Rocket | Mainstar.

white paper

# Basic Concepts of ICF Catalog Backup and Recovery

Rocket®

# Basic Concepts of ICF Catalog Backup and Recovery.

A White Paper by Ronald K. Ferguson

Version 1 March 2013

Performing frequent and timely backups of data files is universally recognized as a prudent course of action. We take backups for local recovery in the event of file corruption or loss. We take disaster recovery backups in the event our local environment is no longer usable. One aspect of backup that is becoming more important in z/OS environments is metadata backup, or more simply put, backing up the control information data that lets us locate and access our actual data files.

## introduction

Performing frequent and timely backups of data files is universally recognized as a prudent course of action. We take backups for *local* recovery in the event of file corruption or loss. We take *disaster recovery* backups in the event our local environment is no longer usable. One aspect of backup that is becoming more important in z/OS environments is ***metadata*** backup, or more simply put, backing up the *control information data* that lets us locate and access our actual data files.

Metadata for a z/OS system comes in many forms: disk volume VTOCs and the VTOC index or DFSMShsm's (HSM) control data sets (MCDS, BCDS, OCDS), the tape management catalog, and ICF catalogs. We will address the importance of frequent backups of your ICF catalogs, specifically the BCS. We will also consider the steps that are necessary to ensure a successful *recovery* in the event of a catalog failure.

# ICF catalog structures

An ICF catalog is comprised of two physical data set structures. The Basic Catalog Structure (BCS) is commonly referred to as 'the catalog'; a 'user catalog' when it catalogs application data sets, or a 'master catalog' when it catalogs system data sets. The second structure is the VSAM Volume Data Set (VVDS).

The BCS is where all data sets are cataloged in today's z/OS environment, and indeed, with SMS implemented, it has become a requirement that all data sets be cataloged. Usually, an installation will have several BCSs, ranging in number from one to five, for a modest-sized data center, to hundreds in a large data center. Using a facility called 'alias match', where the alias name of a user catalog matches the high level qualifier of a data set's name, this identifies the particular user catalog (BCS) where individual data sets will be cataloged. The two most important pieces of information contained in the catalog for a data set are the volume serial number on which the data set resides and the address pointer to locate the VVDS record.

For VSAM data sets, the VVDS is very similar to a VTOC in information content. There is one VVDS for each disk volume that contains VSAM data sets. On SMS-managed volumes there is a VVDS regardless of whether VSAM data sets are present on the volume. For each VSAM data set component, there is a VVDS record called a VVR that describes the physical data set attribute, statistics, and volume information. The information in the VVR is used to open a VSAM object from a program-issued OPEN macro. On SMS-managed volumes, there are NVR records in the VVDS, one for each non-VSAM data set on the volume, containing primarily SMS-related class information for the data set (the VTOC F1 record for the non-VSAM data set is used to open it).

The most common types of records in the BCS consist of:

- **Cluster Sphere Record**—one for each VSAM base cluster that is cataloged in the BCS. If the cluster has an alternate index, the cluster sphere record also contains all catalog information about the alternate index. The key of the Cluster Sphere Record is the base cluster data set name, and this is the name specified on the DD statement DSNAME parameter that is used to locate the base cluster.
- **Truename Record**—one for each VSAM data set component, containing an association pointer to the base cluster. The key of the Truename Record is the component's actual data set name. The purpose of a Truename Record is to provide access to the individual component, for whatever reason you might have (for example, to access the data component of a KSDS, as if it was an ESDS, in the event of an index failure). As a rule, Truename Records are infrequently accessed for day-to-day application program execution. Furthermore, because there is no Truename record for the data component of the BCS, you can't open the data component of a BCS as an ESDS in order to recover from a broken catalog (therefore you do need a vendor product like Rocket Mainstar **Catalog RecoveryPlus** ).
- **Path Record**—usually one for each alternate index that a VSAM cluster has, identifying the base and its related alternate index. The key of the Path Record is the path name and it's this name that is specified on the DD statement DSNAME parameter that is used to locate the alternate index.
- **Non-VSAM Record**—one for each non-VSAM data set that is cataloged in the BCS. This record points to the VOLSER and VTOC location, so that the data set can be located and opened for processing.

The BCS is a fairly standard VSAM KSDS in structural layout and usage. Its key is 45 bytes (data set name, plus a 45th byte that is used as an extension record count in the event the record length exceeds the maximum allowed for the BCS). As a KSDS, the BCS has a data component and an index component. The BCS is 'cataloged' within itself. (Note: The BCS has a cluster name of 45 bytes of x'00', resulting in the BCS's cluster sphere record always occupying the first record within itself, and therefore, the record is easily obtainable—this record is called the BCS's self-describing record.)

The VVDS is also a fairly standard VSAM structure, being an ESDS in physical layout and usage. Its CISZ is always 4096, enabling fairly efficient space utilization for the relatively small VVR and NVR records. The VVDS is 'special' in the sense of how Catalog Management provides for space management within the VVDS. It utilizes a space map of free areas within each CI, and this space map is contained in a special control record, the VVCR, occupying the first CI within the VVDS.

