]
If you specify Y, all data sets with blksize > 32760 that are found in the ALLOCATE category are moved to the INCLUDE category (ABARS does not support data sets with blksize > 32760 in the ALLOCATE category).

Valid RC for N = 0, 4, and 8.

Valid RC for Y = 0 and 4.

[CHG_ALC_OF_UN_DSORG_TO_INC(N0|N4|N8|Y0|Y4)]
If you specify Y, whenever command processing finds a data set that has an unsupported DSORG in the ALLOCATE category, that data set is moved to the INCLUDE category (ABARS does not support certain DSORGs in the ALLOCATE category).

Valid RC for N = 0, 4, and 8.

Valid RC for Y = 0 and 4.

[CHG_ALIAS_TO_TRUENAME(N0|N4|N8|Y0|Y4)]
If you specify Y, for each data set in the SDS that is an alias of another data set, the AGGMGMT command replaces the data set's alias with the true name.

Valid RC for N = 0, 4, and 8.

Valid RC for Y = 0 and 4.

[CHG_CATLG_TO_ALC(N0|N4|N8|Y0|Y4)]
If you specify Y, all ICF catalogs found in the Selection Data Set are moved to the ALLOCATE list.

Valid RC for N = 0, 4, and 8.

Valid RC for Y = 0 and 4.

[CHG_DASD_DSN_ACC_TO_INC(N0|N4|N8|Y0|Y4)]
If you specify Y, all DASD data sets found in the ACCOMPANY category are moved to the INCLUDE category (ABARS does not support DASD data sets in the ACCOMPANY category).

Valid RC for N = 0, 4, and 8.

Valid RC for Y = 0 and 4.

[CHG_MGR_DSN_ALC_TO_INC(N0|N4|N8|Y0|Y4)]
If you specify Y, all migrated data sets found in the ACCOMPANY category are moved to the INCLUDE category (ABARS does not support migrated data sets in the ACCOMPANY category).

Valid RC for N = 0, 4, and 8.

Valid RC for Y = 0 and 4.

[CHG_TAPE_DSN_ALC_TO_INC(N0|N4|N8|Y0|Y4)]
If you specify Y, all TAPE data sets found in the ALLOCATE category are moved to the INCLUDE category (ABARS does not support TAPE data sets in the ALLOCATE category).

Valid RC for N = 0, 4, and 8.

Valid RC for Y = 0 and 4.

[FUZZY_CAT_STATUS_RC(0,4,8)]
A return code of 0 or 4 causes the backup to be submitted, even though it might have problems or fail.

A return code of 8 causes the backup to fail.

During ABARS Manager preprocessing of an **ABACKUP** request, ABARS Manager can detect any changes to the catalog entry of a data set referenced from an SDS. These changes can occur when the application is still running and a volume is added to a data set, or a new generation data set is created. If catalog changes are ignored by ABARS Manager, then errors can occur and the **ABACKUP** is no longer a point-in-time backup.

[INCR(Y|N)]
If set as Y for an aggregate or aggregate mask, it will check SMS and make sure it finds 9999 or NOLIMIT, or else it will skip it and report it in SYSOUT. Therefore, it is important that the SMS values must be set up correctly.

[MOVE_MULTI_CAT_DSN_TO_INC(N8|Y0|Y4)]
If you specify Y, all data sets found in multiple categories (INCL, ALLOC, ACCOMP) are moved to the appropriate category.

Valid RC for N = 8.

Valid RC for Y = 0 and 4.

[RECOVERY_GENS_TO_RETAIN(*nnnn*)]
Specify a numeric value between 1 and 9999.

[REMOVE_CAT_ERROR_DSNS(N0|N4|N8|Y0|Y4)]
Use this option to specify what you want done when AGGMGMT encounters a catalog error as it obtains information about a data set in the SDS. Specify Y if you want the problem data set removed from the SDS.

Valid RC for N = 0, 4, and 8.

Valid RC for Y = 0 and 4.

[REM_DCOL_DSN_CAT_DIFF_VOL(N0|N4|N8|Y0|Y4)]
If Y, then when data set names are obtained using the DCOLLECT option of SDSL, the possibility exists that a data set from the volume(s) selected is cataloged but the catalog entry points to the same data set name on another volume (for example, NOT CAT 2).

For *non-VSAM* data sets, if the data set name is not removed from the list, ABARS backs up the data set from the volume pointed to by its catalog entry. In other words, if this option is set to No, data sets

may be backed up from volumes other than the volume(s) specified in the SDSL DCOLLECT candidate list.

Specify Y (yes), remove the data set name, if you do not want the data set that the catalog points to included in this backup. Note that because ABARS only backs up data sets by locating their volume(s) from their catalog entries, the only version of a 'duplicate' data set that ABARS will back up is the one pointed to by its catalog entry.

VSAM data sets are not affected by this option setting. If a sphere component's catalog entry points to a different volume, the component (converted to the cluster name) will be left in the selection list.

Valid RC for N = 0, 4, and 8.

Valid RC for Y = 0 and 4.

[REMOVE_GDGB_REF_IF_NOT_ALC(N0|N4|N8|Y0|Y4)]
If you specify Y, any data set in the SDS that is a GDG base and is in any category other than ALLOCATE is removed (because ABARS does not support the GDG base name in any category but ALLOCATE).

Valid RC for N = 0, 4, and 8.

Valid RC for Y = 0 and 4.

[REMOVE_ORPHANED_ALIASES(N0|N4|N8|Y0|Y4)]
If you specify Y, command processing removes from the SDS any data set that is an alias of a data set that is no longer cataloged.

Valid RC for N = 0, 4, and 8.

Valid RC for Y = 0 and 4.

[REMOVE_UNCAT_DSNS(N0|N4|N8|Y0|Y4)]
If you specify Y, all data sets in the SDS that are not cataloged are removed from the SDS.

Valid RC for N = 0, 4, and 8.

Valid RC for Y = 0 and 4.

[DELETE]
Deletes the aggregate and all its events from the Advanced Backup and Recovery for z/OS Inventory Data Set.

Note: When using the DELETE parameter, masks are not supported.

Rules for batch processing

If an aggregate matches the aggregate name or aggregate name mask and is in the SMS address space, but is not in the IDS, it is added to the IDS with the settings specified in the batch job. If no settings are specified, the aggregate is added using the default settings from the Advanced Backup and Recovery for z/OS PARMLIB member BKMINI. The Backup Retain value is set to the same value specified in ISMF/SMS for that aggregate. The Recover Retain value is set to 1.

If you need to change a single setting, or selected settings for an aggregate or aggregate mask, include only those settings in the batch job. The settings not included in the batch job remain unchanged.

If INCR is set as Y for an aggregate or aggregate mask, processing will check SMS for a limit value of 9999 or NOLIMIT; or else it will skip it and report it in SYSOUT. Therefore, the SMS values must be set up correctly before the command is processed.

SDSL will not be supported as a switch because the SDSL code must exist. ABARS Manager asks if the aggregate should be SDSL after the code has been entered and the panel exited.

All error messages appear in a concise report at the end of SYSOUT.

Please note that if optional parameter INCR(Y | N) is used, and the aggregate has not been set up with a limit of 9999 or 'nolimit', no settings will take place for that aggregate.

Examples

The following example shows how to initiate the AGGMGMT command using procedure BKMBPROC.

In this example, an aggregate mask, P*, is being used.

- If any aggregates matching mask P* are not already in the Inventory Data Set, they are added.
- The backup generations to retain is set at 7 and the recovery generations to retain is set at 2. If these parameters were omitted from this batch job, the value for backup generations would be set using the default value defined in SMS.
- The recovery generations would be set to 1.
- All backup options settings included in the batch job will override the corresponding setting values in PARMLIB member BKMINI. Note that the settings for backup options not included in the batch job retain the values specified for them in BKMINI.

```
//STEPNAME  EXEC BKMBPROC
//BKMBATCH.SYSIN  DD *
    AGGMGMT AGG(P*)                 +
         BACKUP_GENS_GRPS_TO_RETAIN(7)   +
            RECOVERY_GENS_TO_RETAIN(2)      +
            MOVE_MULTI_CAT_DSN_TO_INC(Y4)   +
            CHG_ALC_OF_UN_DSORG_TO_INC(Y4)  +
            CHG_DASD_DSN_ACC_TO_INC(Y4)     +
            CHG_TAPE_DSN_ALC_TO_INC(Y4)     +
            CHG_ALC_DSN_B32760_TO_INC(Y4)   +
            REMOVE_CAT_ERROR_DSNS(Y4)       +
            REMOVE_UNCAT_DSNS(Y4)           +
            CHG_ALIAS_TO_TRUENAME(Y4)       +
            REMOVE_ORPHANED_ALIASES(Y4)     +
            CHG_CATLG_TO_ALC(Y4)            +
            REMOVE_GDGB_REF_IF_NOT_ALC(Y4)  +
            FUZZY_CAT_STATUS_RC(4)          +
            REM_DCOL_DSN_CAT_DIFF_VOL(Y4)
```

About the CONVSDSL command

The **CONVSDSL** command converts SDSL syntax to ABARS syntax so it can be examined without immediate submission; it is an optional technique that provides dynamic, real-time analysis of data sets for selection in an aggregate.

Based on the specified parameters, the results update the Selection Data Set with a list of specific data sets matching the coded criteria prior to backup processing. This standardizes frequent changes to your installation's data sets by automatically examining them each time an **ABACKUP EXECUTE** or **CONVSDSL** command is executed to include the matching file names in an aggregate's Selection Data Set.

The **ABACKUP EXECUTE** command gives you dynamic real-time analysis at the time of backup. When the **ABACKUP** is executed, the SDSL code is analyzed and the data sets are written to the Selection Data Set for immediate ABARS processing.

The **CONVSDSL** command builds the Selection Data Set in real-time and can be used in the following ways:

- To examine the resulting Selection Data Set to ensure you achieve the results you want.
- As immediate input to ABARS Aggregate LoadBalancer to split up the Selection Data Set. An ABARS Aggregate LoadBalancer then triggers the **ABACKUP** job.

Submitting the CONVSDSL command

Use the **CONVSDSL** command to convert SDSL syntax to ABARS syntax so you can examine it without having to submit it immediately for execution. Submit the **CONVSDSL** command as SYSIN to the BKMBPROC member of the Advanced Backup and Recovery for z/OS JCL library.

CONVSDSL command syntax

This topic shows the **CONVSDSL** command syntax, provides parameter descriptions, and shows examples of its usage.

You can use either the + symbol or the - symbol as a line continuation character. Make certain that you leave at least one blank space at the end of the line, between the last parameter and the continuation symbol.

CONVSDSL AG(*aggname1* | *aggnamemask1 ...aggnamen* | *aggnamemaskn*)

Parameters

AG(*aggname1* | *aggnamemask1 ...aggnamen* | *aggnamemaskn*)
Within the parentheses, you can specify multiple blank delimited aggregate names, aggregate name masks, or logical aggregate names. Up to 256 aggregates can be processed with one **CONVSDSL** command. You cannot use a mask to specify logical aggregate names. Any number of aggregate names may be included with a **CONVSDSL** command for concurrent execution.

Example

The following example shows how to initiate **CONVSDSL** commands in batch mode using procedure BKMBPROC.

```
//STEPNAME  EXEC  BKMBPROC
//BKMBATCH.SYSIN  DD *
    CONVSDSL AG(PAYD01)
```

In this example, the SDSL statements for PAYD01 are resolved and written to PAYD01's Selection Data Set. This step can be followed with the ABARS Aggregate LoadBalancer program to split up the Selection Data Set contents into other ABARS Selection Data Sets, or followed with an **ABACKUP**. An **ABACKUP** automatically resolves the SDSL code prior to submission to ABARS for backup. Therefore, following **CONVSDSL** with an **ABACKUP** is not necessary unless you are executing both programs separately to review and test the aggregate's contents.

About the LOADACT command

Use the **LOADACT** (Load Activity Log) command to load the contents of an ABARS activity log into the Advanced Backup and Recovery for z/OS Inventory Data Set (IDS), in batch mode.

Submitting the LOADACT command

Submit the **LOADACT** command as SYSIN to the BKMBPROC member of the Advanced Backup and Recovery for z/OS JCL library. You can also use the ISPF panels to submit the **LOADACT** command.

LOADACT command syntax

This topic shows the **LOADACT** command syntax, provides parameter descriptions, and shows examples of its usage.

You can use either the + symbol or the - symbol as a line continuation character. You must leave at least one blank space at the end of the line, between the last parameter and the continuation symbol.

LOADACT ACTLOG(*activity-log-name*)STACK|NOSTACK

Optional parameters:

[INCREMENTAL]

Parameters

ACTLOG(*activity-log-name*)
Specify the ABARS activity log name of the ABACKUP or ARECOVER event to load into the Inventory Data Set.

STACK|NOSTACK
Specify whether this ABACKUP event's output was STACKED or not.

[INCREMENTAL|INCR]
Specifies that this backup event is an incremental version. If you do not specify this parameter, the ABACKUP is assumed to be a base backup.

Example

The following example shows how to initiate **LOADACT** commands in batch mode using procedure BKMBPROC.

```
//STEPNAME  EXEC  BKMBPROC
//BKMBATCH.SYSIN  DD *
    LOADACT  ACTLOG(activity-log-name) STACK
```

In this example, the contents of the specified activity log will be loaded into the Inventory Data Set. STACK was specified identifying that the **ABACKUP** that ran and created this activity log was run with the output stacked.

```
//STEPNAME  EXEC  BKMBPROC
//BKMBATCH.SYSIN  DD *
    LOADACT  ACTLOG(activity-log-name)
```

In this example, the contents of the specified activity log will be loaded into the Inventory Data Set.

About the LOADIDS command

Use the **LOADIDS** (Load Inventory Data Set) command to perform a batch load of a backup copy (previously unloaded) of the Advanced Backup and Recovery for z/OS Inventory Data Set.

Submitting the LOADIDS command

Submit the **LOADIDS** command as SYSIN to the BKMBPROC member of the Advanced Backup and Recovery for z/OS JCL library. You can also use the ISPF panels to submit the **LOADIDS** command.

LOADIDS command syntax

This topic shows the **LOADIDS** command syntax, provides parameter descriptions, and shows examples of its usage.

Note: The default is to load ALL data previously extracted from the Inventory Data Set. ALL data refers to all aggregate records from ABARS Manager and all job records from Critical Backup Tracking and Inventory. Optional parameters can be used to selectively load previously extracted Inventory Data Set data, instead of the default of all extracted data. Be aware that if selected loads are required, it is recommended that the selection be made during the **UNLOADIDS** rather than unloading everything and selectively loading to increase speed.

You can use either the + symbol or the - symbol as a line continuation character. Make certain that you leave at least one blank space at the end of the line, between the last parameter and the continuation symbol.

LOADIDS INFILE(*dsn*)

[AG(*|aggname1|aggnamemask1|logaggname1 ...aggnamen|aggnamemaskn*)]

[ARECOVERS(*|nn*)]

[EXCLUDE(*|aggname1|aggnamemask1|logaggname1 ...aggnamen|aggnamemaskn*)]

[GENS(*nn*)]

[NOERRORS]

[NOINCOMPLETE]

[NOWARNINGS]

[REPLACE]

[VERIFIES(*|*nn*)]

Parameters

INFILE(*dsn*)
Specify the data set name of the file containing previously unloaded Inventory Data Set records. You cannot enclose the data set name in quotes and you cannot specify the name of a GDG base.

[AG(*|*aggname1*|*aggnamemask1*|*logaggname1 ...aggnamen*|*aggnamemaskn*)]
Use this parameter to specify aggregates to be loaded.

The **AG** parameter is required if you want to load ABARS Manager data either selectively (only the aggregates you are interested in) or separately from Critical Backup Tracking and Inventory. You can specify an aggregate name, an aggregate mask or a logical aggregate name.

The **JOB** parameter is a Critical Backup Tracking and Inventory parameter. The relationship between **AG** and **JOB** is as follows:

- * causes all aggregates to be loaded.
- If both **AG** and **JOB** parameters are omitted, all data is loaded.
- If only one of these parameters is specified, only the data specified is loaded.

[ARECOVERS(*|*nn*)]
Specify which **ARECOVER** generations to load.

- *nn* – Specify the number of generations of previously loaded aggregate ARECOVERS, you want loaded when the command executes. Generations are loaded in reverse chronological order, newest to oldest.
- * – (Default) Specify a single asterisk if you want all previously loaded generations of aggregate ARECOVERS to be loaded when the command executes.

[EXCLUDE(*|*aggname1*|*aggnamemask1*|*logaggname1 ...aggnamen*|*aggnamemaskn*)]
Specify which aggregates you want excluded from the load. You specification can be an aggregate name, an aggregate mask, or a logical aggregate name.

[GENS(*nn*)]
Specify which **ABACKUP** generations you want loaded.

- nn – Specify the number of generations you want loaded, in reverse chronological order (from newest to oldest).
- *– Specify a single asterisk to have all **ABACKUP** generations loaded.

[NOERRORS]
Use this parameter to indicate that you want to skip events that have errors. This pertains to **ABACKUPS**, **ARECOVERS**, and **VERIFIES**.

[NOINCOMPLETE]
Use this parameter to indicate that you want to skip events whose status is incomplete. This pertains to **ABACKUPS**, **ARECOVERS**, and **VERIFIES**.

[NOWARNINGS]
Use this parameter to indicate that you to skip events that contain warnings. This pertains to **ABACKUPS**, **ARECOVERS**, and **VERIFIES**. This parameter cannot be used unless you have also specified the **AG** parameter.

[REPLACE]
Use this parameter to specify that you want to replace current entries. Only one use of **REPLACE** is allowed per execution of **LOADIDS**. This parameter is common to ABARS Manager and Critical Backup Tracking and Inventory.

[VERIFIES(*|*nn*)]
Specify which ABACKUP VERIFIES generations to load.

- nn – Specify the number of previously loaded generations of aggregate ABACKUP VERIFIES you want loaded. Generations are loaded in reverse chronological order, from newest to oldest.
- * – Specify a single asterisk to have all previously loaded generations of aggregate ABACKUP VERIFIES loaded when the command is executed.

Examples

The following example shows how to initiate **LOADIDS** commands in batch mode using procedure BKMBPROC.

In this example, both ABARS Manager and Critical Backup Tracking and Inventory parameters are used:

```
//STEPNAME  EXEC  BKMBPROC
//BKMBATCH.SYSIN  DD *
    LOADIDS INFILE(dsn)
```

In this example, all data previously extracted from the IDS is loaded from the specified **INFILE(dsn)** into the IDS.

Note: Quotes around the data set name are not valid.

```
//STEPNAME  EXEC  BKMBPROC
//BKMBATCH.SYSIN  DD *
    LOADIDS INFILE(PROD.BKM.INVDB.UNLOAD.IDS) -
           AG(*) GENS(2) ARECOVERS(*) VERIFIES(*)
```

In this example, two generations of **ABACKUPS**, all generations of **ARECOVERS**, and all generations of **ABACKUP VERIFIES** are loaded from the **INFILE(dsn)** into the Inventory Data Set.

Note: Quotes around the data set name are not valid

```
//STEPNAME  EXEC  BKMBPROC
//BKMBATCH.SYSIN  DD *
    LOADIDS INFILE(PROD.BKM.INVDB.UNLOAD.IDS) -
           AG(*) JOB(PAY*) EXCLJOB(PAY1) GENS(3)
```

In this example, three generation of all aggregates, and three generations of Critical Backup Tracking and Inventory tracked jobs beginning with PAY* (except for PAY1) are loaded from the **INFILE(dsn)** into the Inventory Data Set.

Note: Quotes around the data set name are not valid.

About the UNLOADIDS command

Use the **UNLOADIDS** (Unload Inventory Data Set) command to perform an unload, in batch, of the data from the Inventory Data Set, to be used as a backup, or for other purposes.

ABARS Manager and Critical Backup Tracking and Inventory share the same Inventory Data Set (IDS). If you want to remove Critical Backup Tracking and Inventory entries from the Inventory Data Set, you can use **UNLOADIDS** in conjunction with **LOADIDS**:

1. Execute an ABARS Manager **UNLOADIDS** to extract the ABARS backups.
2. Delete the Inventory Data Set.
3. Define a new Inventory Data Set.
4. Use **LOADIDS** to reload the previously unloaded ABARS Manager entries.

Submitting the UNLOADIDS command

Submit the **UNLOADIDS** command as SYSIN to the BKMBPROC member of the Advanced Backup and Recovery for z/OS JCL library.

UNLOADIDS command syntax

This topic shows the **UNLOADIDS** command syntax, provides parameter descriptions, and shows examples of its usage.

Note: The default is to load ALL data previously extracted from the Inventory Data Set. ALL data refers to all aggregate records from ABARS Manager and all job records from Critical Backup Tracking and Inventory. Optional parameters can be used to selectively load previously extracted Inventory Data Set data, instead of the default of all extracted data. Be aware that if selected loads are required, it is recommended that the selection be made during the **UNLOADIDS** rather than unloading everything and selectively loading to increase speed.

You can use either the + symbol or the - symbol as a line continuation character. You must leave at least one blank space at the end of the line, between the last parameter and the continuation symbol.

```
UNLOADIDS
[AG(*|aggname1|aggnamemask1|logaggname1 ...aggnamen|aggnamemaskn)]
[JOBN(*|jobname1|jobnamemask1|logjobname1, ...jobnamen|jobnamemaskn)]ARECOVERS(*|nn)]
[DATE(yyyymmdd) [TIME(hhmmss)]]
[EXCLUDE(*|aggname1|aggnamemask1|logaggname1 ...aggnamen|aggnamemaskn)]
[GENS(nn)]
[EXCLUDEMC(mgmtclass1 | mgmtclassmask1... mgmtclassn | mgmtclassmaskn)]
[GENS(nn)]
[INCLUDEMC(mgmtclass1 | mgmtclassmask1... mgmtclassn | mgmtclassmaskn)]
[NOERRORS]
[NOINCOMPLETE]
[NOWARNINGS]
[OUTFILE(dsn)]
[VERIFIES(*|nn)]
```

Parameters

[AG(*|*aggname1*|*aggnamemask1*|*logaggname1 ...aggnamen*|*aggnamemaskn*)]
Use this parameter to specify aggregates to be unloaded.

The **AG** parameter is required if you want to unload ABARS Manager data either selectively (only the aggregates you are interested in) or separately from Critical Backup Tracking and Inventory. You can specify an aggregate name, an aggregate mask or a logical aggregate name.

The **JOB** parameter is a Critical Backup Tracking and Inventory parameter. The relationship between **AG** and **JOB** is as follows:

- * causes all aggregates to be unloaded.
- If both **AG** and **JOB** parameters are omitted, all data is unloaded.
- If only one of these parameters is specified, only the data specified is unloaded.

[JOBN(*|*jobname1*|*jobnamemask1*|*logjobname1 ...jobnamen*|*jobnamemaskn*)]
Use this parameter to specify which Critical Backup Tracking and Inventory tracked backup jobs are to be unloaded.

The **JOBN** parameter is required if you want to unload Critical Backup Tracking and Inventory data either selectively (only the backup jobs you are interested in) or separately from Critical Backup Tracking and Inventory. You can specify a job name, a job name mask, or a logical job name.

The **JOB** parameter is a Critical Backup Tracking and Inventory parameter. The relationship between **AG** and **JOB** is as follows:

- * causes all jobs to be unloaded.
- If both **AG** and **JOB** parameters are omitted, all data is unloaded.
- If only one of these parameters is specified, only the data specified is unloaded.

[ARECOVERS(*|*nn*)]
Specify which **ARECOVER** generations to unload.

- *nn* – Specify the number of generations of aggregate **ARECOVERS** you want unloaded when the command executes. Generations are unloaded in reverse chronological order, newest to oldest.
- * – (Default) Specify a single asterisk if you want all generations of aggregate **ARECOVERS** to be unloaded when the command executes.

[DATE(*yyyymmdd*) [TIME(*hhmmss*)]]
DATE() and **TIME()** are additional keywords for **JOB()** selection and are alternatives to **GENS()**. Provide **DATE()** and **TIME()** values that match the job event time for jobs that were captured by this product.

Specifying **DATE()** and **TIME()** provides specific selection of data to unload without needing the relative generation it belongs. **DATE()** and **TIME()** values that contain wildcards select a wider range of data than **GENS()**.

You cannot specify **TIME** unless you also specify **DATE**; **TIME** values that are specified without a corresponding **DATE** value are ignored.

DATE values must have the format yyyymmdd and may contain wildcards. **TIME** values must have the format hhmmss and may contain wildcards.

If you do not specify **TIME()**, the selection process bypasses the comparison for time:

- Omit **TIME()** if you want to obtain data from multiple candidate jobs from the same date.
- Specify **TIME()** if you want to granularly limit that data to a single instance of a job within multiple candidate jobs.

Note: DATE, **TIME**, and **GENS** are not mutually exclusive, but **UNLOADIDS** is unlikely to produce any relevant results if they are specified together.

[EXCLUDE(*|aggname1|aggnamemask1|logaggname1 ...aggnamen|aggnamemaskn*)]
Specify which aggregates you want excluded from the unload. You specification can be an aggregate name, an aggregate mask, or a logical aggregate name.

[GENS(*nn*)]
Specify which **ABACKUP** generations you want unloaded.

- nn – Specify the number of generations you want unloaded, in reverse chronological order (from newest to oldest).
- *– Specify a single asterisk to have all **ABACKUP** generations unloaded.

[INCLUDEMC(*mgmtclass1 | mgmtclassmask1... mgmtclassn | mgmtclassmaskn*)]
Use this parameter to include in the unload only those aggregates matching the specified management classes or masks. You cannot use this parameter unless you also specify the **AG** parameter.

[NOERRORS]
Use this parameter to indicate that you want to skip events that have errors. This pertains to **ABACKUPS**, **ARECOVERS**, and **VERIFIES**

[NOINCOMPLETE]
Use this parameter to indicate that you want to skip events whose status is incomplete. This pertains to **ABACKUPS**, **ARECOVERS**, and **VERIFIES**

[NOWARNINGS]
Use this parameter to indicate that you to skip events that contain warnings. This pertains to **ABACKUPS**, **ARECOVERS**, and **VERIFIES**. This parameter cannot be used unless you have also specified the **AG** parameter.

[OUTFILE(*dsn*)]
Specify the name of the output data set into which you want the IDS records unloaded.

If the dsn specified in the **OUTFILE** parameter does not exist, it will be allocated dynamically. The default allocation is:

```
SPACE=(CYL,(1,1)),UNIT=SYSALLDA,
RECFM=VBS,LRECL=32756,BLKSIZE=32760
```

The default allocation values may be changed by modifying the values of the following tokens in the :PROCESSING_OPTIONS section of the BKMINI PARMLIB member:

```
UNLOADIDS_SPACE = 'CYLINDERS SPACE(1 2)'
UNLOADIDS_UNIT = SYSALLDA
```

You cannot surround dsn with quotes. Only one **OUTFILE** specification can be included in a single UNLOADIDS. If you do not specify the **OUTFILE** parameter, a report is created, but no IDS records are unloaded.

[VERIFIES(*|nn)]
For an unload of selected **VERIFIES**, use this parameter to specify which **ABACKUP VERIFIES** are to be unloaded.

- nn – Specify the number of generations of aggregate **ABACKUP VERIFIES** to unload in reverse chronological order, from newest to oldest.
- * – Specify a single asterisk to have all generations of aggregate **ABACKUP VERIFIES** unloaded.

Examples

The following example shows how to initiate **UNLOADIDS** commands in batch mode using procedure BKMBPROC.

In this example, both ABARS Manager and Automated Critical Data Identification parameters are specified:

```
//STEPNAME  EXEC  BKMBPROC
//BKMBATCH.SYSIN  DD *
     UNLOADIDS OUTFILE(dsn)
```

All data is unloaded from the Inventory Data Set into the specified **OUTFILE(dsn)**, including both ABARS Manager and Critical Backup Tracking and Inventory records. Had the optional parameter **OUTFILE(dsn)** not been specified, a report is created, listing the aggregates and jobs unloaded from the Inventory Data Set; however, no IDS records would have been unloaded.

