A VSAM/ICF Catalog
Technical Paper
From Mainstar Software Corporation

Why Catalog RecoveryPlus?

Part 1: ICF Catalog Reliability

By Ronald K. Ferguson

Preface: *ICF catalogs are a critical metadata structure of your z/OS environment. You undoubtedly have hundreds of thousands, possibly millions, of data sets on your mainframe system, virtually all of them cataloged in just a handful of ICF catalogs. Therefore, even though ICF catalog outages aren't very frequent, an outage on any catalog can affect your access to many data sets, affecting multiple applications. The result can be serious business disruption and lost data.*

An ICF catalog outage can be caused by physical or logical errors in either the BCS or VVDS structure of the catalog. An unplanned re-org can cause an outage when a BCS or VVDS unexpectedly fills up and errors occur during update or access.

Many outages can be avoided, and when an outage does occur, the impact and recovery time can be reduced if good catalog maintenance and management principles are adhered to.

In a series of articles, Ron Ferguson will tell you how to accomplish better reliability, availability, and visibility (RAV) of your ICF catalogs. Each article will demonstrate an aspect of Mainstar: Catalog RecoveryPlus (CR+), Mainstar's product for ICF catalog backup/recovery, diagnostic, and repair.

ICF Catalog Reliability – What Is Required?

Many of today's z/OS systems require 24x7 uptime, as the mainframe system has become the data server for critical business functions. Since all data in a z/OS system must be cataloged to be accessible, and all data sets for an application are cataloged in a single catalog, the ICF catalog environment becomes a single point of failure – where an unplanned outage of a single ICF catalog causes serious disruption for one or more applications.

User catalogs contain the access information for user data sets (differing from a master catalog, which is identical in format and structure to a user catalog, but contains access information about the z/OS system data sets) are typically opened by the system's catalog management software, operating within the Catalog Address Space (CAS). After a system IPL, a user catalog is opened within CAS by the first task that requires a user data set to be located. Unless explicitly closed by an operator CAS command (MODIFY . . . CLOSE), each open user catalog remains open until the z/OS system is shut down (which could be days or weeks later). Throughout that time, whenever any data set cataloged in a catalog needs to be located (for example, to process a DD statement containing the data set's name), the system's data set allocation routines search the catalog, using the data set name as the search key. That's why the catalog is a KSDS - keyed on data set names.

If a catalog has an outage, whether planned or unplanned, all data set locates, data set open and close processes, new data set allocations, deletions of existing data sets, and data set allocation extensions will be unsuccessful for the duration of the outage. In other words, all processing for the application(s) assigned to the user catalog suffer an outage for the duration of the catalog's outage. This places serious importance on catalog reliability.

Since the mid-1990s, ICF catalog reliability has improved to the point that many people who support z/OS systems hold the belief that ICF catalogs never break, never suffer outages, and have inherent stability and reliability. While it is true that ICF catalogs have a high degree of reliability, outages definitely do occur, and with greater frequency than most people realize.

The internal structure of an ICF catalog has changed very little from 20 years ago. The most

significant structural changes were introduced with SMS (System Managed Storage) cells containing the SMS class information.

What has changed significantly is the software code that accesses, updates, and manages the ICF catalog environment. The number of z/OS systems at a typical installation has grown dramatically, increasing the number of systems sharing access to a catalog. As anyone in the software business knows, any software code change can introduce integrity and reliability challenges to the structure(s) they support, and the ICF catalog environment is no different.

Five Ways to Increase Catalog Reliability

Keep an Eye Out for "CATBREAKER" APARs

Early warning intelligence is crucial. As often as you can, search IBMLINK for APARs containing the code words “CATBREAKER” or “DSBREAKER”. The former identifies APARs specifically known to break ICF catalogs (either BCS or VVDS); the latter are APARs known to break VSAM data set structures (the BCS is a KSDS, and the VVDS is an ESDS, so they break like any other VSAM data set). At any time, there are 15 to 20 APARs containing "CATBREAKER" – scan them to determine which are relevant for your current or planned release of z/OS. If they indicate PTFs are available, install them as soon as possible; if not, eliminate the listed circumstances that break catalogs.

Spread Your Catalog Load

These days, more catalogs are better than fewer. Many years ago, before the Catalog Address Space (CAS) was introduced, when storage within the user address space was constrained, most installations opted for the fewest number of catalogs. Today, this constraint is no longer valid, but the opinion that the fewer catalogs the better has persisted. In many cases, there's no longer anyone working in the environment who knows why there is a particular number of catalogs – and they’re afraid to risk changing it. The result is: too many data sets, cataloged in too few catalogs, exposing too many applications when the inevitable outage occurs on a catalog.

What to do? Bite the bullet and organize a project to analyze your current catalog environment; identify which applications will be affected in the event of an outage, and assess the risk to your critical business functions. If you currently have 20 user catalogs, there’s nothing wrong with expanding that to 40 (or whatever number makes sense). If you have 100 aliases assigned to a single catalog, for example, (just making up numbers here) break that down into 10 aliases assigned to 10 catalogs.

You’ll undoubtedly run into resistance from various directions - applications, operations, management - but when the project is completed, it will be well worth it.