## BCS Failure Risks

Since virtually all data sets in a z/OS environment are cataloged, losing a BCS can be a disastrous situation if you don't have a backup, or if you don't know how to restore the backup and perform a Forward Recovery to bring it up to date. While it is commonly accepted these days that BCSs hardly fail anymore, there still remains a fairly constant risk that a BCS will be damaged to the point where a recovery is necessary.

Here are a few failure types that you should be aware of:

- Lack of knowledge in how to do a restore and Forward Recovery in the event of a failure. In a survey of 28 z/OS installations, 15 of the responders (over half) admitted that their installation had never tested a BCS recovery! Ten installations (over a third) had tested a BCS recovery only once. A grand total of three responders actually test their BCS recovery plan on a regular basis! For the 25 (out of 28 total) installations that have never tested (or have only tested once), it's almost a certainty they will experience great difficulty if ever faced with a real recovery situation. At best, the recovery will take a lot longer than expected because you will be learning how to do the recovery at exactly the worst time. To ensure this doesn't happen at your installation, know how to effect a BCS recovery, set up job streams ahead of time to perform a recovery, document the recovery process, and test it often. If you perform a disaster recovery test at least twice a year—why not test your BCS Forward Recovery plan at least twice a year?
- BCS backups that prove to be bad, when you need them the most. In our survey, 19 installations indicated they don't currently run an IDCAMS EXAMINE INDEXTEST on their BCSs prior to the backup. As a result, the BCS can have a broken, but undetected, index structure, and the result can be an incorrect or incomplete backup. This is particularly true if you're using IDCAMS EXPORT to perform the backup, from which you'll most likely receive a RC=0. The most frequent cause of this situation is the dreaded 'SSI broken horizontal chain pointer' problem. To ensure this doesn't happen at your installation, always run an EXAMINE INDEXTEST on each BCS prior to the backup and if you receive a RC>0, investigate it and correct any problems. If you're using **Catalog RecoveryPlus** BACKUP to perform your BCS backups, an EXAMINE INDEXTEST is run by default prior to each BCS backup.

As a telling anecdote, one installation told us they run EXAMINE INDEXTEST prior to their BCS backups, and on one BCS, EXAMINE gives back a RC=8, indicating a serious error. In the next step, though, the BCS is backed up with EXPORT, getting a RC=0, but since it's getting the good RC from EXPORT, they figure the bad RC from EXAMINE must be spurious. When we helped them to examine the BCS's problem, we found that they had been taking a bad backup all along, but thankfully, had never had to recover the BCS from the backup.

- In addition to EXAMINE INDEXTEST, you should also consider the EXAMINE DATATEST command as well. This command 'structure-checks' the CI and CA blocking architecture of the BCS, ensuring that every CI and CA within it is sound, that catalog records within the CIs can be accurately de-blocked and accessed, and that there isn't any garbage lurking within the structure. It also checks for out-of-sequence and duplicate key records. Since this command is relatively expensive (in terms of time) to run, you might want to schedule it for every BCS on a monthly or bi-monthly basis. If you're using **Catalog RecoveryPlus** BACKUP to perform your BCS backups, an EXAMINE DATATEST can be run by explicit specification of the EXAMINE(DATATEST) keyword on the BACKUP command.
- To achieve even greater BCS integrity checking, there is a further command, IDCAMS DIAGNOSE BCS, that will go inside each BCS record to determine if the data content is correct. For example, it will check each base cluster record to determine if all expected components are found, whether cells and fields contain valid data, and whether all cluster sphere records match with truename records and vice versa. This is the most detailed command test; therefore it is the most costly in time and I/O resources. If you're using **Catalog RecoveryPlus** BACKUP to perform your BCS backups, you can explicitly specify the DIAGNOSE-BCS keyword, and an IDCAMS DIAGNOSE BCS command will be issued 'under the covers' for any BCS processed with this BACKUP command.
- Missing or incomplete SMF data. A Forward Recovery of the BCS is contingent upon having all relevant SMF records available, from the exact time of the backup, through to the time of the recovery. It's imperative that you've been collecting the correct records, and that you can lay your hands on them when you need them. For a BCS Forward Recovery, the type 61, 64, and 65 records are necessary. These indicate every data set define, delete, and extension to a new volume.
- Locating the backup file of your BCS that requires recovery. Your day can be completely ruined if the BCS that needs to be restored is the one that contains the catalog record for the BCS backup files, and you haven't foreseen that problem. The solution is to manually keep a record of the backup file's location, but that can be a real pain in the neck in keeping track of backup volumes. At the very least, you should ensure that the backup file is not cataloged in the BCS being backed up. If it is, and that catalog is not accessible, locating your backup file may be 'problematic', to put it delicately. Have more than one BCS backup copy, in the event that a backup is bad. Your backup 'system' should provide for several generations of backup, so it is most likely that you'll need to make it a GDG. That way, if you find that a

backup is mysteriously bad, you can revert to the previous backup. However, it could be that whatever is causing your backups to go bad is systemic, and previous backups might also be bad. Another option is to produce duplex copies of your backups. **Catalog RecoveryPlus** can accomplish this by simply specifying multiple data set or DD names on the output specification, and BACKUP will write multiple copies.