Note: Quotes around the data set name are not valid.

```
//STEPNAME  EXEC  BKMBPROC
//BKMBATCH.SYSIN  DD *
    UNLOADIDS OUTFILE(PROD.BKM.INVDB.UNLOAD.IDS) -
              AG(*) GENS(2) ARECOVERS(2) VERIFIES(1)
```

In this example, all aggregates in the Inventory Data Set are unloaded into the specified **OUTFILE(dsn)**. Two generations of **ABACKUPS**, two generations of **ARECOVERS**, and one generation of **ABACKUP VERIFIES** are unloaded from the Inventory Data Set.

```
//STEPNAME  EXEC  BKMBPROC
//BKMBATCH.SYSIN  DD *
```

```
    UNLOADIDS OUTFILE(PROD.BKM.INVDB.UNLOAD.IDS) -
              AG(*) JOB(PAY*) EXCLJOB(PAY1) GENS(3)
```

In this example, three generation of all aggregates, and three generations of Critical Backup Tracking and Inventory tracked jobs beginning with PAY* (except for PAY1) are unloaded from the Inventory Data Set.

UNLOADIDS record counts

Record counts are calculated as follows for ABARS Manager:

- The **VERIFIES** count applies only to **ABACKUPS**.
- The **ARECOVERS** count is the latest n number of **ARECOVERS** (either VERIFY or EXEC).
- The Incremental Backup count is a base and all its related incremental backups.
- The Incremental **ARECOVER** count is the latest n number of **ARECOVERS** (either VERIFY or EXEC). **ARECOVERS** for incremental cannot be grouped like the **ABACKUPS** because there is no base or incremental identifier associated with it.

About the REPORT command

Use the **REPORT** command to submit Advanced Backup and Recovery for z/OS output reports. Several types of report are supported.

- OVERLAPD – Overlap by Data Set Report
- OVERLAPD-RVC – Overlap by Data Set Recovery Report
- OVERLAPE – Overlap by Event Report
- DSNLIST – Data Set List Report
- EVTLIST – Event List Report
- TAPEPULL – Tape Pull List Report
- SPACE – Disaster Recovery Space Summary Report

Submitting the REPORT command

REPORT commands are passed to ABARS in batch mode using the BKMBPROC member of the Advanced Backup and Recovery for z/OS JCL library. Submit the **REPORT** command as SYSIN to the BKMBPROC member of the JCL library.

Any number of **REPORT** commands can be included as SYSIN in a BKMBPROC invocation. You can specify only one type of report in each **REPORT** command.

The following example shows a request for multiple report types:

```
//STEPNAME EXEC BKMBPROC
//BKMBATCH.SYSIN DD *
  REPORT OVERLAPD DS(dsname1) SPACE(K,2)
  REPORT OVERLAPE EV(evname1 evname2*)  -
         GEN(0)                 -
         COMPGEN(-5,-10)
  REPORT DSNLIST DS(dsname1)
```

REPORT OVERLAPD command syntax

This topic shows the **REPORT OVERLAPD** command syntax that generates the Overlap by Data Set report, provides parameter descriptions, and shows examples of its usage.

You can use either the + symbol or the - symbol as a line continuation character. Make certain that you leave at least one blank space at the end of the line, between the last parameter and the continuation symbol.

REPORT OVERLAPD DS(*dsn1 ...dsnn* | *dsnmask1 ...dsnmaskn*)

[INCLEVT(*event name*)]

[EXCLEVT(*event name*)]

Parameters

OVERLAPD
Specify this subcommand to create an Overlap by Data Set report.

DS(*dsn*)
Use this parameter to specify which data sets you want searched for overlaps. You can specify data set names or name masks.

[INCLEVT(*event name*)]
Specify one or more fully-qualified event names, or one or more event masks to be included in the overlap search. Masked names must conform to the Advanced Backup and Recovery for z/OS extended masking rules.

[EXCLEVT(*event name*)]
Specify one or more fully-qualified event names to be excluded from the overlap search. If you specify this parameter without also specifying the **INCLEVT** parameter, any event except those that match the **EXCLEVT** parameter can be selected. When used with **INCLEVT** parameter, any event that matches the **INCLEVT** parameter and does not match the **EXCLEVT** parameter is selected.

Examples

The following examples show how to initiate Overlap by Data Set reports in batch mode using procedure BKMBPROC.

```
//STEPNAME EXEC BKMBPROC
//BKMBATCH.SYSIN DD *
  REPORT OVERLAPD DS(dsname1)
```

In this example, only one data set is searched for overlaps.

A second method is to code the same JCL, but have more than one data set name in the REPORT OVERLAPD command.

```
//STEPNAME EXEC BKMBPROC
//BKMBATCH.SYSIN DD *
  REPORT OVERLAPD DS(dsname1 dsname2*)
```

In this case, multiple overlap by data set searches are done. Overlaps are searched for 'dsname1' and any data set matching the mask 'dsname2*'.

The output for the Overlap by Data Set report can be found under the DD name BKMBREPT in the batch SYSOUT.

REPORT OVERLAPD-RCV command syntax

This topic shows the **REPORT OVERLAPD-RCV** command syntax that generates the Overlap by Data Set Recovery report, provides parameter descriptions, and shows examples of its usage.

You can use either the + symbol or the - symbol as a line continuation character. Make certain that you leave at least one blank space at the end of the line, between the last parameter and the continuation symbol.

REPORT OVERLAPD-RCV DS(*dsn1 ...dsnn*|*dsnmask1 ...dsnmaskn*)

[INCLEVT(*event name*)]

[EXCLEVT(*event name*)]

Parameters

OVERLAPD-RCV
Specify this subcommand to create an Overlap by Data Set report.

DS(*dsn*)
Use this parameter to specify which data sets you want searched for overlaps. You can specify data set names or name masks.

to *that*
to somewhere else

[INCLEVT(*event name*)]
Specify one or more fully-qualified event names, or one or more event masks to be included in the overlap search. Masked names must conform to the Advanced Backup and Recovery for z/OS extended masking rules.

[EXCLEVT(*event name*)]
Specify one or more fully-qualified event names to be excluded from the overlap search. If you specify this parameter without also specifying the **INCLEVT** parameter, any event except those that match the **EXCLEVT** parameter can be selected. When used with **INCLEVT** parameter, any event that matches the **INCLEVT** parameter and does not match the **EXCLEVT** parameter is selected.

Examples

The following examples show how to initiate Overlap by Data Set Recovery reports in batch mode using procedure BKMBPROC.

```
 //STEPNAME EXEC BKMBPROC
//BKMBATCH.SYSIN DD *
  REPORT OVERLAPD-RCV DS(dsname1)
```

In this example, only one data set is searched for overlaps.

A second method is to code the same JCL but have more than one data set name in the **REPORT OVERLAPD-RCV** command.

```
//STEPNAME EXEC BKMBPROC
//BKMBATCH.SYSIN DD *
  REPORT OVERLAPD-RCV DS(dsname1 dsname2*)
```

In this case, multiple overlap by data set searches are done. Overlaps will be searched for 'dsname1' and any data set matching the mask 'dsname2*'.

The output for the Overlap by Data Set Recovery report can be found under the DD name BKMBREPT in the batch SYSOUT.

REPORT OVERLAPE command syntax

This topic shows the **REPORT OVERLAPE** command syntax that generates the Overlap by Event report, provides parameter descriptions, and shows examples of its usage.

You can use either the + symbol or the - symbol as a line continuation character. Make certain that you leave at least one blank space at the end of the line, between the last parameter and the continuation symbol.

REPORT OVERLAPE{EV(*evname1 ...evnamen*|*evnamemask1 ...evnamemaskn*)|
LOGAGG(*logaggdsn*)}GEN(*nnnn*)[COMPEVT(*fromgen,togen*)]

[COMPEVT(*fromgen,togen*)]

[COMPGEN(*fromgen,togen*)]

[PREVENT_BACKLEVEL]

Parameters

OVERLAPE
Specify this subcommand to indicate that you want an Overlap by Event report.

EV(*evname1 ...evnamen*|*evnamemask1 ...evnamemaskn*)
Use this parameter to specify the ABARS backup event names or name masks that you want searched for data set overlaps. You must specify either the **EV** or **LOGGAGG** parameter.

GEN(*nnnn*)
Use this parameter to specify the relative generation of the backup event you want searched for data set overlaps. Your entry must be an integer in the range 0 to -9999, inclusive.

LOGAGG(*logaggdsn*)
Use this parameter to specify the logical aggregate events you want searched for data set overlaps. You cannot specify a mask for this parameter. The aggregate name you specify can have a maximum length of 30 characters. You must specify either the **LOGAGG** or the **EV** parameter.

[COMPEVT(*evname1 ...evnamen|evnamemask1 ...evnamemaskn)])**]**
Use this parameter to indicate that you want to compare events matching the specified relative generations. Specify an integer in the range 0 to -9999, inclusive.

[COMPGEN(*fromgen,togen*)]
Use this parameter to indicate that you want to compare data set generations matching the specified relative generations. Specify an integer in the range 0 to -9999, inclusive.

- *fromgen* – Specify the starting relative generation of the data sets to be compared to detect overlaps.
- *togen* – Specify the ending relative generation of the data sets to be compared to detect overlaps.

[PREVENT_BACKLEVEL]

Use this parameter to instruct the REPORT OVERLAPE report processor to add its results to the back-level data set.

Examples

The following examples show how to initiate Overlap by Event reports in batch mode using procedure BKMBPROC.

```
//STEPNAME EXEC BKMBPROC
//BKMBATCH.SYSIN DD *
  REPORT OVERLAPE EV(evname1)   +
                  GEN(0)
```

In this example, only one event is searched for overlaps. A second method is to code the same JCL, but have more than one event name in the **REPORT OVERLAPE** command.

```
//STEPNAME EXEC BKMBPROC
//BKMBATCH.SYSIN DD
  REPORT OVERLAPE EV(evname1 evname2*)  -
                  GEN(0)                -
                  COMPGEN(-5,-10)
```

In this case, multiple overlap by event searches are done. Overlaps are searched using data set names from evname1 and data sets belonging to any event matching the mask evname2*.

The output for the Overlap by Event report can be found under the DD name BKMBREPT in the batch SYSOUT.

REPORT DSNLIST command syntax

This topic shows the **REPORT DSNLIST** command syntax that generates the Data Set List report, provides parameter descriptions, and shows examples of its usage.

You can use either the + symbol or the - symbol as a line continuation character. Make certain that you leave at least one blank space at the end of the line, between the last parameter and the continuation symbol.

REPORT DSNLIST DS(*dsn1 ...dsnn* | *dsnmask1 ...dsnmaskn*)

[DATE(*fromdate,todate* | *fromdatemask,todatemask*)]

[EV(*evname1 ...evnamen* | *evnamemask1 ...,evnamemaskn*)]

[GEN(*fromgen,togen*)]

Parameters

DSNLIST
Use this parameter to indicate that you want a Data Set List report generated.

DS(*dsn1 ...dsnn* | *dsnmask1 ...dsnmaskn*
Use this parameter to specify the names or name masks of the data sets you want listed in the data set list report.

[DATE(*fromdate,todate* | *fromdatemask,todatemask*)]
Use this parameter to specify a start and end date for the data sets or events you want included in the report.

- *fromdate* or *fromdatemask* – Specify the starting date or mask of the data sets or events to be listed in the data set list report. Dates must be in the format YYYY/MM/DD or MM/DD/YYYY.
- *todate* or *todatemask* – Specify the ending date or mask of the data set(s) or event(s) to be listed in the data set list report.

For relative date masking, use * for today, *-1 for yesterday, *+*nnn* for today minus a number of days (up to 999), and *-*nnn* for *nnn* days ago. Example: *-5 will search for five days back from today.

[EV(*evname1 ...evnamen* | *evnamemask1 ...,evnamemaskn*)]
Specify any number of event names or masks of events you want listed in the data set list report.

[GEN(*fromgen,togen*)]
Use this parameter to specify range of relative generations of data sets or events you want listed in the report.

- *fromgen* – Specify the starting relative generation (nnnn) of the data sets or events to be listed in the data set list report.
- *togen* – Specify the ending relative generation of the data sets or events to be listed in the data set list report.

Examples

The following examples show how to initiate Data Set List reports in batch mode using procedure BKMBPROC.

```
//STEPNAME EXEC BKMBPROC
//BKMBATCH.SYSIN DD *
  REPORT DSNLIST DS(dsname1)
```

In this example, only one data set is searched for.

A second method is to code the same JCL, but have more than one event name in the REPORT DSNLIST command.

```
//STEPNAME EXEC BKMBPROC
//BKMBATCH.SYSIN DD *
  REPORT DSNLIST DS(dsname*)             -
```

```
  EV(evname1 evname2*)     -
 GEN(0)                    -
DATE(*,2011/03/31)
```

In this case, any data set matching *dsname** in events *evname1* or *evname2* from generation 0, between 2011/03/31 and today, is displayed. The output for the Data Set List report can be found under the DD name BKMBREPT in the batch SYSOUT.

REPORT EVTLIST command syntax

This topic shows the **REPORT EVTLIST** command syntax that generates the Event List report.

Multiple lines may be continued using either the ' + ' or ' - ' characters by leaving at least one blank after the parameter value. Continued lines may start in any position.

REPORT EVTLIST EV(*evname ...evnamen*|*evnamemask1 ...evnamemaskn*)|LOGAGG(*logaggdsn*)

[DATE(*fromdate,todate*|*fromdatemask,todatemask*)]

[GEN(*fromgen,togen*)]

[TIME(*fromtime,totime*)]

Parameters

EVTLIST
Specify this subcommand to indicate that you want an Event List report generated.

EV(*evname ...evnamen*|*evnamemask1 ...evnamemaskn*) *that*
Use this parameter to specify the names or name masks of the events you want listed in the Event List report. If you specify this parameter, you cannot also specify the **LOGAGG** parameter.

LOGAGG(*logaggdsn*)
Use this parameter to specify the logical aggregate data set name of the logical aggregate events you want listed in the report. The name you specify must be an absolute name, rather than a mask. The name you specify can be a maximum of 30 characters in length.

If you specify this parameter, you cannot also specify the **EV** parameter.

[DATE(*fromdate,todate*|*fromdatemask,todatemask*)]
Use this parameter to specify a start and end date for the events you want included in the report.

- *fromdate* or *fromdatemask* – Specify the starting date or mask of the events to be listed in the data set list report. Dates must be in the format YYYY/MM/DD or MM/DD/YYYY.
- *todate* or *todatemask* – Specify the ending date or mask of the events to be listed in the data set list report.

For relative date masking, use * for today, and *-*nnn* for today minus a number of days, up to 999. Example: *-5 will search for five days back from today.

[GEN(*fromgen,togen*)]
Use this parameter to specify a range of relative generations of events you want listed in the report.

- *fromgen* – Specify the starting relative generation (nnnn) of the events to be listed in the event list report.
- *togen* – Specify the ending relative generation of the events to be listed in the event list report.

[TIME(*fromtime,totime*)]
Use this parameter to specify a time range for the events you want listed in the report.

- *fromtime* – Specify the starting time of the events you want listed in the report. Use the 24-hour clock format, HHMM.
- *totime* – Specify the ending time of the events you want listed in the report. Use the 24-hour clock format, HHMM.

Examples

The following examples show how to initiate Event List reports in batch mode using procedure BKMBPROC.

```
//STEPNAME EXEC BKMBPROC
//BKMBATCH.SYSIN DD *
  REPORT EVTLIST EV(evname1)
```

In this example, the search is for one event only. If found, the event is displayed with a space format of Kilobytes and two decimal places.

A second method is to code the same JCL but have more than one event name in the **REPORT EVTLIST** command.

```
//STEPNAME EXEC BKMBPROC
//BKMBATCH.SYSIN DD *
  REPORT EVTLIST EV(evname*)                -
                 GEN(0)                     -
                 DATE(*,2001/02/29)
```

In this case, any event matching *evname** from generation 0, between today and 2001/02/29, are included in the report.

```
//STEPNAME EXEC BKMBPROC
//BKMBATCH.SYSIN DD *
  REPORT EVTLIST EV(evname1)                -
                 DATE(2006/07/05,2006/07/06) -
                 TIME(0900,1000)
```

In this example, only one event is searched for, between the dates of 2006/07/05 and 2006/07/06, and between the times of 0900 and 1000. If found, the event is displayed with a space format of Kilobytes and two decimal places. The output for the Event List report can be found under the DD name BKMBREPT in the batch SYSOUT.

REPORT TAPEPULL command syntax

This topic shows the **REPORT TAPEPULL** command syntax that generates the Tape Pull List report, provides parameter descriptions, and shows examples of its usage.

You can use either the + symbol or the - symbol as a line continuation character. You must leave at least one blank space at the end of the line, between the last parameter and the continuation symbol.

REPORT TAPEPULL EV(*evname1 ...evnamen*|*evnamemask1 ...evnamemaskn*)|LOGAGG(*logaggdsn*)

[DATE(*fromdate,todate*|*fromdatemask,todatemask*)]

[GEN(*fromgen,togen*)]

[INCLABARS]

[INCLACCOMP]

[SORTBYNAME]

[SORTBYVOL]

[SORTBYSLOT]

[VOLUMEONLY]

Parameters

TAPEPULL
Specify this subcommand to indicate that you want a tape pull list report generated.

EV(*evname1 ...evnamen*|*evnamemask1 ...evnamemaskn*)
Use this parameter to specify the names or name masks of the tape data sets you want listed in the Tape Pull List report. If you specify this parameter, you cannot also specify the **LOGAGG** parameter.

LOGAGG(*logaggdsn*)
Use this parameter to specify the logical aggregate tape data set name of the logical aggregate events you want listed in the report. The name you specify must be an absolute name, rather than a mask. The name you specify can be a maximum of 30 characters in length.

If you specify this parameter, you cannot also specify the **EV** parameter.

[DATE(*fromdate,todate*|*fromdatemask,todatemask*)]
Use this parameter to specify a start and end date for the events you want included in the report.

- *fromdate* or *fromdatemask* – Specify the starting date or mask of the events to be listed in the tape pull list report. Dates must be in the format YYYY/MM/DD or MM/DD/YYYY.
- *todate* or *todatemask* – Specify the ending date or mask of the events to be listed in the tape pull list report.

For relative date masking, use * for today, and *-*nnn* for today minus a number of days, up to 999. Example: *-5 will search for five days back from today.

[GEN(*fromgen,togen*)]
Use this parameter to specify a range of relative generations of events you want listed in the report.

- *fromgen* – Specify the starting relative generation (nnnn) of the events to be listed in the tape pull list report.
- *togen* – Specify the ending relative generation of the events to be listed in the tape pull list report.

[INCLABARS]
Specify this parameter if you want to search for any ABARS output data sets and add their volumes to the tape pull list report.

Note: If you do not specify **INCLABARS** or **INCLACCOMP**, processing includes both ABARS output data sets and accompany data sets and adds their volumes to the tape pull list report.

[INCLACCOMP]
Specify this parameter if you want to search for any accompany data sets and adds their volumes to the tape pull list report.

Note: If you do not specify **INCLABARS** or **INCLACCOMP**, processing includes both ABARS output data sets and accompany data sets and adds their volumes to the tape pull list report.

[SORTBYNAME]
Specify this parameter if you want the tape pull list report sorted by event name.

Note: This parameter cannot be used if you have specified the **SORTBYVOL** parameter.

[SORTBYVOL]
Specify this parameter to sort the tape pull list by volume.

[SORTBYSLOT]
Specify this parameter generate a slot-based report. This option is valid for CA 1 tape management environments only.

[VOLUMEONLY]
Specify this parameter if you want the tape pull list report to show volume numbers only.

Examples

The following example shows how to initiate a Tape Pull List report in batch mode using procedure BKMBPROC.

```
//STEPNAME EXEC BKMBPROC
//BKMBATCH.SYSIN DD *
  REPORT TAPEPULL EV(evname1)   -
           SORTBYVOL
```

In this example, the search is for one event only. If that event is found, the volumes assigned to that event are displayed sorted by volume. Because **INCLABARS** and **INCLACCOMP** were not specified, both data sets types are included in the report.

A second method is to code the same JCL, but have more than one event name in the **REPORT TAPEPULL** command.

```
//STEPNAME EXEC BKMBPROC
//BKMBATCH.SYSIN DD *
  REPORT TAPEPULL EV(evname1 evname2*)        -
                  GEN(0)                      -
                 DATE(*,2001/02/29)           -
         SORTBYNAME                           -
         INCLACCOMP
```

In this case only ACCOMPANY volumes from any event matching evname1 or evname2* from generation 0, between today and 2001/02/29, are included in the report, which is sorted by event name.

The output for the Tape Pull List report can be found under the DD name BKMBREPT in the batch SYSOUT.

REPORT SPACE command syntax

This topic shows the **REPORT SPACE** command syntax that generates the Disaster Recovery Space Summary report.

Multiple lines may be continued using either the ' + ' or ' - ' characters by leaving at least one blank after the parameter value. Continued lines may start in any position.

REPORT SPACE EV(*evname ...evnamen*|*evnamemask1 ...evnamemaskn*)|LOGAGG(*logaggdsn*)

[DATE(*fromdate*,*todate* | *fromdatemask*,*todatemask*)]

[GEN(*fromgen*,*togen*)]

[BYEVENT]

[SPACE-FORMAT(T|G|M|K)]

Parameters

SPACE
Specify this subcommand to indicate that you want a Disaster Recovery Space Summary report created.

EV(*evname1 ...evnamen*|*evnamemask1 ...evnamemaskn*)
Use this parameter to specify the names or name masks of the events you want listed in the Disaster Recovery Space Summary report. If you specify this parameter, you cannot also specify the **LOGAGG** parameter.

LOGAGG(*logaggdsn*)
Use this parameter to specify the logical aggregate tape set name of the logical aggregate events you want listed in the report. The name you specify must be an absolute name, rather than a mask. The name you specify can be a maximum of 30 characters in length.

If you specify this parameter, you cannot also specify the **EV** parameter.

[DATE(*fromdate*,*todate* | *fromdatemask*,*todatemask*)]
Use this parameter to specify a start and end date for the events you want included in the report.

- *fromdate* or *fromdatemask* – Specify the starting date or mask of the events to be listed in the disaster recovery space summary report. Dates must be in the format YYYY/MM/DD or MM/DD/YYYY.
- *todate* or *todatemask* – Specify the ending date or mask of the events to be listed in the disaster recovery space summary report.

For relative date masking, use * for today, and *-*nnn* for today minus a number of days, up to 999. Example: *-5 will search for five days back from today.

[GEN(*fromgen*,*togen*)]
Use this parameter to specify a range of relative generations of events you want listed in the report.

- *fromgen* – Specify the starting relative generation (nnnn) of the events to be listed in the disaster recovery space summary report.
- *togen* – Specify the ending relative generation of the events to be listed in the disaster recovery space summary report.

[BYEVENT]
Specify this parameter if you want the report organized by event, including a grand total. If you omit this parameter, the report includes only the grand total.

[SPACE-FORMAT(T|G|M|K)]
Use this parameter to indicate the unit you want used for space amounts shown in the report.

- T – Display the space usage in terms of terabytes.
- G – Display the space usage in terms of gigabytes.
- M – Display the space usage in terms of megabytes.
- K – (Default) Display the space usage in terms of kilobytes.

Examples

The following example shows how to initiate a SPACE report in batch mode using procedure BKMBPROC.

```
//STEPNAME EXEC BKMBPROC
//BKMBATCH.SYSIN DD *
  REPORT SPACE EV(evname1)
```

In this example, the search is for a single event. If that event is found, the space information for that event is displayed, sorted by volume. By default, space usage is shown in terms of kilobytes.

A second method is to code the same JCL, but have more than one event name in the REPORT SPACE command.

```
//STEPNAME EXEC BKMBPROC
//BKMBATCH.SYSIN DD *
  REPORT SPACE EV(evname1 evname2*)      -
               GEN(0)                    -
              DATE(*,2001/02/29)         -          BYEVENT
```

In this case, only events matching evname1 or evname2* from generation 0, between today and 2001/02/29, are displayed. Space totals are shown by event. The output for the SPACE report can be found under the DD name BKMBREPT in the batch SYSOUT.

REPORT BACKUPLIST command syntax

This topic shows the **REPORT BACKUPLIST** command syntax that generates the Backup List Report, provides parameter descriptions, and shows examples of its usage.

You can use either the **+** symbol or the **-** symbol as a line continuation character. Make certain that you leave at least one blank space at the end of the line, between the last parameter and the continuation.

REPORT BACKUPLIST DS(*dsn1 ...dsnn*|*dsnmask1 ...dsnmaskn*)

[DATE(*fromdate*,*todate* | *fromdatemask*,*todatemask*)]

[EV(*evname1 ...evnamen*|*evnamemask1 ...evnamemaskn*)]

[GEN(*fromgen*,*togen*)]

Parameters

BACKUPLIST
Use this parameter to indicate that you want a Backup List Report generated.

DS(*dsn1 ...dsnn*|*dsnmask1 ...dsnmaskn*)
Use this parameter to specify the names or name masks of the data sets you want listed in the Backup List Report generated.

[DATE(*fromdate,todate* | *fromdatemask,todatemask*)]
Use this parameter to specify a start and end date for the data sets or events you want included in the report.

- *fromdate* or *fromdatemask* – Specify the starting date or mask of the data set(s) or event(s) to be listed in the Backup List Report. Dates must be in the format YYYY/MM/DD or MM/DD/YYYY.
- *todate* or *todatemask* – Specify the ending date or mask of the data set(s) or event(s) to be listed in the Backup List Report.

For relative date masking, use * for today, *-1 for yesterday, *+nnn for today minus a number of days up to 999, and *-nnn for nnn days ago. Example: *-5 will search for five days back from today.

EV(*evname1 ...evnamen*|*evnamemask1 ...evnamemaskn*)
Specify any number of event names or masks of events you want listed in the Backup List Report.

[GEN(*fromgen,togen*)]
Use this parameter to specify range of relative generations of data sets or events you want listed in the report.

- *fromgen* – Specify the starting relative generation (nnnn) of the data sets or events to be listed in the Backup List Report.
- *togen* – Specify the ending relative generation of the data sets or events to be listed in the Backup List Report.

Examples

The following example shows how to initiate Backup List Reports in batch mode using procedure BKMBPROC.

```
//STEPNAME EXEC BKMBPROC
//BKMBATCH.SYSIN DD *
  REPORT BACKUPLIST DS(dsname1)
```

In this example, only one data set is searched for.

A second method is to code the same JCL, but have more than one event name in the REPORT BACKUPLIST command.

```
//STEPNAME EXEC BKMBPROC
//BKMBATCH.SYSIN DD *
  REPORT BACKUPLIST DS(dsname*)  -
  EV(evname1 evname2*)  -
              GEN(0)  -
             DATE(*,2011/03/31)
```

In this case, any data set matching dsname* in events evname1 or evname2 from generation 0, between 2011/03/31 and today, is displayed. The output of the Backup List Report can be found under the DD name BKMBREPT in the batch SYSOUT.

Batch ABARS output reporting

When **ABACKUP** or **ARECOVER** requests are initiated either in batch mode through BKMBPROC, or online through ISPF, the final ABARS Manager job or task executed produces an output listing.

The SYSOUT listing contains the following types of information:

- ABARS Manager commands processed log
- ABARS Manager summary
- Aggregate generations summary

ABARS Manager summary

The ABARS Manager summary section of the SYSOUT listing shows the activity log name, data sets processed, storage requirements, data sets created by ABARS, total execution time and expired aggregate generations.

```
ACTIVITY LOG: HSMACT.H1.ABACKUP.AGGNAME1.D99344.T132023
```

The summary output listing also shows a complete list of aggregate data sets that have been successfully VERIFIED, backed up by ABACKUP, or recovered by ARECOVER (for an ARECOVER event). In addition, an indicator (described in the NOTES section at the end of the ABARS Manager summary) is appended to the data set identifying the data set type.

```
*******************************************
* ABACKUP ENDED SUCCESSFULLY:               *
* THE FOLLOWING WERE SUCCESSFULLY BACKED UP: *
*******************************************
   TEST1.X.CNTL                                L0        NONVSAM
   TEST.DSN.ML1                                ML1       NONVSAM
   TEST2.ML2                                   ML2       NONVSAM
```

The BYTES DUMPED BY ABARS TO OUTPUT – USED SPACE section of the ABARS Manager summary information shown below is taken from the DFSMShsm activity log. These are *not* the space values stored in the Advanced Backup and Recovery for z/OS Inventory Data Set (IDS). The DFSMShsm activity logs show used space values for the whole aggregate, and they do not reflect allocated space or individual data set space. ABARS Manager calculates allocated space for each data set and the whole aggregate during preprocessing of the backup. Space values are allocated values in the VTOC and MCDS.

When you select an individual aggregate on the **ABARS Events** panel and then run the N line command (Data Set Name List) for that aggregate, the space shown is allocated space. If you then select an individual data set on the **View Data Set Detail** panel and run the I line command (View Data Set Detail) against that data set, you can view both used and allocated space. Space values from the DFSMShsm activity log and ABARS Manager vary due to the difference between used versus allocated space.

```
*******************************************
* BYTES DUMPED BY ABARS TO OUTPUT - USED SPACE *
*******************************************
   L0=693K ML1=150K ML2=200K TOTAL=1043K
```

The ABARS Manager summary also includes information about the data sets created by ABARS, as shown in the following figure:

```
*******************************************
* OUTPUT DATASET NAMES / VOLSERS            *
*******************************************
   BKM.AGGNAME1.TODISK.C.C01V0003                "C" FILE:
      SMS001
   BKM.AGGNAME1.TODISK.D.C01V0003                "D" FILE:
      SMS001
   BKM.AGGNAME1.TODISK.O.C01V0003                "O" FILE:
      SMS001
```

The total execution time is included in the ABARS Manager summary information, as shown below:

```
EVENT ELAPSED TIME: 00:02:43
```

A list of expired backup generations that were deleted from the IDS is also included in the ABARS Manager summary information. If no expired backup generations are found, a message to that effect appears in the SYSOUT listing, as shown in the following figure:

```
*********************************
*EXPIRED BACKUPS DELETED FROM IDS *
*********************************
   TYPE     DATE      TIME
   **** NONE FOUND ****
```

Aggregate generations summary

The aggregate generations summary section of the SYSOUT listing shows inventory information for all the aggregates processed.

For successful **ABACKUP(EXECUTE)** events only, the aggregate generation summary lists the information for all generation entries (from the IDS) for this aggregate. An example is shown below.

```
*******************************
* BACKUP EVENT VERSION SUMMARY *
*******************************
  "**" = BACKUP INCOMPLETE OR ENDED WITH ERRORS
  "->" = BACKUP ENDED WITH WARNINGS
 GEN #  IGEN # TYPE    DATE    TIME            CFILE NAME                  VOLSER
--------------------------------------------------------------------------------
     0          B   2001/12/10 13:20  BKM.AGGNAME1.TODISK.C.C01V0003
123456
    -1          B   2001/11/29 08:55  BKM.AGGNAME1.TODISK.C.C01V0002
123456
**  -2          B   2001/11/20 10:21  BKM.AGGNAME1.TODISK.C.C01V0001
123456
```

The top Generation line entry is the current entry (from the ABACKUP event just processed). Each entry lists the specific generation number, type of backup, date and time, control file data set name and its associated volumes.

Checking ABACKUP and ARECOVER results

You can determine the outcome of an **ABACKUP** or an **ARECOVER** by checking the final return code, error messages, **ABACKUP** data set space calculations, **ARECOVER** data set usage, and examining changes to the Selection Data Set.

Checking ABACKUP or ARECOVER return codes and error messages

Whenever an **ABACKUP** or **ARECOVER** does not complete successfully, HSM returns a nonzero return code to ABARS Manager, which in turn terminates with a nonzero return code. The ABARS Manager return code is available to subsequent condition code checking job steps.

Any DFSMShsm or DFSMSdss error messages present in the activity log are displayed in an ISPF panel. They are also placed in the Advanced Backup and Recovery for z/OS Inventory Data Set (IDS) with the last aggregate event information. These recent errors are retained in the IDS until the next successful **ABACKUP** or **ARECOVER** event occurs and expires all events greater than the number to retain for that aggregate.

An example of the errors resulting from a failed ARECOVER attempt is shown below.

```
 BROWSE -- ERROR MESSAGES FOR AGGREGATE: AGGNAME1   Line 00000000 Col 001
 080
 Command ===>                                                Scroll ===> HALF
********************************* Top of Data ********************************
ARC6107E NO MIGRATION LEVEL 1 VOLUMES AVAILABLE FOR AGGREGATE GROUP AGGNAME1
USING CONTROL FILE DATASET BKM.AGGNAME1.TODISK.C.C01V0018
 ******************************** Bottom of Data ********************************
```

In the example output listing shown in the preceding example, the original event was an **ARECOVER** with the VEREXE option. Because errors were encountered from DFSMShsm, ABARS Manager aborted the **ARECOVER EXECUTE** command portion.

ABARS Manager ABEND and return codes are derived as follows:

1. First, ABARS Manager looks for any ABARS Manager ABENDs or backup option switches that could terminate the backup or result in a higher return code than the activity log.
2. If all are successful, ABARS Manager looks at the activity log.
3. The backup options take precedence over the activity log unless a higher return code is received from the activity log.

The **BKM_RC_EQUAL_ABARS_RC** parameter found in the BKMINI member of the product parameter library determines which return code is received:

- If **BKM_RC_EQUAL_ABARS_RC** = Y, you receive the return code specified by ABARS in the activity log.
- If **BKM_RC_EQUAL_ABARS_RC** = N:
  - If the data set was bypassed by the ARCBEEXT, ABARS Manager reports it in the ABARS Manager SYSOUT.
  - If you receive a warning message from the ABARS activity log, the return code is 4, unless an override has been included in the **ABARS_RC_OVERRIDE** section of the BKMINI product parameter library member.
  - If you receive an error message from the ABARS activity log (and it was not bypassed by the ARCBEEXT), and the return code is 8 – ABARS Manager issues a return code of 8, unless overridden in the **ABARS_RC_OVERRIDE** section of the BKMINI product parameter library member.

ABARS Manager displays two return codes in the ABARS Manager panels. One shows the return code ABARS produced (shown on the panels as RCA), and the other shows the return code produced from ABARS Manager (shown on the panels as RCB). In some cases, the two return codes can be different. The following examples illustrate how this can occur.

Consider an example in which two different return codes occur when the backup options settings or return code overrides are used. Settings in the `:BACKUP_OPTIONS` section of the Advanced Backup and Recovery for z/OS product parameter library BKMINI member can be modified for an individual aggregate by using line commands (**A** or **C**) for an individual aggregate on the **Aggregate Management with SMS Interface** panel. These backup options can fix category conflicts and other conditions and set a return code that you specify. If the backup options are set to fix any of the categories, and you specify anything other than 0, ABARS gets a zero return code because the error was fixed by ABARS Manager, but the you indicated that a notification should be issued.

If the backup options are all set to N, no conflicts or conditions are corrected and there could be a number of reasons why the return codes differ.

- An error occurred updating the Advanced Backup and Recovery for z/OS Inventory Data Set after a backup or recovery completed.
- An activity log error occurred.
- A postprocessing error occurred.

ABARS Manager issues a U0999 ABEND for any pre- or postprocessing error. For further information about the ABEND, look in the SYSOUT for a description of the error condition.

Note: There is one exception to the U0999 ABEND. When `BKM09897E NO DATA SETS INCLUDED BY SELECTION DATA SETS` is issued, ABARS Manager terminates with a return code of 20. This exception supports Incremental ABARS. A Selection Data Set might be empty because no new or changed data sets were found. A U0999 ABEND invalidates the Incremental backup.

Viewing ABACKUP data set space calculations

Whenever an **ABACKUP(EXECUTE)** successfully completes, ABARS Manager automatically calculates the space for each non-tape data set included in the Data Sets Processed List. Space values are calculated as part of the backup preprocessing and are taken from the allocated values in the VTOC and MCDS.

To view used space information, run the **N** line command against the appropriate aggregate in the list on the **ABARS Events** panel and press Enter.

To view both used and allocated space for an individual data set, run the **I** line command against the appropriate data set in the list on the **View Data Set Detail** panel and press Enter.

Space values from the DFSMShsm activity log and ABARS Manager might be slightly different, due to activity against the data set between the time the **ABACKUP** is submitted and when it ends. The batch output listing includes the space information for PRIMARY, ML1, ML2, and the number of user tape data sets backed up.

About ARECOVER data set space usage

ABARS Manager does not perform the actual backup or recover; it passes the command to ABARS, which in turn uses either DFSMSdss or DFSMShsm. After restoring a data set, allocated space for a data set might not be the same as the original allocated space ABARS Manager recorded.

This difference is normal because when DFSMSdss restores data sets, the condition of the volumes could be much different than when the data sets were originally allocated.

Consider an example in which a VSAM file has extent specifications of cylinders 100,10. If at the primary site the first volume had run out of room after 120 cylinders (primary allocation plus two secondary allocations), when it extends to a second volume, the initial allocation would have been for another 100 cylinders because the primary allocation is always taken for a new volume. So the file, while requiring 130 cylinders, could actually occupy 230 cylinders. In all likelihood, when space is allocated during a restore, the packs are not fragmented–therefore, the file could find space on a single volume and the allocated space is much less than was required for the same file at the primary site.

In situations where the primary allocation is much larger than the secondary allocation, it is possible that at the primary site, the number of extents taken on new volumes could result in far more allocated space than results from a recovery, where fewer volumes are needed due to the difference in how the volumes are loaded.

Selection Data Set handling by ABARS Manager

ABARS Manager was designed to fix category conflicts and correct incorrectly categorized data set situations prior to starting a backup. To remedy these problems, ABARS Manager must manipulate the Selection Data Set in various ways.

A category conflict, such as a tape data set being included in the ALLOCATE list or a DASD data set being included in the ACCOMPANY list, or an incorrectly categorized data set can cause a backup to fail. ABARS Manager uses the backup options you have specified in the BKMINI member of the Advanced Backup and Recovery for z/OS product parameter library and the specifications you made on the product panels for the aggregate being backed up to fix or report on these error conditions.

When you specify any backup option, ABARS Manager makes a copy of your Selection Data Set, compresses it, and stores the it in the Inventory Data Set (IDS). After the original Selection Data Set is stored in the IDS, ABARS Manager offloads the Selection Data Set to expand the masks, check for conflicts or unsupported data set errors that would cause the backup to fail, make the appropriate corrections to the list of data sets to be backed up, and writes the revised list back to the Selection Data Set. It is this revised Selection Data Set that is processed during an **ABACKUP**.

When **ABACKUP** completes successfully, ABARS Manager restores the original Selection Data Set. However, if **ABACKUP** ends abnormally, the Selection Data Set being processed at that point is the revised Selection Data Set. If you subsequently re-attempt the **ABACKUP**, ABARS Manager checks to see if the Selection Data Set is in its original format, and if it is not, restores it to the original Selection Data Set prior to processing. If, after an unsuccessful **ABACKUP**, you want to restore the original Selection Data Set so you can make changes to it before re-attempting the backup, run the **VS** line command against the appropriate aggregate from the **ABARS Events** panel. When the **SDS(s) Used Before/During Backup** panel opens, you can type E next to the Selection Data Set you want to view. When the ISPF Edit session displays the Selection Data Sets, use the **SAVE** command. If you made no changes on the ISPF Edit panel, ABARS Manager restores the original Selection Data Set; otherwise, your changes to the Selection Data Set are saved.

Appendix F. RSP operator commands

RSP (BKMSTSKR) collects information about application data set usage, through standard system exits. SMF data is captured using IEFU83 and JCL data is captured using IEFUJI.

Starting the RSP started task

You can start the RSP started task and all individual subtasks or you can start the RSP started task without starting the individual subtasks.

- To start RSP and the individual subtasks, issue the console command:

  F RSP,AUTOSTART

  The **AUTOSTART** command issues these commands:

  ```
  F RSP,U83(LOAD)
  F RSP,UJI(LOAD)
  F RSP,DSPC(CREATE)
  F RSP,U83(TURNON)
  F RSP,UJI(TURNON)
  F RSP,ACDIUPDT(START)
  F RSP,OVFIMTSK(START)
  F RSP,OVFEXTSK(START)
  F RSP,BKMSTSKR(START)
  ```

  The **AUTOSTART** command starts the RSP started task and all individual subtasks that are required for RSP to track job names. Typically, **AUTOSTART** is the only command needed to start BKMSTSKR. (The procedure name is BKMSTSKR and the started task name is 'RSP'.) All modify commands are based on the identifier 'RSP'.

  Note:

  – It is recommended that you start the started task, RSP, and associated subtasks at the same time using the **AUTOSTART** command.
  – The only variable within the structure is the data space size. The number of buffers is controlled by the DSPC_BUFFERS token in the ACDI_RSP section of the BKMINI member.

    The DSPC_BUFFERS token specifies the maximum number of buffers RSP should use. If the DSPC_BUFFERS token is set to a number less than 5000, or if it is not present, RSP defaults to 5000 buffers. RSP uses half of the buffer space (about 2500 buffers) immediately and saves the remaining space for buffer space increases.

- You can start the started task using the JOBNAME= parameter. When starting the BKMSTSKR, the target PROCLIB member name must be specified on the start command. (The procedure name is BKMSTSKR and the started task name is RSP.) All modify commands are based on the identifier RSP. To start the SMF collection main task and associated subtasks at the same time, use one of these commands:

  S BKMSTSKR,JOBNAME=RSP,PARM=AUTOSTART

  or

  S BKMSTSKR,JOBNAME=RSP,PARM=A

  Note: If the AutoStart token in the :ACDI_RSP section of the BKMINI member is set to YES, the command S BKMSTSKR,JOBNAME=RSP automatically performs an **AUTOSTART**.

- To start the started task without the individual subtasks, issue the command: S BKMSTSKR,JOBNAME=RSP

  Each individual subtask is *not* started or loaded and therefore, you need to start the individual subtasks separately.

Stopping the RSP started task

This information describes how to stop the RSP started task.

- Issue the following command to stop the RSP started task: **P RSP**

  This command stops the RSP structure and terminates BKMSTSKR using these commands:

  ```
  F RSP,ACDIUPDT(STOP)
  F RSP,OVFIMTSK(STOP)
  F RSP,OVFEXMTSK(STOP)
  F RSP,RSPMTASK(STOP)
  F RSP,U83(TURNOFF)
  F RSP,DSPC(DELETE)
  F RSP,U83(UNLOAD)
  ```

RSP Monitor task commands

Use these commands to start or stop RSP Monitor.

Issue these commands using the format: `F RSP,BKMTSKR(...)`

START
Starts the RSP Monitor task.

For example: `F RSP,BKMTSKR(START)`

STOP
Stops the RSP Monitor task.

For example: `F RSP,BKMTSKR(STOP)`

IEFU83 commands

These commands should only be used at the direction of technical support.

The IEFU83 command syntax is: `F RSP,U83(...)`

LOAD
Acquires ECSA space, loads the primary and secondary exits, and places 'BKM9254Z' in the SYS.IEFU83 exit lists. Errors include not being able to obtain ECSA space, unauthorized LOAD libraries, I/O errors during LOAD SVCs and problems with the dynamic exit processor.

Note: You must have IEFU83 defined as an exit within the SMFPRMxx member of SYS1.PARMLIB.

IEFU83 can be added without an IPL.

Example: `F RSP,U83(LOAD)`

UNLOAD
Removes 'BKM9254Z' from the SYS.IEFU83 exit lists and deletes the ECSA space. There are interlocks preventing the UNLOAD when the ACDIUPDT subtask is active and when the data space still exits.

Example: `F RSP,U83(UNLOAD)`

TURNON
Activates the exit in the SYS.IEFU83 exit lists and modifies the primary exit to call the secondary exit. This command starts the SMF data collection. You must be collecting SMF 14, 15, 30, 61, 64, 65, and 66 records. An additional general purpose IEFU83 exit is available for those who want to suppress the logging of these record types, or record types needed by any other IEFU83 exits. If the primary or secondary exit ends abnormally twice, MVS will disable the exit.

Example: F RSP,U83(TURNON)

TURNOFF
Deactivates the exit in the SYS.IEFU83 exit list and modifies the primary exit to not call the secondary exit.

Example: F RSP,U83(TURNOFF)

NEWSECONDARY
In some circumstances, it might be necessary to load a new secondary exit without deleting the data space and its buffers. There are interlocks to prevent this command from working while the exits are active.

To use this command, complete these steps:

1. Copy a new exit (BKMRU83T) to the LOAD library.
2. Quiesce the exits using the **TURNOFF** command.
3. Refresh the code using the **F RSPAMPI,U83(NEWSECONDARY)** command.
4. Activate the exits using a **TURNON** command.

Example: F RSP,U83(NEWSECONDARY)

IEFUJI commands

These commands should only be used at the direction of technical support.

The command syntax is: F RSP,UJI(...)

LOAD
Acquires ECSA space, loads the primary and secondary exits, and places 'BKMUJIPE' in the SYS.IEFUJI exit lists. Errors include not being able to obtain ECSA space, unauthorized LOAD libraries, I/O errors during LOAD SVCs and problems with the dynamic exit processor.

Note: You must have IEFUJI defined as an exit within the SMFPRMxx member of SYS1.PARMLIB.

IEFUJI can be added without an IPL.

Example: F RSP,UJI(LOAD)

UNLOAD
Removes 'BKMUJIPE' from the SYS.IEFUJI exit lists and deletes the ECSA space. There are interlocks preventing the UNLOAD when the ACDIUPDT task is active and when the data space still exits.

Example: F RSP,UJI(UNLOAD)

TURNON
Activates the exit in the SYS.IEFUJI exit lists and modifies the primary exit to call the secondary exit. This command starts the JCL data collection. If the primary or secondary exit ABENDs twice, MVS disables the exit.

Example: F RSP,UJI(TURNON)

TURNOFF
Deactivates the exit in the SYS.IEFU83 exit list, and modifies the primary exit to not call the secondary exit.

Example: F RSP,UJI(TURNOFF)

NEWSECONDARY
Under some circumstances, it might be necessary to load a new secondary exit without deleting the data space (and its buffers). There are interlocks to prevent this command from working while the exits are active.

To use this command, complete these steps:

1. Copy a new exit (BKMUJISE) to the LOAD library.

2. Quiesce the exits using the **TURNOFF** command.
3. Refresh the code with the **F RSPAMPI,UJI(NEWSECONDARY)** command.
4. Activate the exits with a **TURNON** command.

Example: `F RSP,UJI(NEWSECONDARY)`

ACDIUPDT task commands

Use these commands to operate start, stop, or remove the ACDIUPDT task.

The command syntax is: `F RSP,ACDIUPDT(...)`

START
Starts the ACDIUPDT task. Errors include ATTACH SVC problems or VSAM data set problems. ATTACH SVC problems will appear on the console while VSAM errors will appear in the BKMSTSKR SYSLOG file.

Example: `F RSP,ACDIUPDT(START)`

STOP
Stops the ACDIUPDT task. If the task ends abnormally, BKMSTSKR will issue a `task not responding` message. In this case, force the task using a **KILL** command.

Example: `F RSP,ACDIUPDT(STOP)`

KILL
If the task will not stop, and you cannot start a new one, you must stop the existing task using the **KILL** command. BKMSTSKR will prompt the operator to confirm the forced removal of the task before performing the operation.

Example: `F RSP,ACDIUPDT(KILL)`

Overflow task commands

Use these commands to start or stop overflow task commands.

Issue these commands using the appropriate format:

- `F RSP,OVFEXTSK(...)` — export overflow task.
- `F RSP,OVFIMTSK(...)` — import overflow task.

START
Starts the import or export overflow task.

For example: `F RSP,OVFIMTSK(START)` starts the import overflow task.

STOP
Stops the import or export overflow task.

For example: `F RSP,OVFEXTSK(STOP)` stops the export overflow task.

Appendix G. Sizing the Advanced Backup and Recovery for z/OS data sets

This appendix provides the information necessary to estimate required Advanced Backup and Recovery for z/OS Inventory Data Set (IDS) and Automated Critical Data Identification database file sizes, as well as important information regarding compression of the IDS.

A common IDS is used for ABARS Manager and Critical Backup Tracking and Inventory. All references to the IDS refer only to data stored in the inventory data set for ABARS Manager and for the Critical Backup Tracking and Inventory feature.

The Automated Critical Data Identification feature has its own database. This database is a VSAM KSDS file structure that is used to maintain and track application information.

All product data sets for Advanced Backup and Recovery for z/OS are distributed as partitioned data sets (PDSs). Product load modules are provided in a single library, which must be authorized using APF (Authorized Program Facility).

Note: Advanced Backup and Recovery for z/OS does not honor your system parameter library member IEASYS00 parameter LNKAUTH=LNKLST, if it has been specified. The product load library needs to be APF authorized.

Sizing suggestions for the product libraries

Use this reference information to help you plan for and manage the Advanced Backup and Recovery for z/OS Inventory Data Set (IDS): estimating the size of the IDS, work data set allocation, and backing up, reorganizing, or resizing the IDS.

As a starting point for this information, it might be helpful to know the approximate sizes (shown as cylinders of 3390-equivalent device capacity) of the Advanced Backup and Recovery for z/OS product libraries:

- ISPPLIB–2 cylinders
- JCL library–2 cylinders
- Product load library–15 cylinders
- Product parameter library–2 cylinders
- IDS–500 cylinders
- Automated Critical Data Identification database–10 cylinders
- Skeleton library–1 cylinder

Estimating the size of the IDS

While the size of a typical production environment IDS file is approximately 500 cylinders, you can calculate a more precise file size estimate by using the criteria described in this topic.

The criteria and estimated space calculations shown in the following table are based on backup tracking only:

Aggregates	Data sets	Copyset	Versions	Estimated space
1	1000, no GDGs	1	2, no errors	858 KB
100	1000 each, no GDGs	1	2, no errors	151.6 MB
500	500 each, no GDGs	2	2, no errors	103.1 MB

Where:

- *Copyset* means the number of ABARS-generated duplicate copies of each aggregate. These backups use separate tapes and are created at the same time as the primary aggregate output.
- *Versions* represents the generation number of an ABARS aggregate. The more versions you track, the more storage they require. The depth of version tracking is tied automatically to the RETAINVERSIONS parameter specified in the ABARS Aggregate Group Definition in ISMF.
- *Estimated space* values are based on the following calculations:
  1. (Number of aggregates * 170 bytes) + 32758 bytes for each aggregate that has SDSL statements
  2. (Number of versions per aggregate * Number of aggregates) = Number of backup event records
  3. (Number of backup event records * 102,779 bytes) = Total amount of space for all backup event records
  4. Add 655 bytes for each DSN within each backup event.
  5. Add 11276 bytes for each GDG within each backup event.
  6. Add 32758 bytes for each event that contains errors.
  7. Add 1380 bytes for each additional copyset.

Work data set allocation

Temporary online work data sets are allocated from 1 track to 1 cylinder, depending on the amount of data.

Backing up, reorganizing, or resizing the IDS

The Advanced Backup and Recovery for z/OS Inventory Data Set (IDS) should be backed up on a regular basis, reorganized regularly, and resized as the inventory data set grows.

Schedule these processes when job tracking is experiencing a slow period. During this time, Critical Backup Tracking and Inventory continues to track backups and create flat files with the pertinent backup information. The flat files created during this time are used to update the IDS after the flat copy processor has been started again.

To resize the IDS, follow these steps:

1. Issue the F `xxx,STOPFC` command to stop the flat copy processor, where xxx is the name (BKMSTSKA) of the Critical Backup Tracking and Inventory started task.

   If a copy is in progress when you issue the command, it runs to completion and then shuts down.
2. Run an `IDCAMS REPRO` to an output file to back up the IDS.
3. Delete and redefine the IDS, using member `BKMDEFDB` in the product JCL library.

   If you anticipate that the IDS could become larger than 4 GB, you must use extended addressing. See the *Sizing the Advanced Backup and Recovery for z/OS Data Sets* appendix for information about estimating the size of the IDS.
4. **REPRO** (the previously REPROed IDS) back into the new IDS.
5. Issue the F `xxx,STARTFC` command to start the flat copy processor, where xxx is the name (BKMSTSKA) of the Critical Backup Tracking and Inventory started task.

Sizing the Automated Critical Data Identification database

This appendix provides the information necessary to estimate the required Automated Critical Data Identification database file size. The estimate is based on the number and size of applications to be tracked by Automated Critical Data Identification for disaster recovery backup assurance.

Sizing of the Automated Critical Data Identification database at installation time is important. Equally important is the monitoring of the database over time. Because the database is updated by RSP in real time, under-allocation can create operational problems later, when RSP attempts to add records to a file that is unable to accept them. You can use Automated Critical Data Identification statistics collected over

time to refine your estimate periodically to ensure that Automated Critical Data Identification remains an effective tracking tool for your disaster recovery assurance processes.

This appendix provides the following reference information to help you estimate the size of your database:

- Two formulas for estimating your Automated Critical Data Identification database
- Explanations of why out-of-space conditions occur and suggestions for preventing them
- Suggestions for setting up database monitoring and a database recovery plan
- A procedure for resizing your Automated Critical Data Identification database

Estimating the size of your Automated Critical Data Identification database

Use the estimation methods described in this topic to estimate the size of your Automated Critical Data Identification database. These methods can be used to ensure that the database can grow when appropriate.

Because the RSP tasks write SMF activity information directly to the Automated Critical Data Identification database, the database must be sized correctly and monitored for potential trouble. On the other hand, the database should not be reorganized unnecessarily. A high number of splits might not be of concern, nor should a high number of extents, unless the VSAM upper limit of 123 extents is being approached.

Like any VSAM KSDS, the potential for 123 extents should be exploited. Keeping the secondary space request relatively small (assuming that a significant number of extents might be allocated) increases the likelihood of finding space on a fragmented volume. Also, provide candidate volumes when allocating the database. This permits the file to grow, if necessary, to available space on another volume, ensuring the availability of the RSP tracking function.

Do not allocate in quantities other than CYLINDERS.

Do not allocate less than 10 cylinders primary and 10 cylinders secondary (`CYL (10 10)`). As applications are added, this might require adjustment.

Quick estimate method

The simplified formula shown below permits you to establish a reasonable allocation quantity quickly. If you need a more precise estimate, use the worksheets provided in the following section.

Primary space (in bytes) = (n * (j + d) * 400®) + 16384

Primary space (in cylinders) = ((primary space in bytes) / 552960), rounded up to the next integer

Consider the following example, in which:

- n is 140 applications
- j is 30 jobs per application (on the average)
- d is 275 data sets per application

Given these values of n, j, and d:

Primary space required (in bytes) = (140 * (30 + 275) * 400) + 16384 = 17,096,384 bytes

Primary space required (in cylinders) = 17,096,384 bytes / 552960 bytes per cylinder = 30.92 cylinders

30.92 cylinders rounded to the next integer = 31 cylinders

The suggested allocation for this example situation would be `CYLINDERS(31 10)`.

Keep in mind the following guidelines:

- Assign at least one VSAM CANDIDATE VOLUME to the data set allocation.
- Specify `FREESPACE (20 20)`.
- Allocate in terms of cylinders only.

- Do not allocate less space than `CYL(10 10)`.
- View PARMLIB member BKMDB01 for all allocation parameters.