## BCS forward recovery logic

A backup of a BCS and a subsequent restore from the backup in the event of a BCS failure are what you need the most to get up and running again. BCSs are an ever-changing structure, though, and every time you delete, define, or extend to a new volume any data set through the BCS, there is an update of some kind to one or more BCS records. Therefore, if you took the BCS backup at midnight and at 4:00 PM the next afternoon the BCS has a major failure, the restore of the midnight backup will obviously back-level the BCS to its exact status at midnight. If this backup is restored, you'll be able to correctly access *most* of your data sets cataloged in the BCS, but any data sets defined, deleted, or extended after the midnight backup will not be accessible (or at least, not correctly accessible).

SMF records that describe all of these events are created at the time of each define, delete, or data set extension. What's needed is a way to 'apply' the information in these SMF records, to update the BCS to current status at 4:00PM. There is no facility in IDCAMS to perform this Forward Recovery on the BCS, and this is where a catalog recovery product comes in, such as **Catalog RecoveryPlus**, with its RECOVER command that has a FORWARD keyword. This command will read the BCS backup file, and merge the appropriate SMF records as the restore is taking place. In this way, the BCS is updated from the time of the midnight backup, bringing it forward to the time of the failure and restore.

Why can't IDCAMS DIAGNOSE be used for this? It can, to a certain extent, but you'd better have lots of time on your hands, and you'd better not have any tape data sets in the BCS that need to be forward recovered. First, you'd have to run DIAGNOSE in a blizzard of directions, comparing the newly restored BCS against the VVDS on *every* volume to which it is 'related'. This can be hundreds of volumes, particularly since today's SMS environment makes large disk storage pools commonplace and data sets are allocated widely across the storage pool. You then have to run another DIAGNOSE in the other direction, from every VVDS that you DIAGNOSEd outwards towards the BCS in question. Then, you'd have to pour over potentially hundreds of pages of output listings, identifying and analyzing every error message, and from that analysis build your own 'fix' to correct each and every problem. Keep in mind that we're talking about diagnosing outwards to VVDSs, which are disk-oriented structures, so any tape data sets that were defined, deleted, or extended between the backup and restore time cannot be identified in this manner. Simply put, using DIAGNOSE is not the way to achieve a Forward Recovery. The fact is a vendor product that will provide this support is absolutely vital in order to achieve a BCS recovery.

There is one point of caution when getting ready for a BCS restore and Forward Recovery. If you're going to personally issue the IDCAMS DELETE command to delete the failing BCS in preparation for the recovery, make sure that you code the RECOVERY parameter on the command. This tells IDCAMS that you are deleting the BCS for recovery purposes, and that cataloged data sets in the BCS *are not to be deleted*. Do not use the FORCE keyword unless you are absolutely certain of what you are doing. As a further suggestion, consider that it might be better to keep the failed BCS as it is, so that you can further explore what caused the failure (and therefore, prevent it from happening again). To allow this, you need to have the capability to define a BCS of a new name, and restore and forward recover the backup into this newly named BCS (**Catalog RecoveryPlus** can do that with the NEWNAME parameter on the RECOVER command).

## BCS backup frequency

How often should you back up a BCS? Unfortunately, there is not a standard answer to that question. All things considered, it might be a good recommendation to consider setting up a production job that will minimally back up every user and master catalog in your environment on a daily basis. That way, the backup will be fairly up-to-date if it ever needs to be restored, and the number of SMF records that have to be located, extracted, and processed will likely be reasonable. To get a feel for how frequently BCS backups are performed, we conducted a survey. In the 36 responses we received, the following BCS backup frequencies were indicated:

| | |
|---|---|
| Every 4 hours | 1 |
| Every 6 hour | 3 |
| Every 8 hour | 4 |
| Every 12 hours | 5 |
| Once a day | 20 |
| Once every two days | 1 |
| Once a week | 2 |

As you can see, most (55%) perform a daily backup of their BCSs and over a third back up multiple times per day.

We also asked how long the backups take (in elapsed time), and the majority indicated that it was no more than five to ten minutes. As you might expect, most people continue to have their BCSs 'in use' at the time of the backup, and therefore, the backups will be 'fuzzy', meaning that updates to a BCS might have been done between the time the backup started and ended. Forward Recovery with SMF data can handle this, though, so there shouldn't be any problem with a fuzzy backup.

## conclusion

Our survey found that virtually everyone is backing up their BCSs (user catalogs), and that's good news. What isn't good news is that many people aren't making serious efforts to ensure their backups are reliable and accurate. Most troubling is that even fewer people have developed any concerted efforts to assure themselves they can do a restore and Forward Recovery if (and when) the eventual need arises.

You should consider your ICF catalogs to be at the very top of the priority list for critical structures, as without them, there's just no access to your data. If you have a disaster recovery plan that you've developed and test on a regular basis, you should certainly have an ICF catalog backup *and recovery* plan that is tested just as frequently. Without it, you may be courting a disaster of your own making!

mainstar.rocketsoftware.com

mainstar@rocketsoftware.com