Detailed estimate method

Record Type	CI%			CI Total	Description
CONTROL	0.25	x		0.25	Static record
			Number of images sharing the database		
RSP STATS	0.34	x	_____	_____	
Filters:			Number defined		
AEF	0.05	x	_____	_____	Appl external
AJF	0.04	x	_____	_____	Appl job
ARF	0.04	x	_____	_____	Appl retention
ASF	0.04	x	_____	_____	Appl evaluation
GRD	0.05	x	_____	_____	Global retention dsn
GRS	0.05	x	_____	_____	Global retention SYSID
GRU	0.05	x	_____	_____	Global retention unit
GRV	0.05	x	_____	_____	Global retention VOLSER
GSD	0.05	x	_____	_____	Global evaluation DSN
GSS	0.05	x	_____	_____	Global evaluation SYSID
GSU	0.05	x	_____	_____	Global evaluation unit
GSV	0.05	x	_____	_____	Global evaluation VOLSER
URF	0.05	x	_____	_____	Universal retention
USF	0.05	x	_____	_____	Universal evaluation
			Number of applications		
AMF	0.04	x	_____	_____	Application filter info
APF	0.04	x	_____	_____	Application filter control
APL	0.34	x	_____	_____	Application profile
APR	0.25	x	_____	_____	Application resource
APV	0.17	x	_____	_____	Application verification
CLM	0.34	x	_____	_____	CONTROL-M Only
EBI	0.15	x	_____	_____	External backup info
JBC non-OPC	0.15	x	_____	_____	Job Collection
JBC OPC	0.20	x	_____	_____	
JCX	0.25	x	_____	_____	JCS Matrix Only
MSQ	0.25	x	_____	_____	Message queue
			MaxLOGMsgs from BKMINI		
LME	0.08	x	_____	_____	Log events

Calculate the following for each application:

Record type	CI%					CI Total	Description
			DSNs from JCL capture		JCL cycles		
DSD	0.14	x	_____	x	_____	_____	Detail data set
			DSNs from RSP SMF capture		RSP cycles		
	0.14	x	_____	x	_____	_____	
			DSNs from Batch SMF scan		SMF cycles		
	0.14	x	_____	x	_____	_____	
			DSNs from RSP JCL capture				
DSR	0.125	x	_____			_____	Data set
			DSNs from RSP SMF capture or Batch SMF scan				
	0.125	x	_____			_____	
			DSNs generated by INCL_EXTDSN				
	0.125	x	_____			_____	
			DSNs generated by ACCOMP_EXTDSN				
	0.125	x	_____			_____	
			DSNs generated by ALLOC_EXTDSN				
	0.125	x	_____			_____	

Note: GDG options might increase the number of DSRs in the database. 'Expand base GDG' adds a DSR for each generation associated with the base name. 'BOTH', for absolute and relative DGD names, adds two DSR records for each generation data set. 'Minimum for rerun' or 'Include all' might add DSRs for generation data sets.

			Additions for GDGs			
	0.13	x	_____		_____	

Calculate the following for each application:

Record type	CI%			CI total	Description
			Number of jobs		
JBE	0.1	x	_____	_____	Job entries + JAS
			Number of applications with alternate SCHID		
SDI	0.03	x	_____	_____	CA7 only
			Total CIs	_____	
			(Total CIs) / 120	_____	= Primary cylinders

The number of CIs is based on the following assumptions:

- The database was defined on a 3390 device.
- The database space allocation was in terms of cylinders.
- The database has CISZ of 4096.
- The database has 20% free space per CI.

- The database has 20% free space per CA.

Preventing the database from running out of space

If the Automated Critical Data Identification database cannot extend when necessary, the RSP VSAM subtask shuts down. Eventually, the RSP buffers fill up and data is lost. If you catch the problem early, you can fix it in real time or correct it so it causes no significant problems. This topic provides suggestions for preventing out-of-space situations.

Preventive measures

The following preventive measures are recommend to reduce the likelihood that your Automated Critical Data Identification database will experience out-of-space situations:

- To minimize RSP subtask downtime, create the JCL and procedures in advance so you can quickly rebuild the database if it becomes full and cannot extend.
- Trap message BKM1010E to detect RSP subtask shutdowns. These shutdowns can signify a problem with the database.
- Monitor the database size and allocate more space or reorganize the database when necessary.
- To detect the loss of vital records, use the Automated Critical Data Identification Monitor to keep track of the data space size. Tracking the data space provides documentation in cases where the data space size is not large enough to meet the requirements of your data center.
- To avoid situations where data is being written continually to the database from an application that is unready to implement in your production environment:
  - Execute BKMAPLEN regularly.
  - Ensure that the RSPCOLLECT set-up option is set to NO.
- Use the database sizing formulas to determine if the current database size is commensurate with the required size.

Set RSP data space size

The recovery procedure must be accomplished in a reasonable amount of time. Have your JCL or procedure to rebuild the database in place ahead of time to minimize the RSP subtasks downtime. If the RSP data space on each image is not large enough to hold the SMF records being captured while the subtasks are stopped, the SMF records will be exported to the overflow file.

The data space size is set by the BKMINI RSP keyword DSPC_BUFFERS. To obtain a worst-case estimate of how many data space buffers you need, examine the number of SMF type 14, 15, 30, 61, 64, 65, and 66 records generated for a time interval. Estimate the time it will take to accomplish the recovery procedure steps, and from this determine approximately how many SMF records might accumulate in the data space. Keep in mind that the data space needs to hold SMF records for only the application jobs being tracked. The total SMF records of the types in concern will likely be much greater than the number actually collected by RSP.

Each data space buffer is approximately 400 bytes, therefore a request for 10000 buffers requires a data space of 4 megabytes.

Capture the BKM1010E message

The Automated Critical Data Identification database is updated continuously by RSP. If the database runs out of room or other problems occur, RSP issues message BKM1010E, indicating that updates to the database have been suspended. RSP then begins buffering data set activity information, but does not attempt to write the data to the database. After any fault is corrected, updates to the database must be restarted manually.

Set up the reallocation JCL in advance

If you take action quickly, you can resize the database and restart the update process without loss of activity data. The JCL procedure BKMSIZE provides the framework.

Monitor database space usage

While the actual amount of space required for the Automated Critical Data Identification database varies, depending upon the number of data sets and jobs that are to be tracked by Automated Critical Data Identification, you can use the quick estimate formula or the detailed estimate formula to calculate the approximate space requirements for the Automated Critical Data Identification database.

Be aware that a new option to retain extra cycles of SMF and JCL data might require a larger database. At present, RSP writes directly to the database from each contributing image. Although a mechanism exists to allow a database reorganization or rebuild with no data loss, you need to ensure that the database is sized correctly and that candidate volumes are available if space might not be available on the current volume.

Resizing the Automated Critical Data Identification database

Use the procedure provided in this topic to reorganize or enlarge the Automated Critical Data Identification database when Automated Critical Data Identification is already in use.

Depending on RSP tracking activity, it is likely that you have several hours in which to perform this procedure before a loss of tracking information is experienced. Automated Critical Data Identification uses the overflow file during this process.

If you anticipate that the Automated Critical Data Identification database could become larger than 4 GB, you must use extended addressing.

1. Issue the command F *rsptaskname*,ACDIUPDT(STOP) to stop the VSAM subtask on each image of the complex that shares a single Automated Critical Data Identification database. If the database has already run out of room, be aware that the VSAM subtasks might have stopped spontaneously; however, issuing command in that situation does no harm.
2. End all batch or ISPF access to the database.
3. Run BKMSIZE to alter the size of the Automated Critical Data Identification database:
   a) Make all necessary updates to JCL member BKMSIZE.
   b) Update product parameter library member BKMDB02 to provide an alternative database name, DATA name, and INDEX name. These updates effectively rename the production Automated Critical Data Identification database.
   c) Update product parameter library member BKMDB01 to reallocate the production database with a more appropriate space allocation. Keep in mind that you need to assign at least one VSAM CANDIDATE VOLUME to the data set allocation, you must specify FREESPACE (20 20), and that your allocation should be in terms of cylinders and no less than CYL (10 10).

      Note: If you anticipate that the Automated Critical Data Identification database could become larger than 4 GB, you must use extended addressing.
   d) Update product parameter library member BKMDB03 to provide the alternative database name, thereby copying the alternative database to the newly reallocated production database.
   e) Update product parameter library member BKMDB04 to provide the alternative database name, thereby deleting the alternative database.
4. On all OS/390 images where RSP is active, issue the command F *rsptaskname*,ACDIUPDT(START) to restart the VSAM subtasks you stopped in step 1.
5. Resume ISPF and batch usage.
6. Use the RSP Monitor feature to view the number of buffers in use. Verify that the number is decreasing. Also look at the number of records dropped to determine if the resizing was done in time.

Appendix H. Inventory data set administration

Backing up, reorganizing, and resizing the Inventory Data Set

The Advanced Backup and Recovery for z/OS Inventory Data Set should be backed up on a regular basis, reorganized regularly, and resized as the inventory data set grows.

Schedule these processes at a slow time of backup job tracking and ABARS Manager activity. During this time, Critical Backup Tracking and Inventory continues to track backups and create flat files with the pertinent backup information. The flat files created during this time will be used to update the inventory data set after the flat copy processor has been restarted. Use the following procedure to resize the IDS.

1. Stop the flat copy processor by issuing the following command: F *xxx*,STOPFC where *xxx* is the name (BKMSTSKA) of the Critical Backup Tracking and Inventory started task. Perform this step for all active Critical Backup Tracking and Inventory started tasks that share the IDS. If a flat copy is in progress, it will shut down after it is complete.
2. Back up the Advanced Backup and Recovery for z/OS Inventory Data Set by running an IDCAMS REPRO to an output file.
3. Delete and redefine the Advanced Backup and Recovery for z/OS Inventory Data Set using member BKMDEFDB of the product JCL library.

   Note: If you anticipate that the IDS could become larger than 4 GB, you must use extended addressing. See the *Sizing the Advanced Backup and Recovery for z/OS Data Sets* appendix for information about estimating the size of the IDS.
4. REPRO (the previously REPROed Advanced Backup and Recovery for z/OS Inventory Data Set) into the new Advanced Backup and Recovery for z/OS Inventory Data Set.
5. On the appropriate LPARs, start the flat copy processor by issuing the command: F *xxx*,STARTFC where *xxx* is the name (BKMSTSKA) of the Critical Backup Tracking and Inventory started task.

Backing up and reorganizing the Inventory Data Set

You can use Advanced Backup and Recovery for z/OS Dynamic Data Set Backup/Reorganization utility to back up or reorganize (or both) the Inventory Data Set (IDS) without stopping the product programs that use the IDS: ABARS Manager and Critical Backup Tracking and Inventory.

If you need to make a backup when the IDS is in use by Critical Backup Tracking and Inventory or ABARS Manager (or both), use the Dynamic Data Set Backup/Reorganization utility, rather than other utilities (such as DFDSS, FDR, and so on). Using other utilities for backup might cause IDS serialization issues.

About the Dynamic Data Set Backup/Reorganization utility

The Dynamic Backup/Reorganization utility pauses ABARS Manager and Critical Backup Tracking and Inventory so a clean backup or reorganization can be run against the Advanced Backup and Recovery for z/OS IDS.

Without the Dynamic Backup utility, attempts to backup the IDS would cause ABARS Manager and Critical Backup Tracking and Inventory to fail because the backup has an enqueue on the IDS, which prevents the product programs from opening and updating the IDS.

The Dynamic Reorganization utility pauses ABARS Manager and Critical Backup Tracking and Inventory programs in progress until the IDS backup or reorganization is complete.

Note: Dynamic Data Set Backup/Reorganization applies to documented VSAM data sets used by Advanced Backup and Recovery for z/OS product programs only.

Dynamic Data Set Backup/Reorganization usage considerations

To use this utility effectively, you need to be aware of both the advantages and drawbacks of dynamic backup and reorganization.

If it is at all possible, backup or reorganize the IDS when Advanced Backup and Recovery for z/OS functions can be quiesced. A drawback of backing up or reorganizing dynamically is the effect that pausing active product programs might have. On the other hand, the advantage of dynamically reorganizing is that it can be done without pausing Advanced Backup and Recovery for z/OS functions, permitting continued application and backup processing.

Because the ABARS Manager and Critical Backup Tracking and Inventory programs are paused, the backup or reorganization program should be placed in a schedule where that pause has minimal impact. Pausing the programs also suspends ISPF access to data sets. No indications are displayed when programs executed under ISPF pause due to a backup or reorganization executing.

Consider an example in which heavy batch use of the IDS by Advanced Backup and Recovery for z/OS begins late in the evening. By scheduling the reorganization program after daytime ISPF access but before batch processing begins minimizes the consequences of access pauses.

A minor danger inherent to the process needs to be assessed. Because the IDS is reloaded, the data set must be defined with the REUSE attribute. The sample jobs in the product JCL library perform IDCAMS ALTER functions to manipulate the REUSE/NOREUSE attribute for the IDS. While this requirement creates no problems for Advanced Backup and Recovery for z/OS use of the IDS, you must be careful to ensure that other programs do not open the IDS for output, or the data set contents can be overlaid.

Lock data set

The lock data set is required by the Dynamic Reorganization utility.

The lock data set is defined in the :SI040_VALUES section in the BKMINI member of the product parameter library. It can have any valid data set name.

Do not create any JCL steps that delete the lock data set. It is used within the Dynamic Reorganization utility only.

At the start of the dynamic reorganization process, the utility frees the lock data set with a DISP=(OLD,DELETE) specification and then allocates the data set. When the reorganization completes successfully, the utility deletes the lock data set.

Using the dynamic data set backup/reorganization utility

Using this utility involves three tasks: modifying the SI040 tokens in the product parameter library, adding the REUSE attribute to the IDS, and tailoring and executing the JCL.

Modifying the SI040 tokens in the Advanced Backup and Recovery for z/OS parameter library

The first step in using the Dynamic Data Set Backup/Reorganization utility is to modify the tokens in the Advanced Backup and Recovery for z/OS parameter library member BKMINI.

The tokens have the following commented settings:

```
*BKM.INVDB.LOCKDSN=BKM.INVDB.USED.AS.A.LOCK.SWITCH
*LOCKDSN.ATTRIB=STORCLAS(DEFAULT) UNIT(SYSALLDA)
```

Use an ISPF edit session to assign appropriate values to these tokens.

Adding the REUSE attribute to the IDS

This step is performed by the sample jobs in the product JCL library. The jobs use IDCAMS to ALTER the IDS attributes before and after the processes. To minimize risk of corruption during regular processing, the jobs set the IDS attribute to REUSE during the processes and back to NOREUSE after completion.

Tailoring and executing the dynamic backup/reorganization utility JCL

After you have modified the SI040 tokens in BKMINI, the final step in using the dynamic backup/reorganization utility is to tailor and execute the utility JCL.

Tailor and execute the backup, reorganization, or reload utility JCL found in these Advanced Backup and Recovery for z/OS JCL members:

- BKMBAKUP–To back up the IDS.
- BKMRELOD–To reload the IDS that was unloaded using the BKMBAKUP member.
- BKMREORG–To reorganize the IDS.

Note: Although the output of the unload phase of the BKMREORG job is essentially a backup, you might want to make your own backup for the first attempt using this facility.

About the Dynamic Backup/Reorganization utility's BKMBAKUP JCL

You can use the BKMBAKUP job in the product JCL library to perform regular Inventory Data Set backups without stopping the product started tasks, application backups, or ISPF access to the Inventory Data Set.

All of these functions are delayed dynamically by the utility program BKM01BRR. To minimize the delay, it is advisable to perform the backup when Inventory Data Set access requirements are at a minimum. The BKMBAKUP job performs the following functions:

- Deletes the backup output file.
- ALTERs the Inventory Data Set to have the REUSE attribute.
- Backs up the Inventory Data Set.
- ALTERs the Inventory Data Set to have the NOREUSE attribute.

Prior to execution of the BKMBAKUP job, you must modify the SI040_VALUES section of the product parameter library BKMINI member to define a lock data set for the Inventory Data Set to be backed up. Refer to the appropriate section of Appendix B for more information about specifying the locking mechanism.

About the Dynamic Backup/Reorganization utility's BKMREORG JCL

You can use the BKMREORG job in the product JCL library to perform Inventory Data Set reorganizations without stopping the product started tasks, application backups, or ISPF access to the Inventory Data Set.

All of these functions are delayed dynamically by the utility program BKM01BRR. To minimize the delay, it is advisable to perform the reorganization when Inventory Data Set access requirements are at a minimum.

The BKMREORG job performs the following functions:

- Deletes the reorganization output/input file.
- ALTERs the Inventory Data Set to have the REUSE attribute.
- Reorganizes the Inventory Data Set.
- ALTERs the Inventory Data Set to have the NOREUSE attribute.

Prior to executing the BKMREORG job, you must modify the SI040_VALUES section of the product parameter library BKMINI member to define a lock data set for the Inventory Data Set to be reorganized. Refer to the appropriate section of *Appendix B* for more detailed information about specifying the locking mechanism.

About the Dynamic Backup/Reorganization utility's BKMRELOD JCL

You can use the BKMRELOD job in the product JCL library to reload the Inventory Data Set from a backup or a failed reorganization without stopping the product started tasks, application backups, or ISPF access to the Inventory Data Set.

All of these functions are delayed dynamically by the utility program BKM01BRR. To minimize the delay, it is advisable to perform the reload when Inventory Data Set access requirements are at a minimum. The BKMRELOD job performs the following functions:

- ALTERs the Inventory Data Set to have the REUSE attribute.
- Reloads the Inventory Data Set from a previous backup.
- ALTERs the Inventory Data Set to have the NOREUSE attribute.

Prior to executing the BKMRELOD job, you must modify the SI040_VALUES section of the product parameter library BKMINI member to define a lock data set for the Inventory Data Set to be reloaded. Refer to the appropriate section in Appendix B for more detailed information about specifying the locking mechanism.

About the BKM01BRR program parameters

For your reference, this topic provides detailed information about the BACKUP, REORG, and RELOAD BKM01BRR program parameter values.

BACKUP

The BACKUP parameter directs the Dynamic Data Set Backup/Reorganization program to unload the VSAM DD, or Advanced Backup and Recovery for z/OS Inventory Data Set (IDS). The output, specified by the `QSAM DD`, is in key sequence and acceptable as input to a REPRO.

A BACKUP taken using BKM01BRR can be executed while ABARS Manager and Critical Backup Tracking and Inventory are accessing the IDS or VSAM DD.

ABARS Manager and Critical Backup Tracking and Inventory programs pause during the backup. The backup might be 'fuzzy' with respect to groups of records representing an event (an ABARS Manager or Critical Backup Tracking and Inventory backup event).

RELOAD

The RELOAD parameter directs the program to perform a reload from a previous backup or a failed load during a reorganization. The reload can succeed only if the QSAM DD data set was retained from a previous execution of BKMBAKUP or BKMREORG.

REORG

The REORG parameter directs the program to perform an in-place reorganization of the IDS named by the `VSAM DD` statement. The program unloads the existing VSAM records in key sequence to the data set named by the `QSAM DD` statement and then reloads the VSAM data set from the QSAM data set. Reorganization is achieved because of the REUSE attribute that allows 'load mode.'

REORG is only possible when the following are true:

- The IDS has the **REUSE** attribute.
- The required LOCKDSN token has been added to the Advanced Backup and Recovery for z/OS product PARMLIB BKMINI member.

Any ABARS Manager or Critical Backup Tracking and Inventory programs accessing the IDS close the data set and pause until the REORG is finished.

The job stream setup should include a method of notifying the appropriate person if the program ends abnormally or ends with a nonzero return code. If the reload phase of the program fails for any reason,

ABARS Manager or Critical Backup Tracking and Inventory programs attempting to access the data set will not function.

Should the reload phase of the program fail, you must ascertain the source of the problem, fix it, and run BKM01BRR again with the RELOAD parm. The work data set used by REORG (QSAM DD) may be used as a backup data set. The data set contents are sufficient to be used as input to a subsequent REPRO. Consider allocating this data set on a permanent volume.

Using the IDS Health Check utility

Over time, the Inventory Data Set might become corrupted due to various causes, such as unfortunately timed cancellations, system failures, and so on. In addition, the cataloged ABARS backup files might be unsynchronized with the IDS contents.

You can run the IDS Health Check utility to validate IDS records, fix the problems found, and identify ABARS backups that are missing or orphaned, according to the IDS contents. Execute this utility on a regular basis to help ensure the ABARS backups you need do exist and your data is recoverable.

Because the IDS Health Check utility obtains and holds exclusive control of the IDS for the duration of utility processing, run the utility at a time when it has minimal impact on other Advanced Backup and Recovery for z/OS processing.

The IDS Health Check is a process that runs a series of independent record verifications. Health Check determines whether IDS records can be cross-referenced and used when data is being tracked, backed up, and recovered. It looks for orphan records and missing records.

Health Check can be run in two modes:

- **SCAN**

  In SCAN mode, Health Check reports on the orphan records and missing records that it detects in the IDS, but does not attempt to remedy the problems that it finds.

- **FIX**

  In FIX mode, Health Check attempts to repair the problems that it detects in the IDS. During the repair process, orphan records are deleted and missing records can be recreated using the data from other related records.

Note: It is strongly recommended that you back up your IDS before running the Health Check utility in FIX mode. However, you do not need to back up the IDS before running the Health Check utility in SCAN mode.

The following figure shows the Health Check JCL:

```
//jobcard ...
//*********************************************************************
//* RUN HEALTH CHECKER
//*********************************************************************
//*TSTIDSHC EXEC PGM=BKMIDSHC,PARM='FIX'
//*TSTIDSHC EXEC PGM=BKMIDSHC,PARM='FIX,BIGD'
//TSTIDSHC EXEC PGM=BKMIDSHC,PARM='SCAN'
//*TSTIDSHC EXEC PGM=BKMIDSHC,PARM='SCAN,BIGD'
//STELIB   DD  DISP=SHR,DSN=bkm.product.load.library
//INI      DD  DISP=SHR,DSN=bkm.product.parmlib(BKMINI)
//BKMIDS   DD  DISP=SHR,DSN=BKMX.PROD.INVDB
//BKMLOG   DD  SYSOUT=*,DCB=(RECFM=VBA,LRECL=133,BLKSIZE=32750)
//SYSPRINT DD  SYSOUT=*,DCB=(RECFM=VBA,LRECL=133,BLKSIZE=32750)
```

As Health Check processes the IDS, it generates report progress status messages and displays them at the location indicated by the BKMLOG DD statement. Faulty records are listed at the location indicated by the SYSPRINT DD statement.

Note:

- PARM='SCAN' causes the utility to scan the IDS and note the error that are encountered, without fixing the errors.

- PARM='FIX' causes the utility to scan the IDS and fix the reparable errors that are encountered.
- For very large IDSs (typically larger than 15 million records), the Health Check utility sometimes shuts down with a return code of 8 and a message similar to the following:

```
BKMS8000E  Error calling BKM01VV1 Function=PUT Table=DSND-TABLE,
(BKMIDSHB+00001FFA)
```

  In this situation, consider specifying the BIGD parameter (as shown in the example JCL), which forces the Health Check utility to use a special technique that is better at handling large databases but takes longer to run.

1. Edit member BKMIDSHC in the product JCL library, verifying that the execute parameter is set to the appropriate value:
   - Specify SCAN if you want the utility to validate the IDS record structure and reconcile the ABARS backups and the IDS contents and report the errors it finds, but do not want the utility to make any changes to the IDS.
   - Specify FIX if you want the utility to validate the IDS record structure and reconcile the ABARS backups and the IDS contents, fix all record inconsistency problems it finds, and report the errors it finds, along with the corrective action taken.
2. Make any other appropriate changes to the JCL.
3. Submit the job for execution.

Checking for ASTE with no ASTO

What is an ASTO record?	An ASTO record is the output file descriptor record for backups and volume dumps. Every ASTO record should have an associated ASTE record.
What is Health Check doing?	Checking that each backup or dump ASTE record has an ASTO record associated with it.
How do I fix errors?	Run Health Check again, using PARM=FIX to delete all backup or dump ASTE records that have no corresponding ASTO record.

This is a sample ASTE with No ASTO report:

```
JOB NAME        DATE       TIME       STEP#    TASKID    EV#     COMMENT
-------------------------------------------------------------------------------
JHLDSSDP    2014/04/29  11:50:43.68    001      0002    00001590
JHLDSSDP    2014/04/29  11:50:43.68    001      0003    00001591
JHLDSSDP    2014/05/29  07:07:14.55    001      0002    00001739
JHLDSSDP    2014/05/29  07:07:14.55    001      0003    00001740
JHLDSSDP    2014/05/29  09:38:45.13    001      0002    00001741
JHLDSSDP    2014/05/29  09:38:45.13    001      0003    00001742
JHLDSSDP    2014/05/29  10:11:40.41    001      0002
00001743
```

Checking for ASTO orphans

What is an ASTO record?	An ASTO record is the output file descriptor record for backups and volume dumps. Every ASTO record should have an associated ASTE record.

What is Health Check doing?	Scanning all ASTO records to verify that each one has an ASTE record associated with it.
How do I fix errors?	Run Health Check again, using PARM=FIX to delete all ASTO records that have no corresponding ASTE record.

This is a sample ASTO Orphans report:

```
                        ASTO ORPHANS
JOB NAME        DATE       TIME       STEP#    TASKID    EV#     COMMENT
-------------------------------------------------------------------------------
EDP#$#$#    2011/10/03  08:34:10.24    001      0001
----
```

Checking for BABR orphans

What is a BABR record?	A BABR record is a copy of the BKM record that is created by ABACKUP and is updated by ARECOVER. Every BABR record should have an associated BAEV record.
What is Health Check doing?	Scanning all BABR records to verify that each one has a BAEV record associated with it.
How do I fix errors?	Run Health Check again, using PARM=FIX to delete all BABR records that have no corresponding BAEV record.

Checking for BAEV with no BCFL

What is a BAEV record?	A BAEV (ABARS Backup Event) record contains general and metadata about a single backup.
What is a BCFL record?	A BCFL record is a backup event output file descriptor record. Every BCFL record should be associated with a BAEV record.
What is Health Check doing?	Scanning all BAEV records to verify that each one has a BCFL record associated with it.
How do I fix errors?	Run Health Check again, using PARM=FIX to delete all BAEV records that have no corresponding BCFL record.

Checking for BCFL orphans

What is a BCFL record?	A BCFL record is a backup event output file descriptor record. Every BCFL record should have an associated BAEV record.

What is Health Check doing?	Scanning all BCFL records to verify that each one has a BAEV record associated with it.
How do I fix errors?	Run Health Check again, using PARM=FIX to delete all BCFL records that have no corresponding BAEV record.

Checking for BDFL orphans

What is a BDFL record?	A BDFL record is a backup event output file descriptor record. The **D** file is created when there are level 0 disk data sets in the INCLUDE list. The BDFL record contains information such as the aggregate name, run date and time, number of output volumes, and the output volume volsers. Every BDFL record should have an associated BAEV record.
What is Health Check doing?	Scanning all BDFL records to verify that each one has a BAEV record associated with it.
How do I fix errors?	Run Health Check again, using PARM=FIX to delete all BDFL records that have no corresponding BAEV record.

Checking for BDSA orphans

What is a BDSA record?	A BDSA record contains metadata about a single backup of a data set. The metadata is used for reporting and data set restoration. Every BDSA record should have an associated ASTE record.
What is Health Check doing?	Scanning all BDSA records to verify that each one has an ASTE record associated with it.
How do I fix errors?	Run Health Check again, using PARM=FIX to delete all BDSA records that have no corresponding ASTE record.

This is a sample BDSA Orphans report:

```
                          BDSA ORPHANS
   EV#    DATASET#  JOB NAME     DATE       TIME      STEP#  TASKID  DATASET NAME
-------------------------------------------------------------------------------
00000000  00000001  JHL#$#$#  2013/09/19  08:00:56.64  001    0001   PRO.DSDSBBST
00000000  00000002  JHL#$#$#  2013/09/19  08:00:56.64  001    0001   PRO.DSDSBBST
00000000  00000003  JHL#$#$#  2013/09/19  08:00:56.64  001    0001   PRO.DSDSBBST
00000000  00000004  JHL#$#$#  2013/09/19  08:00:56.64  001    0001   PRO.DSDSBBST
00000000  00000005  JHL#$#$#  2013/09/19  08:00:56.64  001    0001   PRO.DSDSBBST
00000000  00000006  JHL#$#$#  2013/09/19  08:00:56.64  001
0001
```

Checking for BDSS orphans

What is a BDSS record?	A BDSS record is used during flat file processing to write the official BDSA record. Every BDSS record should have an associated BDSA record.
What is Health Check doing?	Scanning all BDSS records to verify that each one has a BDSA record associated with it.
How do I fix errors?	Run Health Check again, using PARM=FIX to delete all BDSS records that have no corresponding BDSA record.

This is a sample BDSS orphans report:

```
                        BDSS ORPHANS
   EV#    DATASET#  JOBNAME      DATE       TIME      STEP#  TASKID  DATASET NAME
-------------------------------------------------------------------------------
00000000  00000001  JHLDSSDP  2014/06/20  13:54:51.47  001    0002   AE3.CADISK.FILE
00000000  00000001  JHLDSSDP  2014/06/20  13:54:51.47  001    0003   BK.BK3.BUILD.LIST50
00000000  00000002  JHLDSSDP  2014/06/20  13:54:51.47  001    0002   ARH.ARH0621.LOADLIB.XMIT
00000000  00000002  JHLDSSDP  2014/06/20  13:54:51.47  001    0003   CSJUST.SPFTEMP4.CNTL
00000000  00000003  JHLDSSDP  2014/06/20  13:54:51.47  001    0002   ARH.ARH0621.LOADLIB2.XMIT
00000000  00000003  JHLDSSDP  2014/06/20  13:54:51.47  001    0003   CSJUST.TST.SPFTEMPS.CNTL
```

Checking for BERM orphans

What is a BERM record?	A BERM record contains all of the error and warning messages that are issued during an ABACKUP. Every BERM record should have an associated BAEV record.
What is Health Check doing?	Scanning all BERM records to verify that each one has a BAEV record associated with it.
How do I fix errors?	Run Health Check again, using PARM=FIX to delete all BERM records that have no corresponding BAEV record.

This is a sample BERM Orphans report:

```
                           BERM ORPHANS
JOB NAME      DATE       TIME      COMMENT
-------------------------------------------------------------------------------
ALBTEST0  2014/05/16  16:52:28.00
ALBTEST0  2014/06/20  16:50:11.00
ALBTEST0  2014/11/21  10:29:21.00
ALBTEST1  2014/05/16  16:52:28.00
ALBTEST1  2014/07/01  19:56:54.00
ALBTEST1  2014/07/26  18:35:51.00
ALBTEST1  2014/08/01  15:09:44.00
```

Checking for BIFL orphans

What is a BIFL record?	A BIFL record is a backup event output file descriptor record.

	The **I** file is created when an instruction/activity data set has been created. The BIFL record contains information such as the aggregate name, run date and time, number of output volumes, and the output volume volsers. Every BIFL record should have an associated BAEV record.
What is Health Check doing?	Scanning all BIFL records to verify that each one has a BAEV record associated with it.
How do I fix errors?	Run Health Check again, using PARM=FIX to delete all BIFL records that have no corresponding BAEV record.

Checking for BOFL orphans

What is a BOFL record?	A BOFL record is a backup event output file descriptor record. The **0** file is created when there are user tape data sets or migrated data sets in the INCLUDE list. The BOFL record contains information such as the aggregate name, run date and time, number of output volumes, and the output volume volsers. Every BOFL record should have an associated BAEV record.
What is Health Check doing?	Scanning all BOFL records to verify that each one has a BAEV record associated with it.
How do I fix errors?	Run Health Check again, using PARM=FIX to delete all BOFL records that have no corresponding BAEV record.

Checking for missing CFIL records

What is a CFIL record?	CFIL records are created by ABACKUP to cross-reference the control file with the aggregate that was backed up. The control file contains the activity log data set and instruction data set were created during the ABACKUP. The control file also contains space and catalog information for data to be allocated, a copy of the migration control data set (MCDS) records for each migrated data set, space information for recovering the aggregate, D, O, and I file names, and DFMSMS class definitions. Every CFIL record should have an associated BCFL record.
What is a BCFL record?	A BCFL record is a backup event output file descriptor record that contains the aggregate name, run date and time, number of output volumes, and the output volume volsers.

	Every BCFL record should have an associated BAEV record.
What is Health Check doing?	Scanning all BCFL records to verify that each one has a CFIL record associated with it.
How do I fix errors?	Run Health Check again, using PARM=FIX to re-create missing CFIL records and associate them their corresponding BCFL records.

Checking for CFIL with no BCFL

What is a CFIL record?	A CFIL record is created by ABACKUP to cross-reference the control file with the aggregate that was backed up. The control file contains the activity log data set and instruction data set that was created during the ABACKUP job. The CFIL record contains space and catalog information for data to be allocated, a copy of the migration control data set (MCDS) records for each migrated data set, space information for recovering the aggregate, D, O, and I file names, and DFMSMS class definitions. Every CFIL record should have an associated BCFL record.
What is a BCFL record?	A BCFL record is a backup event output file descriptor record that contains information such as aggregate name, the run date and time, number of output volumes, and output volume volsers. Every BCFL record should have an associated BAEV record.
What is Health Check doing?	Scanning all BCFL records to verify that each one has a CFIL record associated with it.
How do I fix errors?	Run Health Check again, using PARM=FIX to re-create missing CFIL record and associate it with its corresponding BCFL record.

This is a sample BAEV with no BCFL report:

```
                        BAEV WITH NO BCFL
AGGNAME      DATE       TIME      COMMENT
-------------------------------------------------------------------------------
ALBTEST0  2014/05/16  16:52:28.00
ALBTEST0  2014/06/20  16:50:11.00
ALBTEST0  2014/11/21  10:29:21.00
ALBTEST1  2014/05/16  16:52:28.00
ALBTEST1  2014/07/01  19:56:54.00
ALBTEST1  2014/07/26  18:35:58.00
ALBTEST1  2014/08/01  15:09:44.00
ALBTEST1  2014/11/24  11:02:49.00
```

Checking for DSNA orphans

What is a DSNA record?	A DSNA record describes a single ABARS data set backup per event. Every DSNA record should have an associated BAEV record.
What is Health Check doing?	Scanning all DSNA records to verify that each one has a BAEV record associated with it.
How do I fix errors?	Run Health Check again, using PARM=FIX to delete all DSNA records that have no corresponding BAEV record.

Checking for missing DSND records

What is a DSND record?	A DSND record is very similar to DSNA record. However, the DSND record has an alternate key. Every DSND record should have an associated DSNA record.
What is Health Check doing?	Scanning all DSNA records to verify that each one has a DSND record associated with it.
How do I fix errors?	Run Health Check again, using PARM=FIX to re-create missing DSND record and associate it to the corresponding DSNA record.

Checking for DSND orphans

What is a DSND record?	A DSND record is a near-duplicate of a DSNA record. However, the DSND record has an alternate key to speed up searches. Every DSND record should have an associated DSNA record.
What is Health Check doing?	Scanning all DSND records to verify that each one has a DSNA record associated with it.
How do I fix errors?	Run Health Check again, using PARM=FIX to delete all DSND records that have no corresponding DSNA record.

Checking for missing DSNX records

What is a DSNX record?	A DSNX record contains metadata that describes a single backup of a data set. The metadata is used for reporting and data set restoration. Every DSNX record should have an associated BDSA record.
What is Health Check doing?	Scanning all BDSA records to verify that each one has a DSNX record associated with it.

How do I fix errors?	Run Health Check again, using PARM=FIX to re-create a missing DSNX record and associate it with the corresponding BDSA record.

Checking for DSNX orphans

What is a DSNX record?	A DSNX record contains metadata that describes a single backup of a data set. The metadata is used for reporting and data set restoration. Every DSNX record should have an associated BDSA record.
What is Health Check doing?	Scanning all DSNX records to verify that each one has a BDSA record associated with it.
How do I fix errors?	Run Health Check again, using PARM=FIX to delete all DSNX records that have no corresponding BDSA record.

This is a sample DSNX Orphans report:

```
                    DSNX ORPHANS
DATASET NAME     JOBNAME      DATE       TIME      STEP#  TASKID   EV#      DATASET#
-------------------------------------------------------------------------------------
AE3.CADISK.FILE  JHLDSSDP  2014/04/29  11:50:43.68  001   0002    00001590  00000001
AE3.CADISK.FILE  JHLDSSDP  2014/05/12  07:07:14.55  001   0002    00001739  00000001
AE3.CADISK.FILE  JHLDSSDP  2014/05/12  09:38:45.13  001   0002    00001741  00000001
AE3.CADISK.FILE  JHLDSSDP  2014/05/12  11:11:40.41  001   0002    00001743  00000001
AE3.CADISK.FILE  JHLDSSDP  2014/05/12  11:54:18.10  001   0002    00001756  00000001
AE3.CADISK.FILE  JHLDSSDP  2014/05/12  12:57:44.51  001   0002    00001758  00000001
```

Checking for EADD orphans

What is an EADD record?	EADD records store additional information about backup events. The nature of the information stored in an EADD record is determined by the backup utility that was used for the event. For example, when the backup utility is CA Disk, the EADD record stores the name of the CA Disk Files Data Set that was in use at the time at which the backup was taken. There should be one EADD record for ASTE record.
What is Health Check doing?	Scanning all EADD records to verify that each one has an ASTE record associated with it.
How do I fix errors?	Run Health Check again, using PARM=FIX to delete all EADD records that have no corresponding ASTE record.

Checking for missing FREB records

What is a FREB record?	Fast Replication backup (FREB) records are used to store Fast Replication Services backup information. FREB records exist for full volume dump events only.

	Every FREB record should have an associated ASTE record.
What is Health Check doing?	Scanning all ASTE records that are associated with Fast Replication Services backups to verify that each ASTE record has a FREB record associated with it.
How do I fix errors?	Run Health Check again, using PARM=FIX to re-create the missing FREB record and associate it with the corresponding ASTE record.

Checking for FREB orphans

What is a FREB record?	Fast Replication backup (FREB) records are used to store Fast Replication Services backup information. FREB records exist for full volume dump events only. Every FREB record should have an associated ASTE record.
What is Health Check doing?	Scanning all FREB records to verify that each one has an ASTE record associated with it.
How do I fix errors?	Run Health Check again, using PARM=FIX to delete all FREB records that have no corresponding ASTE record.

This is a sample FREB Orphans report:

```
                          FREB ORPHANS
JOB NAME        DATE       TIME       STEP#    TASKID    EV#     COMMENT
-------------------------------------------------------------------------------
APPLBKP    2013/06/20  16:27:05.39    001      0001     ----
APPLBKP    2014/06/19  14:32:52.60    001      0001     ----
```

Checking for missing FRET records

What is a FRET record?	Fast Replication backup Token (FRET) records are used to locate a Fast Replication Services backup. Every FRET record should have an associated ASTE record.
What is Health Check doing?	Scanning all FRET records to verify that each one has an ASTE record associated with it.
How do I fix errors?	Run Health Check again, using PARM=FIX to re-create a missing FRET record and associate it with the orphaned ASTE record.

Checking for FRET orphans

What is a FRET record?	Fast Replication backup Token (FRET) records are used to locate a Fast Replication Services backup.

	Every FRET record should have an associated ASTE record.
What is Health Check doing?	Scanning all FRET records to verify that each one has an ASTE record associated with it.
How do I fix errors?	Run Health Check again, using PARM=FIX to delete all FRET records that have no corresponding ASTE record.

Checking for GDGB orphans

What is a GDGB record?	A GDGB record stores GDG information about the backup output. Every GDGB record should have an associated BAEV record.
What is Health Check doing?	Scanning all GDGB records to verify that each one has a BAEV record associated with it.
How do I fix errors?	Run Health Check again, using PARM=FIX to delete all GDGB records that have no corresponding BAEV record.

This is a sample GDGB Orphans report:

```
                      GDGB ORPHANS
AGGNAME      DATE       TIME      COMMENT
-------------------------------------------------------------------------------
ALBTEST0  2014/05/16  16:52:28.00
ALBTEST0  2014/05/16  16:52:28.00
ALBTEST0  2014/05/16  16:52:28.00
ALBTEST0  2014/05/16  16:52:28.00
ALBTEST0  2014/05/16  16:52:28.00
ALBTEST0  2014/05/16  16:52:28.00
ALBTEST0  2014/05/16  16:52:28.00
ALBTEST0  2014/05/16  16:52:28.00
```

IDS Health Check complete

Health check issues this message when it has completed processing.

Checking for missing JOBC records

What is a JOBC record?	A JOBC record contains general job run time information about a single backup job. This type of information includes job name, start date and time, and so on. Every JOBC record should have an associated ASTE record.
What is Health Check doing?	Scanning all ASTE records to verify that each one has a JOBC record associated with it.
How do I fix errors?	Run Health Check again, using PARM=FIX to re-create a missing JOBC record and associate it with the corresponding ASTE record.

Checking for JOBC orphans

What is a JOBC record?	A JOBC record contains general job run time information about a single backup job. This type of information includes job name, start date and time, and so on. Every JOBC record should have an associated ASTE record.
What is Health Check doing?	Scanning all JOBC records to verify that each one has an ASTE record associated with it.
How do I fix errors?	Run Health Check again, using PARM=FIX to delete all JOBC records that have no corresponding ASTE record.

This is a sample JOBC Orphans report:

```
                      JOBC ORPHANS
JOBNAME      DATE       TIME      COMMENT
-------------------------------------------------------------------------------
JHLDSSDP  2014/04/29  11:50:43.68
JHLDSSDP  2014/05/12  07:07:14.55
JHLDSSDP  2014/05/12  09:38:45.13
JHLDSSDP  2014/05/12  10:11:40.41
JHLDSSDP  2014/05/12  11:54:18.10
JHLDSSDP  2014/05/12  12:57:44.51
JHLDSSDP  2014/05/29  11:48:36.05
JHLDSSDP  2014/05/29  11:54:16.30
```

Checking for RABR orphans

What is an RABR record?	An RABR record is a copy of the ABR record that was created by ABACKUP and updated by ARECOVER. Every RABR record should have an associated BAEV record.
What is Health Check doing?	Scanning all RABR records to verify that each one has a BAEV record associated with it.
How do I fix errors?	Run Health Check again, using PARM=FIX to delete all RABR records that have no corresponding BAEV record.

Checking for RDSA orphans

What is an RDSA record?	An RDSA record is created whenever a data set is restored by aggregate by event. Each RDSA record has an associated RSEV record.
What is Health Check doing?	Scanning all RDSA records to verify that each one has an RSEV record associated with it.
How do I fix errors?	Run Health Check again, using PARM=FIX to delete all RDSA records that have no corresponding RSEV record.

Checking for missing RDSD records

What is an RDSD record?	An RDSD record is identical to an RDSA record except that the RDSD record has an alternate key. Each RDSA record has an associated RDSD record.
What is Health Check doing?	Scanning all RDSA records to verify that each one has an RDSD record associated with it.
How do I fix errors?	Run Health Check again, using PARM=FIX to re-create every missing RDSD record and associate the re-created record with its corresponding RDSA record.

Checking for RDSD orphans

What is an RDSD record?	An RDSD record is identical to an RDSA record except for its alternate key. Each RDSD record has an associated RDSA record.
What is Health Check doing?	Scanning all RDSD records to verify that each one has an RDSA record associated with it.
How do I fix errors?	Run Health Check again, using PARM=FIX to delete all RDSD records that have no corresponding RDSA record.

Checking for SDSA orphans

What is an SDSA record?	An SDSA record the Selection Data Set after expansion. Every SDSA record has an associated BAEV record.
What is Health Check doing?	Scanning all SDSA records to verify that each one has a BAEV record associated with it.
How do I fix errors?	Run Health Check again, using PARM=FIX to delete all SDSA records that have no corresponding BAEV record.

This is a sample SDSA Orphans report:

```
                        SDSA ORPHANS
AGGNAME      DATE       TIME      COMMENT
-------------------------------------------------------------------------------
ALBTEST0  2014/05/16  16:52:28.00
ALBTEST0  2014/06/20  16:50:11.00
ALBTEST0  2014/11/21  10:29:21.00
ALBTEST1  2014/05/16  16:52:28.00
ALBTEST1  2014/07/01  19:56:54.00
ALBTEST1  2014/11/24  11:02:49.00
```

Checking for SDSB orphans

What is an SDSB record?	An SDSB record the Selection Data Set before expansion.

	Every SDSB record should have an associated BAEV record.
What is Health Check doing?	Scanning all SDSB records to verify that each one has a BAEV record associated with it.
How do I fix errors?	Run Health Check again, using PARM=FIX to delete all SDSB records that have no corresponding BAEV record.

This is a sample SDSB Orphans report:

```
                       SDSB ORPHANS
AGGNAME      DATE       TIME      COMMENT
-------------------------------------------------------------------------------
ALBTEST0  2014/05/16  16:52:28.00
ALBTEST0  2014/06/20  16:50:11.00
ALBTEST0  2014/11/21  10:29:21.00
ALBTEST1  2014/05/16  16:52:28.00
ALBTEST1  2014/07/01  19:56:54.00
ALBTEST1  2014/11/24  11:02:49.00
ALBTEST1  2014/11/24  11:17:17.00
```

Checking for TRAC orphans

What is a TRAC record?	A TRAC record is a trace event record that is using when an ABACKUP or ARECOVER job is being traced. Every TRAC record should have an associated BAEV or RSEV record.
What is Health Check doing?	Scanning all TRAC records to verify that each one has a BAEV or RAEV record associated with it.
How do I fix errors?	Run Health Check again, using PARM=FIX to delete all TRAC records that have no corresponding BAEV or RAEV record.

This is a sample TRAC Orphans report:

```
                       TRAC ORPHANS
JOBNAME      DATE       TIME      COMMENT
-------------------------------------------------------------------------------
ALBTEST0  2014/05/16  16:52:28.00
ALBTEST0  2014/05/16  16:52:28.00
ALBTEST0  2014/06/20  16:50:11.00
ALBTEST0  2014/06/20  16:50:11.00
ALBTEST0  2014/06/23  08:04:13.00
```

Checking for uncataloged C files

What is Health Check doing?	Retrieving from the IDS all of the CFILE records and checking their catalog status, then generating a report showing the CFILEs that are not cataloged.
How do I fix errors?	This is probably an error, but Health Check cannot fix it. The user must decide whether to delete the uncataloged CFILEs.

This is a sample ABARS Backups Tracked in IDS but C File not cataloged report:

Checking for untracked ABARS backups

What is Health Check doing?	Retrieving the defined SMS aggregate constructs, building a generic catalog LOCATE command for the aggregate C files, and returning a list that is compared to the content of the IDS. If processing finds a match between the list contents and the IDS contents, that backup is a tracked ABARS backup.
How do I fix errors?	This is not an error situation, but processing generates the ABARS backups not Tracked in IDS report, which contains a list of untracked ABARS backups.

This is a sample ABARS Backups Not Tracked in IDS report:

Using option 2

What does it mean?	This is not a message with which the user should be concerned, unless support or development deems it necessary. This message indicates what type of processing is occuring and that information helps development determine whether a BDSA record should be deleted.
What happens when an error is found?	Nothing, even if you run HC in FIX mode.
How do I fix errors?	N/A.

Validating compressed records

What is HC doing?	Reading every record in the IDS and verifying that the compression checksum is valid for every record.
What happens when an error is found?	Nothing, even if you run HC in FIX mode.
How do I fix errors?	REPRO will not fix this type of problem. Use the UNLOAD and LOAD utilities, which are documented in the section Appendix H, “Inventory data set administration,” on page 523

Validating EV# to ASTE usage

What is HC doing?	Verifying that an event number is assigned to only one Critical Backup Tracking and Inventory event (ASTE record). Every time that an Critical Backup Tracking and Inventory event (such as a backup) starts, an ASTE record is created and a unique event number is

	assigned to it. A non-unique event number is an error condition.
What happens when an error is found?	Nothing, even if you run HC in FIX mode.
How do I fix errors?	REPRO will not fix this type of problem. Use the UNLOAD and LOAD utilities, which are documented in the section Appendix H, “Inventory data set administration,” on page 523

Checking for missing VOLX records

What is a VOLX record?	Everytime a full volume dump is performed, a VOLX record is created. Every VOLX record should have an associated ASTE record.
What is Health Check doing?	Scanning all VOLX records to verify that each one has an ASTE record associated with it.
How do I fix errors?	Run Health Check again, using PARM=FIX to re-create the missing VOLX record and associate it with an ASTE record.

Checking for ZACT orphans

What is a ZACT record?	A ZACT record contains the activity log that was produced during an ABACKUP or an ARECOVER. Every ZACT record should have an associated BAEV or RSEV record.
What is Health Check doing?	Scanning all ZACT records to verify that each one has a BAEV or RSEV record associated with it.
How do I fix errors?	Run Health Check again, using PARM=FIX to delete all ZACT records that have no corresponding BAEV or RSEV record.

Checking for ZCOB orphans

What is a ZCOB record?	A ZCOB record contains the conflict data set that is created before an ARECOVER starts. Every ZCOB record should have an associated RSEV record.
What is Health Check doing?	Scanning all ZCOB records to verify that each one has an RSEV record associated with it.
How do I fix errors?	Run Health Check again, using PARM=FIX to delete all ZCOB records that have no corresponding RSEV record.

Checking for ZCON orphans

What is a ZCON record?	A ZCON record contains the conflict data set that is created after an ARECOVER ends. Every ZCON record should have an associated RSEV record.
What is Health Check doing?	Scanning all ZCON records to verify that each one has an RSEV record associated with it.
How do I fix errors?	Run Health Check again, using PARM=FIX to delete all ZCON records that have no corresponding RSEV record.

Checking for ZINS orphans

What is a ZINS record?	A ZINS record contains the instruction data set. Every ZINS record should have an associated BAEV record.
What is Health Check doing?	Scanning all ZINS records to verify that each one has a BAEV record associated with it.
How do I fix errors?	Run Health Check again, using PARM=FIX to delete all ZINS records that have no corresponding BAEV record.

This is a sample ZINS Orphans report:

```
                        ZINS ORPHANS
AGGNAME      DATE       TIME      COMMENT
-------------------------------------------------------------------------------
BMPTEST2  2014/07/26  02:16:24.00
BMPTEST2  2014/07/26  02:55:24.00
```

Checking for ZRSB orphans

What is a ZRSB record?	A ZRSB record contains the restart data set that is created before an ARECOVER starts. Every ZRSB record should have an associated RSEV record.
What is Health Check doing?	Scanning all ZRSB records to verify that each one has an RSEV record associated with it.
How do I fix errors?	Run Health Check again, using PARM=FIX to delete all ZRSB records that have no corresponding RSEV record.

Checking for ZRST orphans

What is a ZRST record?	A ZRST record contains the restart data set that is created after an ARECOVER ends.

	Every ZRST record should have an associated RSEV record.
What is Health Check doing?	Scanning all ZRST records to verify that each one has an RSEV record associated with it.
How do I fix errors?	Run Health Check again, using PARM=FIX to delete all ZRST records that have no corresponding RSEV record.

Loading Critical Backup Tracking and Inventory data into the Advanced Backup and Recovery for z/OS Inventory Data Set

The **Utilities** menu option **Load Inventory (Critical Backup Tracking and Inventory)** reloads Critical Backup Tracking and Inventory data into the Advanced Backup and Recovery for z/OS Inventory Data Set from a data set that was previously unloaded.

Note: ABARS Manager and Critical Backup Tracking and Inventory share the same Advanced Backup and Recovery for z/OS Inventory Data Set.

The default is to load all Critical Backup Tracking and Inventory data previously extracted from the Advanced Backup and Recovery for z/OS Inventory Data Set. Optional keywords JOB(nn) and GENS(nn) can be used to selectively load previously extracted Advanced Backup and Recovery for z/OS Inventory Data Set data, instead of the default of all extracted data.

1. From the action bar at the top of the panel, select **Utilities** and then select the **Load Inventory (Critical Backup Tracking and Inventory)** option.
2. In the **Input File DSN** field, type the name of the data set that contains the data to load into the Advanced Backup and Recovery for z/OS Inventory Data Set.
3. In the **Output Inventory DSN** field, type the name of the new Advanced Backup and Recovery for z/OS Inventory Data Set to load. This parameter will override the default in BKMINI.
4. Specify items to include in the load:
   a) In the **Include Job** field, type the names or masks of the jobs to include in the load of the Advanced Backup and Recovery for z/OS Inventory Data Set. The default is all jobs will be loaded
   b) In the **Include Generations** field, type the number of Critical Backup Tracking and Inventory job generations to include in the load operation. Make the appropriate specification:
      - 0–9999 — The number of generations to load. For example, if you specify 3, the three most recent generations will be loaded. If you specify 0, no generations are loaded.
      - *— (Default) Include all job generations.
   c) In the **Replace** field, type Y or N to specify whether you want the current entries replaced.
5. In the **Exclude Job** field, type the names or masks of the jobs to exclude from the load operation.
6. Select to build and submit the JCL by selecting one of these options:
   - To build and submit the JCL, type S in the **Build/Submit JCL** field. Selecting this option initiates a reload of Critical Backup Tracking and Inventory data into the Advanced Backup and Recovery for z/OS Inventory Data Set. The JCL will be built using the options supplied on the panel and the load will be submitted.
   - To build and view the JCL, type S in the **Build/View JCL** field. Selecting this option allows you to view the JCL before actual submission of the load, or to simply browse the parameters used. The JCL will be built using the options on the panel and the results will be displayed.

Unloading Critical Backup Tracking and Inventory data from the Advanced Backup and Recovery for z/OS Inventory Data Set

Use the Utilities menu option **Unload Inventory (Critical Backup Tracking and Inventory)** to unload Critical Backup Tracking and Inventory event data from the Advanced Backup and Recovery for z/OS Inventory Data Set.

Note: ABARS Manager and Critical Backup Tracking and Inventory share the same Advanced Backup and Recovery for z/OS Inventory Data Set.

The default is to unload all Critical Backup Tracking and Inventory data from the Advanced Backup and Recovery for z/OS Inventory Data Set. Optional keywords JOB(nn) and GENS(nn) can be used to selectively unload data, instead of the default of all extracted data.

1. From the action bar at the top of the panel, select **Utilities** and then select option 6, **Unload Inventory (Critical Backup Tracking and Inventory)**.
2. Specify items to include in the unload:
   a) In the **Include Job** field, type the names or masks of the jobs to include in the unload of the Advanced Backup and Recovery for z/OS Inventory Data Set. The default is all jobs will be unloaded.
   b) In the **Include Generations** field, type the number of Critical Backup Tracking and Inventory job generations to include in the unload operation. Specify:
      - 0 – 9999 — The number of generations to unload. For example, if you specify 3, the three most recent generations will be unloaded. If you specify 0, no generations will be unloaded.
      - *— (Default.) Include all job generations.

      Note: Leave all fields blank to unload all records.
3. In the **Exclude Job** field, type the names or masks of the jobs to exclude from the unload operation.
4. (Optional.) In the **Output File DSN** field, specify the output data set where the Advanced Backup and Recovery for z/OS Inventory Data Set records should be unloaded. The default allocation is: `SPACE=(CYL,(1,1)),UNIT=SYSALLDA, RECFM=FB,LRECL=6159, BLKSIZE=0` You can change the default file allocation values by modifying the values for the `UNLOADIDS_SPACE = 'CYLINDERS SPACE(1 2)' UNLOADIDS_UNIT = SYSALLDA` tokens in the :PROCESSING_OPTIONS section of BKMINI. The default is that all data will be unloaded and a report will be created.

   Note: If the specified DSN does not exist, it will be dynamically allocated, and then deleted at the end of the job.
5. Build the JCL by selecting one of these options:
   - To build and submit the JCL, type S in the **Build/Submit JCL** field. Selecting this option initiates a reload of Critical Backup Tracking and Inventory data into the Advanced Backup and Recovery for z/OS Inventory Data Set. The JCL will be built using the options supplied on the panel and the unload will be submitted.
   - To build and view the JCL, type S in the **Build/View JCL** field. Selecting this option allows you to view the JCL before actual submission of the unload, or to browse the parameters used. The JCL will be built using the options on the panel and the results will be displayed.

Appendix I. Data set activity database (DAD) administration

Tracking data changes in the Data Set Activity Database (DAD)

Both ABARS Manager and Critical Backup Tracking and Inventory use the BKMSTSKD (DAD) started task to track new or changed data in the Data Set Activity Data Base (DAD).

BKMSTSKD tracks data sets in real time and maintains the tracked changes in the DAD, independently of the VTOC change bit.

It uses the following operator commands:

```
S BKMSTSKD,JOBNAME=DAD
F DAD,AUTOSTART
P DAD
```

You can manipulate all the other DAD subtasks using the DAD ISPF Monitor commands from the Action bar's **Utilities** menu.

You can start the DAD using any of these methods:

- Start the DAD and its subtasks using the AUTOSTART command.
- Start the DAD and its subtasks by using the job name.
- Start the DAD without starting its subtasks.

Use the **P DAD** command to stop the DAD.

Using AUTOSTART to start the DAD and its subtasks

The **AUTOSTART** command starts the DAD started task and all of its subtasks. This is the most commonly used method of starting the DAD.

BKMSTSKD is the name of a member in the Advanced Backup and Recovery for z/OS JCL library. It is also the name of the procedure; DAD is the name of the started task and the database itself. All modify commands are based on the identifier 'DAD.'

The only variable within the structure is the data space size. The number of buffers is controlled by the **DAD_DSPC_BUFFERS** token in the `:BKMSTSKD_OPTIONS` section of the BKMINI member in the product parameter library. The default size used in AUTOSTART is 1,002 buffers but it is recommended that you monitor these as more data is tracked to ensure that you do not lose records.

Issue the `F DAD,AUTOSTART` console command.

AUTOSTART issues the following commands:

- `F DAD,U83(LOAD)`
- `F DAD, DSPC(CREATE)`
- `F DAD,83(TURNON)`
- `F DAD,DAD(START)`
- `F DAD,U83(INFO)`

Using job name to start the DAD started task and its subtasks

As an alternative to starting the DAD started task and its subtasks using the **AUTOSTART** command, you can use the **JOBNAME** parameter.

The started task can also be started using the **JOBNAME** parameter. When starting the BKMSTSKD, you must specify the target PROCLIB member name in the start command. See member BKMSTSKD in the JCL distribution library.

Note:

- The member and the procedure are named BKMSTSKD; DAD is the name of the started task. All modify commands are based on the identifier DAD.
- If the AutoStart token in the `:BKMSTSKD_OPTIONS` section of the BKMINI member is set to YES, the command `S BKMSTSKD,JOBNAME=DAD` automatically performs an **AUTOSTART**.

Issue the `S BKMSTSKD,JOBNAME=DAD,PARM=AUTOSTART` or `S BKMSTSKD,JOBNAME=DAD,PARM=A` command.

Starting the DAD started task without also starting the subtasks

Although it is not recommended, there might be situations in which you want to start the DAD started task without also starting its subtasks. You can use a console command to do this.

Issue the following console command: `S BKMSTSKD,JOBNAME=DAD`.

The subtasks do not start unless you start them manually.

Stopping the structure and terminating BKMSTSKD

You can issue the `P DAD` console command to stop the DAD structure and terminate the BKMSTSKD.

Issuing this command causes the following commands to be issued:

```
F DAD,DAD(STOP)
F DAD,U83(TURNOFF)
F DAD,DSPC(DELETE)
F DAD,U83(UNLOAD)
```

Manipulating the IEFU83 exit

The information about using the IEFU83 exit program is intended as a reference only. You are strongly advised against issuing any of these commands without guidance from technical support.

```
F DAD,U83(...)
F DAD,U83(LOAD)
```

These commands acquire ECSA space, load the primary and secondary exits, and place DAD9254Z in the SYS.IEFU83 exit lists.

Errors include not getting ECSA space, unauthorized LOAD libraries, I/O errors during Load SVCs and problems with the dynamic exit processor.

The installation must have IEFU83 defined as an exit within the SMFPRMxx member of SYS1.PARMLIB. IEFU83 can be added without an IPL.

```
F DAD,U83(UNLOAD)
```

This command removes DAD9254Z from the SYS.IEFU83 exit lists and deletes the ECSA space. There are interlocks preventing the Unload when the VSAM/DAD sub task is active and when the data space still exits.

```
F DAD,U83(TURNON)
```

This command activates the exit in the SYS.IEFU83 exit lists and modifies the primary exit to call the secondary exit, which starts the data collection. The installation must be collecting SMF 14, 15, 61, 64, 65, and 66 records. An additional general purpose IEFU83 exit is available if you need to suppress the actual logging of these record types or record types needed by any other IEFU83 exits.

If the primary or secondary exit ends abnormally twice, MVS disables the exit.

```
F DAD,U83(TURNOFF)
```

This command deactivates the exit in the SYS.IEFU83 exit list, and modifies the primary exit so that it does not call the secondary exit.

`F DAD,U83(INFO)` or `F DAD,U83`

These commands display information about the state of the exits. The information also shows the creation date of the ECSA.

```
F DAD,U83(NEWSECONDARY)
```

Under some circumstances, it might be necessary to load a new secondary exit without deleting the data space (and its buffers). There are interlocks to prevent this command from working while the exits are active. Follow these steps to load a new secondary exit:

1. Copy a new exit (DAD9254R) to the LOAD library.
2. Quiesce the exits using the **TURNOFF** command.
3. Refresh the code with the `F DAD,U83(NEWSECONDARY)` command.
4. Activate the exits with a **TURNON** command.

Removing obsolete entries from the DAD database

Data set names are recorded in the DAD database when they are found to be new or to have changed. Some of these entries in the DAD database become obsolete and should be cleaned up as part of normal maintenance.

Obsolete entries can exist in the Data Set Activity Database (DAD) for any of these reasons:

- They are not in the current Base/Incremental (Incremental ABARS).
- They are old generation data sets (GDG) that haven't been deleted. For Incremental ABARS, they are no longer in the current Base/Incremental.
- They have been deleted.

The PURGE_UNCATLG = Y parameter in the BKMINI member of the product parameter library automatically deletes entries from the DAD database when an SMF delete is issued for the data set, but note that the other entries are not deleted automatically.

The BKM9254M JCL member shown in the following figure is provided in the Advanced Backup and Recovery for z/OS JCL library. BKM9254M removes obsolete entries from the DAD database. Use this program on a regular basis with DELETE OLDGDGS to remove obsolete GDG entries from the DAD database. The DAD database does not need to be reorganized after you run BKM9254M.

```
//*******************************************************************
//*  MAKE THE FOLLOWING CHANGES:
//*    1. CHANGE JOB CARD TO LOCAL STANDARDS
//*    2. CHANGE PARMLIB= THE BKM PARMLIB NAME
//*    2. CHANGE LOAD1= TO THE BKM AUTHORIZED LIB NAME
//*******************************************************************
//JOBNAME  JOB ,'...',CLASS=?,MSGCLASS=?         <=== SEE NOTE 1
//BKM9254M PROC PARMLIB=prefix.value.SBKMPARM,   <=== SEE NOTE 2
//        LOAD1=prefix.value.SBKMLOAD            <=== SEE NOTE 3
//S1      EXEC PGM=BKM9254M
//STEPLIB  DD  DISP=SHR,DSN=&LOAD1
//MSCPARM  DD  DISP=SHR,DSN=&PARMLIB(BKMINI)
//SYSPRINT DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//        PEND
//*
//BKM9254M EXEC BKM9254M
//SYSUDUMP DD  SYSOUT=*
//SYSIN    DD  *
```

The valid control cards are as follows:

```
DELETE OLDGDGS
DELETE NOT-IN-CAT
DELETE SCRATCHED
DELETE NAME(...)
DELETE NAME(...) NOT-IN-CAT
DELETE NAME(...) SCRATCHED
```

DELETE
This is a required keyword. You must also specify at least one of the other keywords.

OLDGDGS
Specify this keyword to remove uncataloged GDSs. This is an optional keyword.

NOT-IN-CAT
Specify this keyword to remove uncataloged data sets.

SCRATCHED
Specify this keyword to remove deleted data sets.

NAME
Specify this keyword to identify a data set name, either as a mask (`NAME(PAY.WEEK01.G>>>>V00)NOT-IN-CAT`, for example) or as a literal name.

Appendix J. Critical Backup Tracking and Inventory started task (BKMSTSKA)

Backups to be tracked are intercepted by a seamless Critical Backup Tracking and Inventory intercept triggered by matching job names, step names, and utility program names entered into a user-defined table. Although it is necessary to identify which jobs are backups, no JCL changes are required, thus the risk and procedural difficulties with changing production JCL are eliminated.

Backup jobs are identified to Critical Backup Tracking and Inventory through an ISPF interface to establish batch intercept file (BIF) entries. The batch intercept file entries are initially stored in the Advanced Backup and Recovery for z/OS Inventory Data Set.

Note: After creating the batch intercept file entries, the batch intercept file must then be updated with the new entries and the started task notified so the newly defined backups can be tracked.

Critical Backup Tracking and Inventory uses a started task on each image which installs step initiation exit IEFUSI to intercept backups.

Subtasks under the started tasks then process the results of intercepted utilities and record the pertinent data in the Advanced Backup and Recovery for z/OS Inventory Data Set. For most types of backups, Critical Backup Tracking and Inventory captures catalog and volume metadata in synchronization with the status of the data sets at the time of the backup. This ensures that restore allocations match the status of data sets at the time they were backed up.

The Critical Backup Tracking and Inventory tracked backups are tracked in the Advanced Backup and Recovery for z/OS Inventory Data Set by cycle which is recorded as date and time. Each job run that has been tracked is considered a cycle. Within a cycle, the backups are tracked by backup task within job step.

The Advanced Backup and Recovery for z/OS Inventory Data Set includes job name, step name, name of utility, date, time, return codes, input, output, data set backed up, and so on. The data can be viewed through ISPF and allows dynamic recovery of full volume or selected data sets.

Backups whose output data sets are no longer cataloged on the system are expired, or removed from the Advanced Backup and Recovery for z/OS Inventory Data Set. There are some notable exceptions to this rule:

- DFSMShsm AUTODUMPs are synchronized to the HSM BCDS.
- CA-Disk backups are synchronized to the CA-Disk database.
- FRX backups are synchronized to the FRX database.
- For disk to disk copies, the output target volumes are examined. If the output (target) volumes have been modified since the last backup, the backup is considered to be invalid and is expired.

Job names containing multiple backup data sets are not removed from the IDS until all backups in the job are no longer cataloged on the system.

Functions

BKMSTSKA performs these functions:

- Builds an in-storage job mask table from the batch intercept file
- Installs step initiation exit (IEFUSI)
- Sets up communication with other participating systems and ISPF users
- Creates flat file data containing tracked backup information
- Post-processes the backup results (flat file data) and updates the Advanced Backup and Recovery for z/OS inventory data set (IDS)

System configuration modes and Critical Backup Tracking and Inventory tasks

The Critical Backup Tracking and Inventory started task can collect data for backups defined to Critical Backup Tracking and Inventory in either single image or multi-image (SYSPLEX) modes. This capability allows installations to schedule and run backups on any processor image and still maintain continuous collection of backup data.

BKMSTSKA controls the Critical Backup Tracking and Inventory tasks. The started task, BKMSTSKA, must run on each image where you wish to track backups.

Files used by the tracking started task

PARMLIB
Contains the BKMINI member used by BKMSTSKA.

BKMSETUP
JCL to allocate and initialize the Critical Backup Tracking and Inventory started task data sets:

MSGLOG
Message log file for informational, warning, and error messages generated from BKMSTSKA.

MSGQUEUE
Communication file used to communicate between participating systems and ISPF users.

BIF
File that contains backup job names to track.

Message log

The Critical Backup Tracking and Inventory started task maintains and updates the Critical Backup Tracking and Inventory message log file with diagnostic information from each processor image. The Critical Backup Tracking and Inventory message log file is a required file that is allocated by means of supplied JCL members. To view the message log file use ISPF Browse (do not use ISPF Edit).

Flat file processing overview

The Critical Backup Tracking and Inventory started task, BKMSTSKA, captures backup job step information and writes the backup information to sequential files or to one or more VSAM files. The flat file processing function provides multiple controls and options related to the processing of the information captured by tracking backup jobs. If you so specify, the flat file data is processed by individual started tasks and you can customize the flat file data processing interval. The flat file data processing function reads the flat file data, updates the backup data set and job inventory in the IDS, and then disposes of the recorded flat file data.

When you use sequential files or a single VSAM to contain the flat file data, all started tasks with the flatcopy task active share in the processing of the flat file data. When you use multiple VSAM files to contain the flat file data, the started tasks with the flatcopy task active process flat file data identified by either an INI token or by a DDNAME in the specified started task PROC. A single VSAM data set can be used by all systems running the BKMSTSKA started task; or each system can use a different VSAM data set. You can use the JCL member BKMDEFVF to define the VSAM files and it is recommended that you use Record Level Sharing (RLS) for these data sets. If RLS is to be used, for each VSAM data set, add an INI token to section `:SI040_VALUES` of the product parameter library member BKMINI.

Tracking started task messages

If you want the Critical Backup Tracking and Inventory started task to continually track backups, monitor Critical Backup Tracking and Inventory critical messages with an automated operations tool, if possible, to signal any problems that could arise. This will enable you to be notified immediately so the problem can be fixed. The following message should be tracked:

```
CBTI TRACKING SUSPENDED - module name
```

This message is displayed when Critical Backup Tracking and Inventory has been suspended for any reason.

Tracking DFSMShsm auto dumps

DFSMShsm AUTODUMPs are not automatically tracked by Critical Backup Tracking and Inventory. To update the Advanced Backup and Recovery for z/OS Inventory Data Set with current DFSMShsm AUTODUMP information, execute the utility job in the BKMADUMP member of the product JCL library.

Reducing data to be processed by Critical Backup Tracking and Inventory started task

Although Critical Backup Tracking and Inventory tracks backups without application JCL changes, using discrete job names or job name masks, steps, program names, or job name masks have advantages and disadvantages. Masks can significantly reduce the effort of defining backup jobs — but there will be more data to process.

Critical Backup Tracking and Inventory started task commands

This information describes the Critical Backup Tracking and Inventory started task commands.

The command syntax is: F *xxx*, ...

Where *xxx* is the name of the started task.

LOAD
Load the IEFUSI exit.

Example: F BKMSTSKA,LOAD

UNLOAD
Stop the IEFUSI exit by removing the Critical Backup Tracking and Inventory IEFUSI exit and deleting its resources. This command is issued in preparation of loading a new copy of the IEFUSI exit.

Example: F BKMSTSKA,UNLOAD

TURNON
Activate the IEFUSI exit to start Critical Backup Tracking and Inventory job intercepts.

Example: F BKMSTSKA,TURNON

TURNOFF
Deactivate the IEFUSI exit. Turns off the IEFUSI exit and stops Critical Backup Tracking and Inventory job intercepts. This command can be used if the exit is generating an error, or when you are preparing to load a new copy of the IEFUSI exit code.

Example: F BKMSTSKA,TURNOFF

AUTOSTART
Start the started task and all individual processes that are required for Critical Backup Tracking and Inventory to intercept backups.

Example: F BKMSTSKA,AUTOSTART

NEWSECONDARY
Replace IEFUSI secondary exit.

Example: F BKMSTSKA,NEWSECONDARY

REFRESH
Refresh batch intercept file.

Examples: F BKMSTSKA,REFRESH

RELOAD(modname) | RELOAD(ALL)
The F BKMSTSKA,RELOAD(ALL) command should be issued after the FLATCOPY subtask has been stopped. The F BKMSTSKA,STOPFC command will stop the FLATCOPY subtask.

After the F BKMSTSKA,RELOAD(ALL) command has been issued, the FLATCOPY subtask must be restarted. The following command restarts the FLATCOPY subtask:

F BKMSTSKA,STARTFC

STARTFC
When a supported backup job is intercepted by Critical Backup Tracking and Inventory, a flat file containing the intercepted backup data needed to update the Advanced Backup and Recovery for z/OS Inventory Data Set is created. The flat file copy task requires DFSMShsm to obtain statistics for migrated data sets. If DFSMShsm is not available, the flat file copy task generates error messages.

A task attached by BKMSTSKA locates these data sets. The task collects catalog, VVDS, and VTOC data for the input, output, and backed-up data sets. The task then uses the accumulated data to update the Advanced Backup and Recovery for z/OS Inventory Data Set.

This command starts the flat file copy task.

Example: F BKMSTSKA,STARTFC

STOPFC
Stop the flat file copy task.

Example: F BKMSTSKA,STOPFC

STARTMON
Attach monitor subtask.

Example: F BKMSTSKA,STARTMON

STOPMON
Detach monitor subtask.

Example: F BKMSTSKA,STOPMON

STARTMB
Start flatcopy task and multilevel backup detection.

Example: F BKMSTSKA,STARTMB

STOPMB
Stop flatcopy task and multilevel backup detection.

Example: F BKMSTSKA,STOPMB

STATUS
Display subtask(s) status.

Example: F BKMSTSKA,STATUS

ULRESM
Unload the Resource Manager.

Example: F BKMSTSKA,ULRESM

LISTBKPS
List current running backups. To learn more about what information will be displayed in the list, see descriptions for BKMH0179I, BKMH0180I, and BKMH0182W messages.

Example: F BKMSTSKA,LISTBKPS

Setting up the Critical Backup Tracking and Inventory started task

Use the information in this topic to configure the Critical Backup Tracking and Inventory started task appropriately for your site.

1. Specify the flat file processing configuration and options:
   a) Use the appropriate option in the **:CBTI_DUMP_BACKUP_PROCESS_OPTIONS** section of the product parameter library's BKMINI member to specify whether you want the flat file data stored in sequential or VSAM files: CBTI_FLATFILE_OPTION=VSAM|FLATFILE

b) Use the appropriate option in the product parameter library's BKMINI member to specify the flat file data set names:

- If you want the flat file data stored in VSAM files, do *one* of the following:
  - In the **:CBTI_DUMP_BACKUP_PROCESS_OPTIONS** section, specify CBTI_FLATFILE_VSAMDS=*dsname*
  - In the product's JCL library BKMSTSKA member, specify the VSAMFF parameter: VSAMFF=*dsname*.
- If you want the flat file data stored in a sequential file, in the **:CBTI_DUMP_BACKUP_PROCESS_OPTIONS** section, specify CBTI_FLATFILE_PREFIX=*prefix*.
- If your installation is configured for VSAM Record Level Sharing (RLS), it is recommended that RLS be used for the VSAM flat file. Refer to the section describing :SI040_VALUES in the configuration values appendix for further information about using RLS for VSAM data sets.

c) In the BKMINI member's **:BKMSTSKA** section, set keyword **PROCESS_FLAT_FILES_INTERVAL** to the appropriate flat file processing interval.

2. Set the appropriate token in the **:BKMSTSKA** section of the product parameter library's BKMINI member to specify that you want the flat file copy task started when the BKMSTSKA started task is started: AUTO_PROCESS_FLAT_FILES=Y

   Note: If you want the flat file data processed by all active instances of BKMSTSKA, use the S BKMSTSKA command to start the started task; however, if you want the flat file data processed by some, but not all, active instances of BKMSTSKA, use the S BKMSTSKA,PARM=FFCOF command to start those tasks that are not to process flat file data.

3. If you want selected instances of BKMSTSKA to use specific VSAM files, you must update the VSAMFF parameter: VSAMFF=*dsname* In the product's JCL library BKMSTSKA member to reflect the VSAM file name.

   Note: By default, all instances of BKMSTSKA use the same VSAM file to contain the flat file data, which is specified by the **CBTI_FLATFILE_VSAMDS** token in the **:CBTI_DUMP_BACKUP_PROCESS_OPTIONS** section of the parameter library's BKMINI member.

4. Set the appropriate token in the :BKMSTSKA section of the product parameter library's BKMINI member to specify that you want the BKMSTSKA started task to wait for current running backups to finish before proceeding to full shut down:

   WAIT_FOR_CURRENT_BACKUPS=*Y*

   If N is specified (default value) the BKMINI started task will proceed to full shut down regardless of any current running backups.

Starting the Critical Backup Tracking and Inventory started task

This information describes how to start the Critical Backup Tracking and Inventory started task.

- If the BKMINI member of the Advanced Backup and Recovery for z/OS parameter library includes CBTI_AUTOSTART = YES (the default setting), issue the following command from the console to start Critical Backup Tracking and Inventory:
  S BKMSTSKA,JOBNAME=*xxx* where *xxx* is the name of the Advanced Backup and Recovery for z/OS started task.
- If you specify CBTI_AUTOSTART=NO in the BKMINI member of the product parameter library, you must first issue the start command as described previously and then issue the following command to start the various tasks:
  F *xxx*,AUTOSTART

  where *xxx* is the name of the Advanced Backup and Recovery for z/OS started task.

  This command starts the individual subtasks that are required for Critical Backup Tracking and Inventory to intercept backups. Typically, AUTOSTART is the only command needed to activate the BKMSTSKA started task.

If you are using the BKMINI VSAM flat file option (CBTI_FLATFILE_OPTION=VSAM), individual BKMSTSKA started tasks can use specific VSAM flat files. To use a specific VSAM flat file, other than the one specified in BKMINI, modify the BKMSTSKA start command to specify the name of the VSAM flat file (VSAMFF=dsname) to be used. The default value VSAMFF=NULLFILE indicates the flat file name or prefix is specified in BKMINI token CBTI_FLATFILE_VSAMDS or CBTI_FLATFILE_PREFIX.

Stopping the Critical Backup Tracking and Inventory started task

This information describes how to stop the Critical Backup Tracking and Inventory started task.

Note: The BKMSTSKA task should not be stopped while it is tracking backup jobs. If the BKMSTSKA task is stopped while the IEFUSI exit module is capturing data from a backup, the initiator that is running the backup job terminates abnormally. To avoid abnormal terminations for the other jobs already started on that initiator, you must restart the initiator. When the BKMSTSKA task is restarted, it resumes tracking backup jobs.

- Issue the command: P *xxx*

  where *xxx* is the name of the Advanced Backup and Recovery for z/OS started task.

  If you specify WAIT_FOR_CURRENT_BACKUPS=*Y* in the BKMINI member of PARMLIB and there are running backups during shut down processing, the BKMSTSKA started task will wait for those backups to finish before proceeding to full shut down. In that case the list of current running backups will be displayed and the WTOR question BKMH0178I will be asked which will allow to cancel the wait at any time. To learn more about what information will be displayed in the list see descriptions for BKMH0179I, BKMH0180I, and BKMH0182W messages.

  To cancel the wait and proceed to full shut down issue the command: R *id,N*

  where *id* is the WTOR ID stated in the previously issued BKMH0178I message. All the backups that were still running will not be intercepted.

  If you choose to wait for running backups to finish, after the last backup is finished, the BKMH0181I message will be displayed and the started task will proceed to full shut down. All backups will be intercepted.

  If you specify WAIT_FOR_CURRENT_BACKUPS=*N* in the BKMINI member of PARMLIB and there are running backups during shut down processing, the list of current running backups will be displayed and the started task will proceed to full shut down. All backups that were still running will not be intercepted.

  All Critical Backup Tracking and Inventory processes are shut down.

VSAM Flat File preventive maintenance

Performing preventive maintenance on the VSAM flat file that is used by Critical Backup Tracking and Inventory for tracking output from backup utilities can prevent error conditions and performance degradation.

How does VSAM flat file processing work?

Understanding how the VSAM flat file is processed might help you to determine what type of preventive maintenance is appropriate for your VSAM flat file.

The Critical Backup Tracking and Inventory component of Advanced Backup and Recovery for z/OS tracks output from various backup utilities. When a supported backup utility is started, information about the backed up data sets, output volumes, and other metadata is written to a flat file. That flat file is the input for the Critical Backup Tracking and Inventory started task. The Critical Backup Tracking and Inventory started task reads the flat file and uses the information from the file to update the Inventory Data Set (IDS). After the IDS has been updated, the records in the flat file are deleted.

The flat file can be a sequential file that is generated for each backup that completes and is deleted after it is read by the Critical Backup Tracking and Inventory started task. Alternatively, the flat file can be a

VSAM file that behaves like a container for the flat file records that are generated by the backup process; the VSAM file is not deleted after it is read by the Critical Backup Tracking and Inventory started task, but the backup information records in the VSAM file are deleted, leaving the file empty.

The VSAM flat file is a key sequenced data set (KSDS) and is subject to all of the benefits and drawbacks of a KSDS VSAM file. In a perfect world, the Advanced Backup and Recovery for z/OS VSAM flat file always has a net record count of zero because all of the backup information records are deleted after the Critical Backup Tracking and Inventory started task processes the file. However, in the real world, incomplete backups (due to equipment failure, abnormal terminations, and other causes) can cause orphaned records to be stranded in the VSAM flat file, even after the Critical Backup Tracking and Inventory started task has processes the file. The orphaning of records is not a common occurrence, but recovery processes were built into the Critical Backup Tracking and Inventory started task to minimize the chance that orphaned records remain in the flat file after it has been processed by the started task.

The repeated addition and deletion of backup information records to and from the VSAM flat file can lead to the loss of Control Areas (CAs), which causes the VSAM flat file to grow. However, even though the VSAM flat file grows in size, the amount of usable space within the file does not grow and might even decrease. Eventually, performance suffers, especially for highly used VSAM flat files.

How do I know whether preventive maintenance is needed?

Over time, you might notice unexpected program behaviors that indicate problems with the VSAM flat file.

Any of these situations are indicative of a VSAM file in need of maintenance:

- The VSAM flat file copying process is experiencing increasingly long run times.
- The VSAM flat file copying process is terminating abnormally due to `OUT OF SPACE` conditions.
- Sites that use CA Disk might experience `DSNINDEX record not found` error messages.

Performing preventive maintenance

There are two methods of performing preventive maintenance on the VSAM flat file.

- If your site runs z/OS V1R12 or later, it is strongly recommended that you implement the `RECLAIMCA` function to reclaim empty Control Areas in real-time processing.
- If your site runs an earlier version of z/OS than V1R12 or `RECLAIMCA` has not been implemented for the VSAM flat file, it is recommended that you replace the VSAM flat file periodically.

Replacing VSAM flat files

If your site cannot take advantage of the RECLAIMCA functionality, it is strongly recommended that you replace the VSAM flat file whenever you experience unexpected behavior from the program as it tracks backups.

The RECLAIMCA function is not available in z/OS releases earlier than the V1R12 release.

1. Stop the Advanced Backup and Recovery for z/OS started task.
2. Determine whether any new Flat File Copy processes have been started within 15 minutes of the last one, then take appropriate action:
   - If there are no new Flat File Copy processes, all backups have been processed by the started task and you can go directly to step 3.
   - Otherwise, use the IDCAMS **REPRO** command to copy the records that remain in the VSAM flat file to a permanent fileCritical Backup Tracking and Inventory and then continue with step 3.
3. Delete the current VSAM flat file.
4. Run `BKMDEFVF` to create a new VSAM flat file. The `BKMDEFVF` job is contained in member `BKMDEFVF` of the Advanced Backup and Recovery for z/OS JCL library.
5. Verify that `BKMDEFVF` completed successfully.
6. If you used IDCAMS **REPRO** to copy the records from the original VSAM flat file, use IDCAMS **REPRO** again to load those records into the new VSAM flat file.

7. Restart the Advanced Backup and Recovery for z/OS started task.

Reclaiming Control Areas

If your site can take advantage of the RECLAIMCA functionality, it is strongly recommended that you enable it for your VSAM flat file.

The RECLAIMCA function became available in the z/OS V1R12 release. When the last record in a Control Area (CA) is erased, RECLAIMCA places empty sequence set and high-level index records on a free list. Subsequently, when the new backup information records are added to the flat file, the reclaimed CA can be used in any situation where the needed CA is smaller than or equal to the size of the reclaimed empty CA.

Note: CAs that became empty prior to the system being upgraded to z/OS V1R12 cannot be reclaimed by the RECLAIMCA function. In this situation, after all sharing systems are at V1R12 or later and have RECLAIMCA implemented, reorganize the existing data sets to remove the unreclaimable CAs.

Three methods of implementing the RECLAIMCA function are described. Use the method that is most appropriate for your site.

- To implement RECLAIMCA at the system level, use one of these methods:
  - Use TSO/E PARMLIB UPDATE to modify member IGDSMSxx by adding CA_RECLAIM({NONE|DATACLAS|DATACLASS}).
  - Use the SMS Console command SETSMS CA_RECLAIM({NONE|DATACLAS|DATACLASS}).

    Note: This method immediately implements RECLAIMCA for all KSDSs.
- Use the IDCAMS ALTER command: ALTER *ksdsname* RECLAIMCA|NORECLAIMCA.

  Note: The **ALTER** command does not immediately enable or disable RECLAIMCA. **ALTER** does update the catalog immediately, but the system does not look at this attribute in the catalog until the next time that the control block structure is built; therefore, RECLAIMCA is not enabled until all current OPENs against the control block structure have been closed and the first subsequent OPEN is issued.

Appendix K. Critical Backup Tracking and Inventory skeleton processing

Skeletons are model JCL and utility control statements that, when merged with data set and volume attributes stored in the Advanced Backup and Recovery for z/OS Inventory Data Set, create output tailored to the requirements for restoring a specific data set or volume.

A default skeleton is associated with each utility or backup option to be used to generate restore JCL and utility control statements.

To construct the restore JCL and utility control statements you run the BKMBLDRJ program. The data sets and volumes to be restored are selected by the parameters that you supply to BKMBLDRJ. From the information stored in the Advanced Backup and Recovery for z/OS Inventory Data Set when data sets and volumes were backed up, Critical Backup Tracking and Inventory determines the utility and in some cases, the option that was used within a utility to perform the backups.

Note:

- To support conditional generation of selected output, and to manage arrays of data, skeleton processing includes IF..THEN..ELSE logic and looping capability.
- The file tailoring process is not aware of what any line in the model represents, that is, it might be JCL, IDCAMS control statements, and so on.

You can use skeletons to generate restore JCL and control statements that reflect your disaster recovery plan strategies.

- Deletion before allocation— For restore operations where the utility does not allocate the target data set, the default skeletons generate a DELETE followed by a DELETE NOSCRATCH prior to allocation of the target data set. This assumes the data set about to be restored should replace any existing data set or an existing catalog entry. The DELETE NOSCRATCH statements are generated to address situations where a catalog entry exists but the corresponding data set is not found.
- VSAM KSDS with AIX — Critical Backup Tracking and Inventory does not track backups of alternate indexes (AIXs) and consequently assumes AIXs will be rebuilt rather than restored. For backup and restore utilities that do not offer a SPHERE option, the default skeletons upon encountering a VSAM base cluster generate defines and rebuilds for any associated AIXs, using the association data saved in the Advanced Backup and Recovery for z/OS Inventory Data Set when the base cluster was backed up. Likewise the default skeletons include the generation of associated PATH statements, again from data saved in the Advanced Backup and Recovery for z/OS Inventory Data Set.

Skeleton guidelines and considerations

When you are working with skeletons, follow these guidelines:

- During product maintenance, new versions of the skeletons will be installed thus overwriting any existing skeletons.
- Because the file tailoring process is unaware of the types of statements being generated, and the intended result of the output, the process cannot determine whether the use of a variable will make sense for the desired output.
- Use IF..THEN... ELSE... logic to assure that variables are appropriate for the type of data set being restored and the type of VSAM structure.
- Some data set attributes are always set to one of some number of valid keywords, such as VSAM REUSE | NOREUSE. These types of variables can usually be placed in a skeleton model output line with no concern.
- For attributes that can have no value, if in doubt about whether the resulting model output line will be valid if no value is substituted, surround the model line with an)IF/)ENDIF statement testing the variable for a NULL value such that the model line will not be generated if the variable value is null.

- Arrays of volume serial numbers can be tricky depending on the required output syntax. For this reason, the distributed skeleton library (SKELLIB) has a number of common skeletons designed to be used in a nested skeleton scenario for generating the required JCL and control statements.
- To avoid continuation problems, wherever possible (such as in situations where parameters are enclosed with parentheses), isolate the open and close parentheses to lines with no other parameters so the lines within the parentheses can always include the continuation character.
- PROCs are not allowed within skeletons.

Generic skeleton commands

Skeleton processing commands are denoted by a right parenthesis.

)COPY ***member name***
Copies the specified member name into the current skeleton before continuing processing. The member being copied can contain)COPY commands but nested)COPY commands can copy the same member only once. That is, the process does not support recursive)COPYs. The limit for nested)COPY statements is 50. Use nested)COPYs to simplify skeletons or ensure that common requirements are generated the same way.)COPY statements should be used instead of PROCs.

)COPYJOBSTMT ***member name***
The COPYJOBSTMT statement must be the first statement in each skeleton member. Use the)COPYJOBSTMT rather than a native JOB statement in skeletons. This allows for the file tailoring process to determine when and if an expanded skeleton should contain a JOB statement. The member name is the name of the member in the skeleton library that contains the JOBCARD skeleton to use if Critical Backup Tracking and Inventory determines a job card is needed. A member specified by the)COPYJOBSTMT command cannot contain another)COPYJOBSTMT command or a)COPY command.

)ELSE
Permits alternate processing when the)IF evaluation is false. An ELSE statement must be located between)IF and)ENDIF statements at the same nesting level. Only one)ELSE is allowed for an)IF level.

)ENDIF
Terminates the)IF statement at the same nesting level.

)ENDLOOP
Terminates a)LOOP structure.

)IF ***var1*****comparator*****value*****var2***
Specifies a relational expression that is evaluated for a true or false condition.

- *var1* specifies a declared variable name.
- *comparator* specifies how *var1* will be compared to *value* or *var2*. For character type variables, valid comparators are: EQ (equal) or NE (not equal). For numeric type variables valid comparators are: EQ (equal), NE (not equal), LT (less than), GT (greater than), LE (less than or equal), or GE (greater than or equal).
- *value* specifies a literal value to be compared to *var1*.
- *var2* specifies a variable to be compared to *var1*.

value, or the contents of *var2*, must be the same data type as *var1*. The relational expression is evaluated for a true or false condition. If the condition is true, the skeleton input records between the)IF and the corresponding)ENDIF or)ELSE are processed. If the condition is false, the records down to the next)ELSE or)ENDIF are skipped. These statements can be nested. There is no nesting limit. The)IF must be terminated by an)ENDIF

)IF)THEN)ELSE
Controls conditional generation of output. The most obvious need for this is already eliminated if the default selection of skeleton member by utility is used. However, even within a given utility it might be necessary to conditionally generate output. If conditional output generation becomes too complex,

another option is to create a specific skeleton and supply it in the BKMBLDRJ parameter list as an override to the default.

)LEAVE
Exits the current loop structure.

)LOOP ON *#DSNCNT|#BVOLCNT|#CVOLCNT|#DVOLCNT|#IVOLCNT*
Provides the ability to generate multiple output records or groups of output records from a single input source. Two types situations could require this: one where a single backup command results in multiple data sets backed up that need to be explicitly specified in a restore, and the other where multiple VOLSERs need to be specified in the restore JCL or control statements.

ON is a required keyword for)LOOP meaning loop on a count. The loop count is specified by supplying one of the control variables (denoted by a pound sign as the first character): #DSNCNT, #BVOLCNT, #CVOLCNT, #DVOLCNT, and #IVOLCNT. Associated with each control variable is a variable to be used in the body of the loop whose value will change during each pass through the loop. All statements between the)LOOP and)ENDLOOP are repeated for each pass through the loop.)LOOP commands can be nested. There is no nesting limit.)LOOP ON does not alter the value of control variables.

)SET#
Sets the value of a control name. The first operand must be one of the control names. The second operand can either be a value or an existing numeric character type variable. For example:)SET# #DSNCNT 20

)SETC *var1value|variable-name*
Declares a new character type variable or changes the value of an existing one. The first operand must be a new or existing character type variable name. The second operand can be either a value or an existing character type variable.

)SETN *var1value|var2*[+|-*n*]
Declares a new numeric type variable or changes the value of an existing one. The first operand must be a new or existing numeric type variable name. The second operand can either be a value or an existing numeric character type variable. Optionally, a value (number) can be add to or subtracted from *var1* after its value has been set or changed. For example:)SETN &STEPNO 1)SETN &MYCOUNT &MYCOUNT + 1

Skeleton control names

Skeleton control names are denoted by a pound sign.

#AIXCNT
Count of the AIXs (alternate indexes) associated with a VSAM base cluster. When used to control a LOOP ON, each time through the loop the VSAM attribute variables will reflect each AIX cluster. The intended restore procedure for VSAM clusters with AIXs is to first define and reload the base cluster followed by a loop that contains the define and rebuild for each AIX. This should be followed by another loop to define all PATHs (see #PATHCNT). In a LOOP ON #AIXCNT the variable &AIXNAME will contain the name of the AIX cluster and variable &CDSNAME will contain the name of the associated base cluster.

#BVOLCNT
Backup volser count. #BVOLCNT contains the count of restore input volumes (output from the backup). When used to control a LOOP ON, each time through the loop the value of &VOLSER will be set to the next volser the data set was backed up to. Use this control variable with &VOLSER in skeletons where you need to specify the restore input volumes.

#CVOLCNT
Cluster volser count. #CVOLCNT contains the count of VSAM cluster volumes where the cluster original resided. When used to control a LOOP ON, each time through the loop the value of &CVOLUMES will be set to the next volser where the cluster of concern was allocated.

Note: Although provided, control variables and associated volume variables are supplied for the data and index components if you want to restore the data and index components to the same respective volumes where they were originally allocated.

#DSNCNT
Data set name count. Count of data sets backed up by the current utility command. When used to control a LOOP ON, each time through loop the value of &DSNAME will be set to the next backed-up data set name from the array of names saved in the Advanced Backup and Recovery for z/OS Inventory Data Set.

For example, when restoring from a FDR or DSS logical dump:

```
//FILTDD   DD*
   INCLUDE(   -
   )LOOP ON #DSNCNT
      &DSNAME
   )ENDLOOP
          )
```

#DVOLCNT
Data component volser count. #DVOLCNT contains the count of VSAM data component volumes where the original data component resided. When used to control a LOOP ON, each time through the loop the value of &DVOLUMES will be set to the next volser where the data component of concern was allocated.

#IVOLCNT
Index component volser count. #IVOLCNT contains the count of VSAM index component volumes where the original index component initially resided. When used to control a LOOP ON, each time through the loop the value of &IVOLUMES will be set to the next volser where the data component of concern was allocated.

#PATHCNT
Count of the PATHs associated with AIXs for a VSAM base cluster. When used to control a LOOP ON, each time through the loop the PATH variables will reflect each PATH associated with an AIX. The intended restore procedure for VSAM clusters with AIXes is to first define and reload the base cluster, followed by a loop that contains the define and rebuild for each AIX, followed by a loop to define all PATHs.

#VOLCNT
Non-VSAM volser count. (Valid for non-VSAM data sets only.) #VOLCNT contains the count of non-VSAM volumes where the original data set resided. When used to control a LOOP ON, each time through the loop, the value of &VOLSER will be set to the next volser where the data set of concern was allocated. Use this control variable with &VOLSER in skeletons where you want to allocate the output from the restore to the same volumes where the data set originally resided.

Skeleton variables

A basic function of the file tailoring process is the ability to substitute a value for a variable specified in a skeleton model line. Variables are denoted by an ampersand (&) as the first character.

Note the following:

- Many variables are set by Critical Backup Tracking and Inventory using data set attributes stored from the backup in the Advanced Backup and Recovery for z/OS Inventory Data Set. These variables names are fixed. Not all of these variables might be set to a valid value depending on the applicability for the data set of concern.
- Variables can be defined and set to a value using the generic skeleton)SETC and)SETN commands.
- Any variable name referenced in a skeleton must have been previously declared. All documented Critical Backup Tracking and Inventory variables are declared but might not be set to a valid value if not appropriate for the type of data set being processed.

Generic skeleton variables

This information describes the generic skeleton variables.

Note: In addition to the comparators shown, NULL can be used to test for a variable with no assigned value.

&ACCOUNTING EQ|NE
Accounting data (32 bytes).

&ADALOADLIB EQ|NE
LINKLIST | name of Adabas LOADLIB For decision about whether to insert a STEPLIB in Adabas Restore jobs. LINKLIST indicates no STEPLIB.

&DSNTYPE EQ|NE
A typical use of &DSNTYPE is to vary the statements generated by a skeleton based on the type of data set being restored. See the SKELLIB member REPRO01 for an example of allocation differences between types of VSAM structures and non-VSAM data sets.

Supported data set types are:

- ALIAS
- ESDS
- GDGBASE
- GDS
- HFS
- KSDS
- LDS
- NONVSAM
- PATH
- PDS
- PDSE
- RRDS
- USERCAT
- VRRDS

&EXPDT EQ|NE
YYYYDDD Julian date for the EXPDT parameter for the TO parameter of IDCAMS. Places the same expiry date on the restored data set as the original data set.

&EXPDTJCL EQ|NE
YYYY/DDD Julian date for the EXPDT parameter of JCL. Places the same expiry date on the restored data set as the original data set.

&FDRLOADLIB EQ|NE
LINKLIST | name of FDR LOADLIB Specifies whether to insert a STEPLIB in FDR restore jobs. LINKLIST indicates no STEPLIB.

&NXTVOLSER EQ|NE
Contains the next volume serial of a list or an asterisk (*) which indicates the next volume is a candidate volume.

&NEWHLQ EQ|NE
High-level qualifier to be used when performing an Critical Backup Tracking and Inventory restore with a new name. The value of this variable comes from the BKMINI token RENAME_HLQ. If RENAME_HLQ = *, the value of this variable will be null.

&REPLACE EQ|NE
Y | N For data set restores, specifies whether or not the restore delete the original data set.

&SELECTDSN EQ|NE
Y | N For selective data set restore from a full volume dump.

&SMS EQ|NE
Y | N Test &SMS wherever a difference is required in output statements, depending on whether the data set being restored was originally SMS managed or not.

Tip: Testing &SMS EQ Y is one way to ensure other SMS-only related variables are used only when appropriate.

&USERCAT EQ|NE
The user catalog where the data set of concern was originally cataloged.

&VOLSER EQ|NE
The volume serial a non-VSAM data set originally resided on (input to backup). Use &VOLSER in conjunction with #VOLCNT to restore a data set to the same volumes it originally resided on, such as for non-SMS managed data sets where you wish to allocate to specific volumes rather than using an esoteric.

Skeleton source file variables

Skeleton source file variables contain information relating to the data set and volumes input to the backup utility tracked.

Note: In addition to the comparators shown, NULL can be used to test for a variable with no assigned value. Member names cannot be restored, only data sets.

&BDEVNAME EQ|NE
Device name.

&BDSNAME EQ|NE
Backup output data set name.

&BFILENO EQ|NEGT|GELT|LE
File number (first volume).

&BMEMBER EQ|NE
Member name (if a PDS).

&BVERSION EQ|NEGT|GELT|LE
Version number (HSM backup only).

&BVOLCNT EQ|NEGT|GELT|LE
Number of volumes in &BVOLUMES.

&BVOLUMES EQ|NE
All source volumes.

Skeleton cluster variables

Skeleton cluster variables pertain to the information available at the cluster level for a VSAM data set. Some attributes are available at both the cluster and component level. Distributed default skeletons that allocate VSAM data sets opt to define attributes at the data and index levels rather than at the cluster level where there is a choice.

Note: In addition to the comparators shown, NULL can be used to test for a variable with no assigned value.

&CAIXNAME EQ|NE
AIX name related to base cluster.

&CAIXSUFX EQ|NE
AIX cluster name suffix. This is the AIX cluster name minus the high level qualifier.

&CALLOCTYPE EQ|NE
BLK | CYL | TRK

&CBUFFERSPACE EQ|NEGT|GELT|LE
Buffer size.

&CBWO EQ|NE
TYPECICS | TYPEIMS | NO

&CCISIZE EQ|NEGT|GELT|LE
CI size.

&CDATACLAS EQ|NE
SMS data class.

&CDEVNAME EQ|NE
Generic device name (for example: 3390).

&CDSNAME EQ|NE
Base cluster name.

&CDSNSUFX EQ|NE
Cluster name suffix. This is the cluster name minus the high level qualifier.

&CERASE EQ|NE
ERASE | NOERASE.

&CEXCEPEXIT EQ|NE
Exception exit name.

&CFREESPACECI EQ|NEGT|GELT|LE
CI free space.

&CIMBED EQ|NE
IMBED | NOIMBED.

&CINHIBIT EQ|NE
INHIBIT | UNINHIBIT.

&CKEYLEN EQ|NEGT|GELT|LE
Key length.

&CKEYOFFSET EQ|NEGT|GELT|LE
Key offset.

&CMAXLRECL EQ|NEGT|GELT|LE
Maximum LRECL.

&CMAXRECORDS EQ|NEGT|GELT|LE
Maximum record (RRDS).

&CMGMTCLAS EQ|NE
SMS management class.

&CORDERED EQ|NE
ORDERED | UNORDERED.

&COWNER EQ|NE
Owner.

&CREPLICATE EQ|NE
REPLICATE | NOREPLICATE.

&CREUSE
REUSE | NOREUSE.

&CSHROPT1 EQ|NE
Cross region share option.

&CSHROPT2 EQ|NE
Cross system share option.

&CSPACEPRI EQ|NEGT|GELT|LE
Primary space. Critical Backup Tracking and Inventory cannot determine whether the original data set was allocated with space specified for the entire cluster as opposed to the data and index components. Therefore, it is recommended that the data and index space values be used for allocating the target data set of a restore.

&CSPACESEC EQ|NEGT|GELT|LE
Secondary space. Critical Backup Tracking and Inventory cannot determine whether the original data set was allocated with space specified for the entire cluster as opposed to the data and index components. Therefore, it is recommended that the data and index space values be used for allocating the target data set of a restore.

&CSPANNED EQ|NE
SPANNED | NONSPANNED.

&CSPEED EQ|NE
SPEED | RECOVERY.

&CSTORCLAS EQ|NE
Storage class.

&CUPGRADE EQ|NE
NOUPGRADE | UPGRADE.

&CVOLCNT EQ|NEGT|GELT|LE
Number of volumes in &CVOLUMES.

&CVOLUMES EQ|NE
All cluster volumes.

&CWRITECHECK EQ|NE
WRITECHECK | NOWRITECHECK.

&VSAMTYPE EQ|NE
INDEXED | LINEAR | NUMBERED | NONINDEXED.

Note: &DSNTYPE can also be used to determine the structure of a VSAM data set.

Skeleton path variables

This information describes the skeleton path variables.

Note: In addition to the comparators shown, NULL can be used to test for a variable with no assigned value.

&PATHENTRYNAME EQ|NE
Path entry name.

&PATHESUFX EQ|NE
Path entry name suffix. This is the path entry name without the high level qualifier.

&PATHNAME EQ|NE
Path name.

&PATHSUFX EQ|NE
Path name suffix. This is the path name without the high level qualifier.

&PATHOWNER EQ|NE
Path owner.

&PATHUPDATE EQ|NE
UPDATE | NOUPDATE.

Non-VSAM skeleton variables

This information describes the skeleton non-VSAM skeleton variables.

Note: In addition to the comparators shown, NULL can be used to test for a variable with no assigned value.

&ACCOUNTING EQ|NE
Accounting information.

&ALLOCTYPE EQ|NE
BLK | CYL | TRK.

&BLKSIZE EQ|NEGT|GELT|LE
Block size.

&DEVNAME EQ|NE
Generic device name (for example: 3390) for original data set.

&DIRBLKS EQ|NEGT|GELT|LE
Directory blocks for PDSs

&DSNAME EQ|NE
Backup input data set name.

&DSNSUFX EQ|NE
Backup input data set name suffix. This is the backup input data set name minus the high level qualifier.

&DSORG EQ|NE
Data set organization. Valid values are:

- DA
- IS
- PO
- PS
- U
- VE
- VK
- VL
- VR
- VS
- VV

&GDSSTATUS
ACTIVE | DEFERRED | ROLLEDOUT.

&LRECL EQ|NEGT|GELT|LE
LRECL.

&MGMTCLAS EQ|NE
SMS Management class.

&OWNER EQ|NE
Owner.

&RECFM EQ|NE
Record format. Valid values are:

- F
- FA
- FAS
- FB
- FBA
- FBAM
- FBAS
- FBM
- FBMS
- FBS
- FM
- FMS
- FS
- V
- VA

- VAS
- VB
- VBA
- VBAM
- VBAS
- VBM
- VBMS
- VBS
- VM
- VMS
- VS
- U
- UBA
- UBAM
- UBAS
- UBM
- UBMS
- UBS

&SPACEPRI EQ|NEGT|GELT|LE
Primary space.

&SPACESEC EQ|NEGT|GELT|LE
Secondary space.

&STORCLAS EQ|NE
Storage class.

&VOLCNT EQ|NEGT|GELT|LE
Number of volumes in &IVOLUMES.

&VOLUMES EQ|NE
All cluster volumes.

Non-VSAM alias skeleton variables

Variables are supplied that could be used to recreate non-VSAM aliases and GDG base catalog entries. However, it is not uncommon to initially establish catalogs in disaster recovery scenarios such that all aliases and GDG base definitions are established as part of the initial conditioning of the catalogs.

Note: In addition to the comparators shown, NULL can be used to test for a variable with no assigned value.

&ALIASNAME EQ|NE
Alias name. This variable only supports a single alias. If multiple aliases exist for a true name &ALIASNAME will be set to the last alias name found in the association record.

&ALIASRELATED EQ|NE
Data set alias is related to (truename). &DSNAME1 can also be used to specify the related true name if defining an alias.

GDG base skeleton variables

Variables are supplied that could be used to recreate non-VSAM aliases and GDG base catalog entries. However, it is not uncommon to initially establish catalogs in disaster recovery scenarios such that all aliases and GDG base definitions are established as part of the initial conditioning of the catalogs.

Note: In addition to the comparators shown, NULL can be used to test for a variable with no assigned value.

&GDGEMPTY EQ|NE
EMPTY | NOEMPTY attribute captured from the GDG base catalog record at the time of the backup.

&GDGLIMIT EQ|NEGT|GELT|LE
GDG limit value captured from the GDG base catalog record at the time of the backup.

&GDGSCRATCH EQ|NE
SCRATCH | NOSCRATCH attribute captured from the GDG base catalog record at the time of the backup.

Skeleton event fields

This information describes the skeleton event fields.

Note: In addition to the comparators shown, NULL can be used to test for a variable with no assigned value.

&ADADBNAME EQ|NE
Adabas database name.

&ADADB# EQ|NE
Adabas database number.

&ADAMODE EQ|NE
Mode specified at backup time. Valid modes are: M (Multi) or S (Single).

&ADADEVICE EQ|NE
Device name specified at backup time.

&ADASVC EQ|NE
SVC number specified at backup time.

&ADAAMODE EQ|NE
AMODE at backup time.

&EABARSGEN EQ|NE
Relative generation of an ABARS backup tracked by ABARS Manager.

&EABARSSTACK EQ|NE
STACK | NOSTACK option specified for an ABARS backup tracked by ABARS Manager.

&EABARSUNIT EQ|NE
HSM unit name for an ABARS backup tracked by ABARS Manager.

&EAGGNAME EQ|NE
ABARS aggregate name if data is to be restored from an aggregate backup.

&ECMDNUMBER EQ|NEGT|GELT|LE
Command number. Where the execution of a backup utility includes multiple commands, Critical Backup Tracking and Inventory internally numbers each command beginning with the number one (1).

&EDAD EQ|NE
DSN of the Advanced Backup and Recovery for z/OS DAD data set. Used only when necessary to insert the DAD file name in restore JCL.

&EDEVNAME EQ|NE
Device name for volume that has been dumped.

&EINVDB EQ|NE
DSN of Advanced Backup and Recovery for z/OS Inventory Data Set. Used only when necessary to insert the Advanced Backup and Recovery for z/OS Inventory Data Set file name in restore JCL.

&EJOBNAME EQ|NE
Job name of the backup event.

&EPARMLIB EQ|NE
DSN of Advanced Backup and Recovery for z/OS product parameter library data set. Used only when necessary to insert the P file name in restore JCL.

&EPROCNAME EQ|NE
Proc name of the backup event.

&EREADERDATE EQ|NE
Reader date (SMF date) of the backup event.

&EREADERTIME
Reader time (SMF time) of the backup event.

&ERETCODE1 EQ|NEGT|GELT|LE
Return code from the backup utility.

&ERETCODE2 EQ|NEGT|GELT|LE
Return code from the tracking of the utility by Critical Backup Tracking and Inventory.

&ESTEPDATE
Start date of the backup step.

&ESTEPNAME EQ|NE
Step name of the backup.

&ESTEPTIME
Start time of the backup step.

&ESYSID EQ|NE
System ID of the image where the backup was executed.

&EVOLSER EQ|NE
Source volser for volume dumps.

&LOADLIB EQ|NE
DSN of the Advanced Backup and Recovery for z/OS LOAD1 data set. Used only when necessary to insert the LOAD1 file name in restore JCL.

Skeleton index components

This information describes the skeleton index components.

Note: In addition to the comparators shown, NULL can be used to test for a variable with no assigned value.

&IAIXNAME EQ|NE
AIX name related to base cluster.

&IAIXSUFX EQ|NE
Index component name suffix. This is the index component name without the high level qualifier.

&IALLOCTYPE EQ|NE
BLK | CYL | TRK

&IBUFFERSPACE EQ|NEGT|GELT|LE
Index buffer space.

&ICLUSTER EQ|NE
Cluster name.

&ICISIZE EQ|NEGT|GELT|LE
CI size.

&IDEVNAME EQ|NE
Generic device name (for example: 3390).

&IDSNAME EQ|NE
Index component name.

&IDSNSUFX EQ|NE
Data component name suffix. This is the data component name minus the high level qualifier.

&IERASE EQ|NE
ERASE | NOERASE.

&IEXCEPEXIT EQ|NE
Exception exit name.

&IFREESPACECI EQ|NEGT|GELT|LE
CI free space.

&IKEYLEN EQ|NEGT|GELT|LE
Key length.

&IKEYOFFSET EQ|NEGT|GELT|LE
Key offset.

&IMAXLRECL EQ|NEGT|GELT|LE
Maximum LRECL.

&IOWNER EQ|NE
Owner.

&ISHROPT1 EQ|NE
Cross region share option.

&ISHROPT2 EQ|NE
Cross system share option.

&ISPACEPRI EQ|NEGT|GELT|LE
Primary space.

&ISPACESEC EQ|NEGT|GELT|LE
Secondary space.

&IVOLCNT EQ|NEGT|GELT|LE
The number of volumes in &IVOLUMES.

&IVOLUMES
All cluster volumes.

Skeleton data components

This information describes the skeleton data components.

Note: In addition to the comparators shown, NULL can be used to test for a variable with no assigned value.

&DAIXNAME EQ|NE
AIX name related to base cluster.

&DAIXSUFX EQ|NE
Data component name suffix. This is the data component name without the high level qualifier.

&DALLOCTYPE EQ|NE
BLK | CYL | TRK

&DBUFFERSPACE EQ|NEGT|GELT|LE
Data buffer space.

&DCLUSTER EQ|NE
Cluster name.

&DCISIZE EQ|NEGT|GELT|LE
CI size.

&DDEVNAME EQ|NE
Generic device name (for example: 3390).

&DDSNAME EQ|NE
Data component name.

&DDSNSUFX EQ|NE
Data component name suffix. This is the data component name minus the high level qualifier.

&DERASE EQ|NE
ERASE | NOERASE.

&DEXCEPEXIT EQ|NE
Exception exit name.

&DFREESPACECI EQ|NEGT|GELT|LE
CI free space.

&DKEYLEN EQ|NEGT|GELT|LE
Key length.

&DKEYOFFSET EQ|NEGT|GELT|LE
Key offset.

&DMAXLRECL EQ|NEGT|GELT|LE
Maximum LRECL.

&DOWNER EQ|NE
Owner.

&DSHROPT1 EQ|NE
Cross region share option.

&DSHROPT2 EQ|NE
Cross system share option.

&DSPACEPRI EQ|NEGT|GELT|LE
Primary space.

&DSPACESEC EQ|NEGT|GELT|LE
Secondary space.

&DVOLCNT EQ|NEGT|GELT|LE
The number of volumes in &DVOLUMES.

&DVOLUMES
All cluster volumes.

Appendix L. Automated Critical Data Identification utilities

Automated Critical Data Identification utilities enable you to rename, copy, or delete applications managed by Automated Critical Data Identification. Automated Critical Data Identification also provides a utility that can be used to update Automated Critical Data Identification filters.

Using the BKMMNT01 utility to rename an application

The BKMMNT01 utility renames an Automated Critical Data Identification application.

Before renaming an application, make sure the new application name does not currently exist.

1. Copy member BKMMNT01 from the JCL library.
2. Edit the BKMMNT01 member you copied in step 1 (an example is included in the member).
3. Run BKMMNT01 to rename the Automated Critical Data Identification application.
4. From the **Application AutoBuild Options** menu, select the **SETUP** option and change the ABARS aggregate name (if applicable) and Selection Data Sets.
5. Run BKMJRFSH to update the job table.

Using the BKMAPPCP utility to copy an application

The BKMAPPCP utility copies application setup information from a source application to one or more target applications. Job collection and filter information are not copied from the source application.

Note:

- **ABARS users**. An ABARS aggregate definition must be created and active in SMS for the new or copied target aggregate before running the BKMAPPCP copy program. This provides Automated Critical Data Identification with the ability to find the new Selection Data Set name in the SMS address space and automatically place it into the new, or copied, target application so it does not need to be manually changed later.
- **Non-ABARS users**. You cannot insert a new Selection Data Set name during the copying process.
- The BKMAPPCP program only supports the use of the same Selection Data Set name for both RSP and Verification.

1. Copy member BKMAPPCP from the product JCL library.
2. Specify the applications you want to copy using the syntax:

   ```
   COPY APPLICATION|APP(source target1 target2 target3..)
   ```

   where *source* specifies the source of the Setup options. *target* specifies the new application that will be created using the source application setup option data. Multiple applications can be specified. For example:

   ```
   COPY  APPL             -
         (                        -
          APPL0001                -     SOURCE APPLICATION
          APPL0002                -     TARGET APPLICATION
          APPL0003                -     TARGET APPLICATION
         )
   ```

3. After BKMAPPCP has completed and the Setup options have been copied, perform the normal steps including job collection, BKMJRFSH, BKMAPLEN, and optional filters.

Using the BKMDELAP utility to delete an application

The BKMDELAP utility deletes an application and all its related records in the Automated Critical Data Identification Application Data Inventory database (including filter entries).

Note: If you have a large number of deletions (more than a few applications) run BKMDELAP during a period of low activity.

1. Copy member BKMDELAP from the product JCL library.
2. Specify the application name using the syntax:
   `PARM= (applname)`, where *applname* specifies the application to delete.
3. Run BKMDELAP to delete the Automated Critical Data Identification application.
4. Run BKMJRFSH to update the job table.

Using the BKMFILTR utility to perform batch filter processing

The BKMFILTR utility batch filter processing uses commands and keywords to update BKMFILTR filters.

1. Copy member BKMFILTR from the JCL library.
2. Customize the job card and specify FILTERS keywords as input to the BKMFILTR utility program to define filter actions.
3. Submit the BKMFILTR job.

FILTERS command keywords

All filter actions must begin with the **FILTERS** keyword in the command.

Table 7 on page 574 provides a reference to the FILTERS command keywords.

Table 7. FILTERS command keyword summary. The FILTERS command keywords, a list of valid keyword values for each, and a description of the keyword. Columns can include multiple rows whenever multiple keyword values require explanation.

FILTERS command keyword	Keyword values	Keyword description
ADD_FIRST	None.	The ADD_FIRST keyword specifies the filters are that are to be added *before* existing filters. When ADD_FIRST keyword is specified, new filters are added before existing filters. **Note:** The ADD_FIRST keyword is mutually exclusive with the REPLACE keyword.
ADD_LAST	None.	The ADD_LAST keyword specifies the filters that are to be added after existing filters. When ADD_LAST keyword is specified, the new filters are added *after* existing filters. **Note:** This keyword is mutually exclusive with the REPLACE keyword.

Table 7. FILTERS command keyword summary. The FILTERS command keywords, a list of valid keyword values for each, and a description of the keyword. Columns can include multiple rows whenever multiple keyword values require explanation. *(continued)*

FILTERS command keyword	Keyword values	Keyword description
APPLICATION(*applname*)	Application name.	The APPLICATION keyword specifies Local filters. All parameters following the APPLICATION keyword specify Local (application level) filters. **Note:** The APPLICATION keyword is mutually exclusive with the UNIVERSAL and GLOBAL keywords.
EVALUATION(*evaluation filters*)	Evaluation filters: • Eval_DSN(*filter*) • Eval_Job(*filter*) • Eval_Vol(*filter*) • Eval_Unit(*filter*) • Eval_Sysid(*filter*) • Eval_Pgm(*filter*) • Incl_DSN(*filter*) • Incl_Job(*filter*) • Incl_Vol(*filter*) • Incl_Unit(*filter*) • Incl_Sysid(*filter*) • Incl_Pgm(*filter*) • Excl_DSN(*filter*) • Excl_Job(*filter*) • Excl_Vol(*filter*) • Excl_Unit(*filter*) • Excl_Sysid(*filter*) • Excl_Pgm(*filter*)	The EVALUATION keyword specifies the filter type. See Chapter 15, “Filters and controls,” on page 145 for more information. **Note:** This keyword is mutually exclusive with the RETENTION and EXTERNAL keywords.
	• Alloc_DSN(*filter*) • Accomp_DSN(*filter*) • Global_DSN • Global_Job • Global_Vol • Global_Unit • Global_Sysid • Global_Pgm	These values can only be used with the APPLICATION keyword. The Alloc or Accomp type keyword values are added to the application evaluation filter set. The Global keyword values enable the specified set of global filters for the application.
EVALUATION_DDN(*ddname*)	The ddname of a filter list.	Filter lists can be stored in a data set specified by the EVALUATION_DDN and EVALUATION_DSN keywords. The data set LRECL must be 80 and the filters must start in column 2.
EVALUATION_DSN(*dataset name*)	The data set name of a filter list.	

Table 7. FILTERS command keyword summary. The FILTERS command keywords, a list of valid keyword values for each, and a description of the keyword. Columns can include multiple rows whenever multiple keyword values require explanation. *(continued)*

FILTERS command keyword	Keyword values	Keyword description
EVALUATION_DEL(*parm value*)	Delete retention filters: • *All* • *DSN* • *Job* • *Vol* • *Unit* • *Sysid*	*DSN*, *Job*, *Vol*, *Unit*, and *Sysid* are only valid for Global filters. For all other deletes, the value *All* is required.
EXTERNAL(*external filters*)	External filters: • Incl_EXTDSN(*filter*) • Alloc_EXTDSN(*filter*) • Accomp_EXTDSN(*filter*)	The EXTERNAL keyword specifies the filter type. **Note:** This keyword and its values can only be used with the APPLICATION keyword.
EXTERNAL_DDN(*ddname*)	The ddname of a filter list.	Filter lists can be stored in a data set specified by the EXTERNAL_DDN and EXTERNAL_DSN keywords. The data set LRECL must be 80 and the filters must start in column 2.
EXTERNAL_DSN(*dataset name*)	The data set name of a filter list.	
EXTERNAL_DEL(*parm value*)	*All*.	The EXTERNAL_DEL keyword deletes all external filters. **Note:** This keyword can only be used with the APPLICATION keyword.
GLOBAL	None.	All parameters following the GLOBAL keyword specify global filters. **Note:** This keyword is mutually exclusive with the UNIVERSAL and APPLICATION keywords.
JOB_COLLECT(*job collection filters*)	Job collection filters: • Incl_Job(*filter*) • Excl_Job(*filter*)	The JOB_COLLECT keyword specifies job collection processing. Job collection filters are only used in job collection processing. **Note:** This keyword and values can only be used with the APPLICATION keyword.
JOB_COLLECT_DSN(*dataset name*)	The data set name of a filter list.	Filter lists can be stored in a data set. The data set LRECL must be 80 and the filters must start in column 2.
JOB_COLLECT_DDN(*ddname*)	The ddname of a filter list.	
JOB_COLLECT_DEL(*parm value*)	*All*.	Delete External filters. **Note:** This keyword can only be used with the APPLICATION keyword.

Table 7. FILTERS command keyword summary. The FILTERS command keywords, a list of valid keyword values for each, and a description of the keyword. Columns can include multiple rows whenever multiple keyword values require explanation. *(continued)*

FILTERS command keyword	Keyword values	Keyword description
OVERVIEW	None.	The Overview report shows all the filters in the Automated Critical Data Identification Application Data Inventory database that apply to a specific application. Universal filters are shown and Global filters are shown for the Global types enabled for that application.
REPLACE	None.	When the REPLACE keyword is specified, the new filter list replaces existing filters in Automated Critical Data Identification Application Data Inventory database. **Note:** This keyword is mutually exclusive with the ADD_FIRST and ADD_LAST keywords.
RETENTION(*retention filters*)	Retention filters: • Discard_DSN(*filter*) • Discard_Job(*filter*) • Discard_PGM(*filter*) • Discard_Vol(*filter*) • Discard_Unit(*filter*) • Discard_Sysid(*filter*) • Keep_DSN(*filter*) • Keep_PGM(*filter*) • Keep_Job(*filter*) • Keep_Vol(*filter*) • Keep_Unit(*filter*) • Keep_Sysid(*filter*)	Specifies the retention filter type. These values can be used with UNIVERSAL, GLOBAL, or APPLICATION keywords. **Note:** This keyword is mutually exclusive with the EVALUATION and EXTERNAL keywords.
	• Global_DSN • Global_Job • Global_Vol • Global_Unit • Global_Sysid • Global_Pgm	These values enable the specified global retention filter set for the application. **Note:** These values can only be used with the APPLICATION keyword.
RETENTION_DSN(*dataset name*)	The data set name of a filter list.	Filter lists can be stored in a data set. The data set LRECL must be 80 and the filters must start in column 2.
RETENTION_DDN(*ddname*)	The ddname of a filter list.	

Table 7. FILTERS command keyword summary. The FILTERS command keywords, a list of valid keyword values for each, and a description of the keyword. Columns can include multiple rows whenever multiple keyword values require explanation. *(continued)*

FILTERS command keyword	Keyword values	Keyword description
RETENTION_DEL(*parm value*)	Delete retention filters: • *All* • *DSN* • *Job* • *Vol* • *Unit* • *Sysid*	Delete retention filters. *DSN*, *Job*, *Vol*, *Unit*, and *Sysid* are only valid for Global filters. For all other deletes, the value *All* is required.
UNIVERSAL	None.	All parameters following the UNIVERSAL keyword specify Universal filters. **Note:** This keyword is mutually exclusive with the GLOBAL and APPLICATION keywords.

Batch filter processing examples

This information provides batch filter processing syntax examples.

Example: Delete Universal Retention filters

This example deletes all existing Universal Retention filters:

```
//FILTERS.XYXIN  DD  *
      FILTERS            -
      UNIVERSAL          -
      RETENTION_DEL(     -
         ALL             -
      )
```

Example: Delete Global DSN Evaluation filters

This example deletes all existing GLOBAL_DSN filters:

```
//FILTERS.XYXIN DD  *
      FILTERS            -
      GLOBAL             -
      EVALUATION_DEL(    -
         DSN             -
      )
```

Example: Update Local (application) Evaluation filters

This example shows the replacement of the current application filter list with a new list:

```
//FILTERS.XYXIN DD  *
      FILTERS                    -
      APPLICATION(Test42)        -
      EVALUATION(                -
         INCL_JOB(TESTJOB1)      -
         EXCL_JOB(TESTJOB*)      -
         GLOBAL_DSN              -
         EVAL_DSN(LOCAL.**.TEST) -
           )                     -
      REPLACE
```

Example: Update Global DSN and job Evaluation filters

The first part of this example replaces any GLOBAL_JOB evaluation filters with the three filters shown. The second part inserts the two GLOBAL_DSN filters ahead of any existing Global Evaluation DSN filters.

```
//FILTERS.XYXIN DD *
      FILTERS -
      GLOBAL -
      EVALUATION( -
         INCL_JOB(TESTJOB1) -
         EXCL_JOB(TESTJOB*) -
         EVAL_JOB(PRODJOB) -
            ) -
      REPLACE
      FILTERS -
      GLOBAL -
      EVALUATION( -
         INCL_DSN(PROD.DATASET.X1) -
         EXCL_DSN(PROD.DATASET.**) -
            ) -
      ADD_FIRST
```

Example: Update Universal Retention filters (using DATASET)

This example adds a Universal Retention DSN filter after any current Universal Retention DSN filters.

```
//FILTERS.XYXIN  DD  *
      FILTERS               -
      UNIVERSAL             -
      RETENTION_DSN(        -
         UNIV.FILTER.DATASET -
            )               -
  ADD_LAST
```

Example: Update Local (application) Retention Filters (Using DDNAME) and External filters and Overview

The first part of this example replaces any Application (local) Retention filters with the filters found in the specified data set. The second FILTERS command replaces any external DSN references for application TEST42 with the filter shown. The final command generates a filter overview report for that application.

```
//RFILTER DD DSN= TEST.FILTER.DATASET,DISP=SHR
//FILTERS.XYXIN DD *
      FILTERS                    -
      APPLICATION(TEST42)        -
      RETENTION_DDN(             -
         RFILTER                 -
         )                       -
      REPLACE
      FILTERS                    -
      APPLICATION(TEST42)        -
      EXTERNAL(                  -
INCL_EXTDSN(PROD.TEST42.XTA(0)   -
         )                       -
      REPLACE
      FILTERS                    -
      APPLICATION(TEST42)        -
      OVERVIEW
```

Appendix M. Sample job collection procedures

These cataloged job collection procedures can be found in the product JCL library. Symbolic parameters in the cataloged procedures are automatically resolved with values provided in theBKMINI product parameter library member.

ASG-Zeke job collection procedure

This information provides a listing of the BKMJOBZK job collection procedure for ASG-Zeke.

Sample

```
//BKMJOBZK PROC ACDIDB=prefix.value.ACDIDB,
//       DISK=SYSALLDA,
//       JOBSSPCE=(CYL,(10,10),RLSE),
//       LOAD=prefix.value.SBKMLOAD,
//       PARMLIB=prefix.value.SBKMPARM,
//       SOUT='*',
//       ZEKEJOBS=ACDI.ZEKE.JOBS
//S1      EXEC PGM=&ZPGM,PARM=&APPL
//STEPLIB  DD  DSN=&LOAD,DISP=SHR
//ACDIDB   DD  DSN=&ACDIDB,DISP=SHR
//INI      DD  DSN=&PARMLIB(BKMINI),DISP=SHR
//ZEKEJOBS DD  DSN=&ZEKEJOBS,DISP=SHR
//JOBS     DD  DSN=&&JOBS,UNIT=&DISK,DISP=(,PASS),
//             LRECL=110,RECFM=FB,SPACE=&ZEKESPCE
//SYSPRINT DD  SYSOUT=&SOUT
//SYSUDUMP DD  SYSOUT=&SOUT
//        PEND
```

CONTROL-M job collection procedure

This information provides a listing of the BKMCM606 job collection procedure for CONTROL-M.

Sample

```
//BKMCM606 PROC DISK=SYSALLDA,
//       JOBSSPCE=(CYL,(10,10),RLSE),
//       LOAD=prefix.value.SBKMLOAD,
//       SOUT='*'
//* ---------------------------------------------------------------
//* THIS PROCEDURE IS USED WHEN BUILDING JOB SCAN JCL USING THE PANELS
//* ---------------------------------------------------------------
//S1      EXEC PGM=&MPGMI,PARM=&APPL
//STEPLIB  DD  DSN=&LOAD,DISP=SHR
//CTLMIN   DD  DSN=&RPT,DISP=SHR
//JOBS     DD  DSN=&&JOBS,UNIT=&DISK,DISP=(,PASS),
//             LRECL=110,RECFM=FB,SPACE=&JOBSSPCE
//SYSPRINT DD  SYSOUT=&SOUT
//SYSUDUMP DD  SYSOUT=&SOUT
//        PEND
```

Cybermation ESP job collection procedure

This information provides a listing of the BKMESPEX job collection procedure for Cybermation ESP.

Sample

```
//BKMESPEX PROC DISK=SYSALLDA,
//       ESPJOBS=PDABC.ESP.REPORT,
//       JOBSSPCE=(CYL,(10,10),RLSE),
//       LOAD=prefix.value.SBKMLOAD,
//       SOUT='*'
//STEP01  EXEC PGM=IEFBR14
```

```
//DD02     DD  DSN=&ESPRPT,DISP=(MOD,DELETE),SPACE=(0,0),
//             UNIT=&DISK
//STEP04  EXEC PGM=&EPGMI             ACDI EXTRACTION
//STEPLIB  DD  DISP=SHR,DSN=&LOAD
//ESPJOBS  DD  DISP=SHR,DSN=&ESPJOBS
//ESPRPT   DD  DSN=&&ESPRPT,UNIT=&DISK,DISP=(,PASS),
//             SPACE=&JOBSSPCE
//ASPPRINT DD  SYSOUT=&SOUT
//SYSUDUMP DD  SYSOUT=&SOUT
//ABNLIGNR DD  DUMMY                   DO NOT REMOVE IF USING ABENDAID
//        PEND
```

TWS job collection procedure

This information provides a listing of the BKMTWJOB job collection procedure for TWS.

Sample

```
//BKMTWJOB PROC ACDIDB=?hlq.ACDIDB,
//       BATCHOPT=BATCHOPT,
//       DISK=SYSALLDA,
//       EQQAD=NULLFILE,
//       EQQMLIB=NULLFILE,
//       EQQPARM=NULLFILE,
//       JOBSSPCE=(CYL,(10,10),RLSE),
//       LOAD=BKM0202X.QABETA.LOAD,
//       PARMLIB=?hlq.SBKMPARM,
//       SOUT='*'
//*******************************************************************
//* STEP AXR00: OBTAIN LIST OF TWS/OPC APPLICATION IDS
//*******************************************************************
//AXR00   EXEC PGM=EQQBATCH,PARM='EQQAXR00',REGION=2M
//EQQMLIB  DD  DISP=SHR,DSN=&EQQMLIB
//EQQPARM  DD  DISP=SHR,DSN=&EQQPARM(&BATCHOPT)
//ADREPORT DD  DSN=&&REPORT,DISP=(,PASS),SPACE=(CYL,(5,1)),
//             UNIT=&DISK,
//             DCB=(RECFM=FBA,LRECL=121,BLKSIZE=6050)
//ADWKIN   DD  DCB=(LRECL=220,BLKSIZE=4400,RECFM=FB),
//             SPACE=(CYL,(3,1)),DISP=(,DELETE),UNIT=&DISK
//ADWKOUT  DD  DCB=(LRECL=220,BLKSIZE=4400,RECFM=FB),
//             SPACE=(CYL,(3,1)),DISP=(,DELETE),UNIT=&DISK
//ADWKWK01 DD  SPACE=(CYL,(3,1)),UNIT=&DISK
//ADWKWK02 DD  SPACE=(CYL,(3,1)),UNIT=&DISK
//ADWKWK03 DD  SPACE=(CYL,(3,1)),UNIT=&DISK
//EQQADDS  DD  DISP=SHR,DSN=&EQQAD
//SYSIN    DD  DISP=SHR,DSN=&PARMLIB(EQQAXR00)
//EQQMLOG  DD  SYSOUT=&SOUT
//SYSPRINT DD  SYSOUT=&SOUT
//SYSOUT   DD  SYSOUT=&SOUT
//EQQDUMP  DD  SYSOUT=&SOUT
//SYSMDUMP DD  DUMMY
//*******************************************************************
//* STEP TWS0: MATCH TWS/OPC APPLICATION NAMES TO THOSE FOR THE
//*            ACDI APPLICATION
//*******************************************************************
//TWS0    EXEC PGM=BKMTWSAP,PARM=&APPL,COND=(0,NE)
//STEPLIB  DD  DISP=SHR,DSN=&LOAD
//ASPINVDB DD  DISP=SHR,DSN=&ACDIDB
//RPTIN    DD  DISP=(OLD,DELETE,DELETE),DSN=*.AXR00.ADREPORT
//TWSAPPLS DD  DSN=&&TWSAPP,UNIT=&DISK,
//             DISP=(,PASS),SPACE=(CYL,(1,1),RLSE),
//             RECFM=FB,LRECL=80
//SYSPRINT DD  SYSOUT=&SOUT
//SYSUDUMP DD  SYSOUT=&SOUT
//*******************************************************************
//* STEP ADPRT: EXTRACT TWS APPLICATION DETAIL INFORMATION
//*******************************************************************
//ADPRT   EXEC PGM=EQQBATCH,PARM='EQQADPRT',REGION=2M,COND=(0,NE)
//EQQMLIB  DD  DISP=SHR,DSN=&EQQMLIB
//EQQPARM  DD  DISP=SHR,DSN=&EQQPARM(&BATCHOPT)
//ADREPORT DD  DSN=&&RPT,DISP=(,PASS),UNIT=&DISK,
//             SPACE=(CYL,(5,1)),
//             DCB=(RECFM=FBA,LRECL=121,BLKSIZE=6050)
//ADREPRT2 DD  DCB=(RECFM=FBA,LRECL=121,BLKSIZE=6050),
//             SPACE=(CYL,(3,1)),DISP=(NEW,DELETE),UNIT=&DISK
//ADPRIN   DD  DCB=(LRECL=200,BLKSIZE=3200,RECFM=FB),
//             SPACE=(CYL,(3,1)),DISP=(,DELETE),UNIT=&DISK
//ADPROUT  DD  DCB=(LRECL=200,BLKSIZE=3200,RECFM=FB),
```

```
//             SPACE=(CYL,(3,1)),DISP=(,DELETE),UNIT=&DISK
//ADPRWK01 DD  SPACE=(CYL,(3,1)),UNIT=&DISK
//ADPRWK02 DD  SPACE=(CYL,(3,1)),UNIT=&DISK
//ADPRWK03 DD  SPACE=(CYL,(3,1)),UNIT=&DISK
//EQQADDS  DD  DISP=SHR,DSN=&EQQAD
//EQQAD2DS DD  DISP=SHR,DSN=&EQQAD
//ADUSERDS DD  DISP=SHR,DSN=*.TWS0.TWSAPPLS  APPLICATION NAMES
//SYSIN    DD  DISP=SHR,DSN=&PARMLIB(EQQADPRT)
//EQQMLOG  DD  SYSOUT=&SOUT
//SYSPRINT DD  SYSOUT=&SOUT
//SYSOUT   DD  SYSOUT=&SOUT
//EQQDUMP  DD  SYSOUT=&SOUT
//EQQDMSG  DD  SYSOUT=&SOUT
//SYSMDUMP DD  DUMMY
//*******************************************************************
//* STEP TWS1: EXTRACT THE LIST OF JOBS FOR THE ACDI APPLICATION
//*******************************************************************
//TWS1    EXEC PGM=BKMTWSJB,PARM='(&APPL,&TWSDATE)',COND=(0,NE)
//STEPLIB  DD  DISP=SHR,DSN=&LOAD
//INI      DD  DISP=SHR,DSN=&PARMLIB(BKMINI)
//RPTIN    DD  DISP=(OLD,DELETE,DELETE),DSN=*.ADPRT.ADREPORT
//JOBS     DD  DSN=&&JOBS,UNIT=&DISK,DISP=(,PASS),
//             LRECL=110,RECFM=FB,SPACE=&JOBSSPCE
//SYSPRINT DD  SYSOUT=&SOUT
//SYSUDUMP DD  SYSOUT=&SOUT
//        PEND
```

Unicenter CA-7 job collection procedure

This information provides a listing of the BKMC7JOB job collection procedure for Unicenter CA-7.

Sample

```
//BKMC7JOB PROC DISK=SYSALLDA,
//       JOBSSPCE=(CYL,(10,10),RLSE),
//       LOAD=?hlq.SBKMLOAD,
//       SOUT='*'
//S1      EXEC PGM=BKMC7JOB,PARM=&APPL
//STEPLIB  DD  DSN=&LOAD,DISP=SHR
//CA7IN    DD  DSN=&RPT,DISP=SHR
//JOBS     DD  DSN=&&JOBS,UNIT=&DISK,DISP=(,PASS),
//             LRECL=110,RECFM=FB,SPACE=&JOBSSPCE
//SYSPRINT DD  SYSOUT=&SOUT
//SYSUDUMP DD  SYSOUT=&SOUT
//        PEND
```

Unicenter CA-Jobtrac job collection procedure

This information provides a listing of the BKMJBTCV job collection procedure for Unicenter CA-Jobtrac.

Sample

```
//BKMJBTCV PROC DISK=SYSALLDA,
//       JOBSSPCE=(CYL,(10,10),RLSE),
//       LOAD=?hlq.SBKMLOAD,
//       SOUT='*'
//S1      EXEC PGM=BKMJBTCV,PARM=&APPL
//STEPLIB  DD  DSN=&LOAD,DISP=SHR
//SELJOBS  DD  DSN=&SELJOBS,DISP=SHR
//JBTJOBS  DD  DSN=&JBTJOBS,DISP=SHR
//JOBS     DD  DSN=&&JOBS,UNIT=&DISK,DISP=(,PASS),
//             LRECL=110,RECFM=FB,SPACE=&JOBSSPCE
//SYSPRINT DD  SYSOUT=&SOUT
//SYSUDUMP DD  SYSOUT=&SOUT
//        PEND
```

Pre-staged data set job collection procedure

This information provides a listing of the BKMPRECV job collection procedure for ISPF Edit Mode and Job Pre-staged Dataset (batch mode).

Sample

```
//BKMPRECV PROC DISK=SYSALLDA,
//       JOBSSPCE=(CYL,(10,10),RLSE),
//       LOAD=?hlq.SBKMLOAD,
//       SOUT='*'
//S1      EXEC PGM=BKMPRECV,PARM=&APPL
//STEPLIB  DD  DSN=&LOAD,DISP=SHR
//PREJOBS  DD  DSN=&PRESTAGE,DISP=SHR
//JOBS     DD  DSN=&&JOBS,UNIT=&DISK,DISP=(,PASS),
//             LRECL=110,RECFM=FB,SPACE=&JOBSSPCE
//SYSPRINT DD  SYSOUT=&SOUT
//SYSUDUMP DD  SYSOUT=&SOUT
//        PEND
```

Appendix N. Relative date masking

Where permitted, relative date masking enables you to specify date search criteria relative to the current date.

Relative date masks use these characters:

Character	Usage
*	Represents the current date
+	Inclusive
-	Exclusive
n	Represents the number of days (1-999)

Note: When a date field is left blank, all dates are displayed. If you specify a specific date, it must be in the format YYYY/DD/MM or MM/DD/YYYY.

You can specify date criteria using these relative date mask formats:

Mask	Meaning
*	Display events that occurred today.
*+*n*	(**Full Volume Dump Search Setup** panel and **Jobname List Search Setup** panel.) Display all generation 0 events from today through *n* days from today. All generations of each event are displayed regardless of the date. **Note:** On the **History** panel, this format displays all events equal to today through *n* relative days from today.
*-*n*	Display items *n* days from the current date.

Examples

Specify `*` to display events from today.

Specify `*-1` to display events from yesterday.

Specify `*+1` to display events from today and yesterday.

On the **Full Volume Dump Search Setup** panel, specify `*+4` to display all events, from today through four days from today, with each of their subsequent generations (regardless of the date).

Specify `*-4` to display events four days from today.

Appendix O. Data set naming conventions

There are some restrictions on the names you can use forAdvanced Backup and Recovery for z/OS data sets.

You can use a hyphen (-) as a character in a data set name; however, when you use a hyphen as the last character in a data set name, it might be mistaken for a continuation symbol and failures sometimes occur. This restriction is also documented in IBM's IBM document DFSMShsm Storage Administration Reference.

Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

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, MD-NC119
Armonk, NY 10504-1785
US

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

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 jurisdictions 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 Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

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.

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.

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 illustrate 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.

Trademarks

IBM, the IBM logo, and ibm.com® are trademarks or registered marks 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: http://www.ibm.com/legal/copytrade.shtml.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions:

Applicability: These terms and conditions are in addition to any terms of use for the IBM website.

Personal use: You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

Commercial use: You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Rights: Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

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.

Index

Special Characters

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Z

Product Number: 5698-B17

SC23-9730-04