Give the Same Priority to Catalog Reliability as You Do to Your Disaster Recovery Plan

Your installation almost surely has a disaster recovery (DR) plan. You almost certainly practice that plan at least once a year. Consider how small the risk is that you’ll ever experience a business disruption that will actually trigger a move to your DR site – yet, your installation spends many millions of dollars, each and every year on your DR strategy.

Now consider the risks associated with your ICF catalog environment. The number of catalogs cataloging every data set across your entire z/OS environment is typically in the two-digit range. Lose one catalog and your business disruption could easily be as severe as any natural or man-made disaster that you’re preparing for with your DR plan.

Finally, consider how little thought and effort is expended on your ICF catalog recovery plan – most likely, close to zilch! Sure, you probably have an old set of catalog recovery procedures lying around somewhere – but when was the last time you actually tested an honest-to-goodness forward recovery of a catalog – as if it had failed 12 hours after the backup, and now you have to 'bring it forward' to current time, with the type of complexities you’ll find in a production system (some simple test in a tiny test environment doesn't count).

In other words, given that an ICF catalog failure – a real failure – could be as serious as any disaster, shouldn't you give the same priority to ICF catalog recovery as you do to your overall DR plan?

Keep Your ICF Catalog Environment Clean

Most real-world ICF catalog environments are disasters waiting to happen, with structural and logical errors scattered throughout; yet they still

run – for now anyway. One beautiful design aspect that IBM incorporated into VSAM is its resiliency – similar to how a World-War-II era B-17 could make it home with hundreds of bullet holes in the airframe. However, many B-17s didn't make it home, when they took a hit in just the wrong location or were too far out to make the final glide.

An ICF catalog outage is the 'hit in just the wrong location'. You could have a broken index record for months or years, and the catalog would still run. If you're not paying attention, you might not know anything about it. Suddenly one day, you need to do something out of the ordinary, or catalog management needs to access in just the area where the error exists – and kaboom! – the catalog fails. That's the beauty – and the treachery of VSAM ICF catalogs.

The solution is to keep your ICF catalogs squeaky clean – both BCSs and VVDSs – by running thorough and regularly-scheduled diagnostics, and most importantly, fixing the identified errors.

A quick (and true) story helps illustrate this. The catalog support team at an (unnamed) installation knew they had a broken index in a catalog where the failure would cause an unending loop in IDCAMS LISTCAT whenever it hit the broken section. Rather than fix the catalog's broken index, they created a rule that a LISTCAT on more than one data set name should never be run. This worked fine until, during a VSAM course I was teaching their COBOL programmers, a student was asked at a break to run a LISTCAT on a particular VSAM data set. The student came back from the break with a surprising statement: "The LISTCAT hasn't finished". A half-hour later, there was an urgent page over the PA system for one of their systems programmers, stating that system such-and-such was down. Turns out, the innocent LISTCAT for a single data set was mistakenly submitted as a LISTCAT ALL, sending IDCAMS into a report loop, filling up the MVS console buffers with the "EXCEEDED LINES" message, and taking down an entire system. With the right tools in hand (and this installation did have them), it would have taken ten minutes to fix the index much earlier, but no one took the time to do it. As a result, the subsequent outage was far worse than it should have been.

The moral is: learn the various diagnostics necessary to identify all types of problem situations, test for them as best you can, and be prepared for as many types of failures as you can.

Know Where to Get Help When You Need It Most

Almost every unplanned catalog outage will be new to you. How you react to it, how well you know what you're doing, and how familiar you are with the tools at hand, will likely determine how severe the outage is, and how long it'll take to recover. Often, the crucial factor is knowing who to call for help – someone who knows what they're doing, knows the structure of catalogs like the back of their hand, and knows the ramifications of what they're doing.

Two stories will illustrate this, both are about Mainstar, and both have to do with our legendary Technical Support.

The first story occurred on Thanksgiving day – a day that most people in the U.S. aren't at work. Luckily for a systems programmer at an installation in Paris, the 24x7 support at Mainstar was available when he accidentally deleted the index of a production ICF catalog. This poor chap had no idea what to do, but since he was evaluating **CR+** at the time, he contacted our Tech Support. At first, e-mails flew back and forth, but instant messenger proved faster, so they moved to that. The Mainstar **CR+** Level 3 Tech Support staff soon grasped the problem, had the guy allocate a dummy (i.e., empty) catalog on the same volume, and after first checking out a highly experimental procedure on the Mainstar system, demonstrated how to 'zap' the name of the dummy index to equal that of the accidentally deleted index, then back up the data component of the catalog with **CR+** and immediately restore it. Presto! The problem was solved. It all happened so fast the guy in Paris was astonished – and grateful – to be back up and running after an incident that would have required hours with his only known recovery techniques.

The second story occurred on a Saturday night; a catalog failure at an installation that didn't have any Mainstar software products – but they had recently attended a Mainstar seminar, and felt more comfortable calling Mainstar Tech Support than the vendor for their current ICF catalog tool. Supporting an installation that didn't even have Mainstar's **CR+** product was not an issue. The 3 to 4 people on the phone with Mainstar for upwards of 2 hours saw the recovery of the failed catalog – using an ICF catalog tool that wasn't Mainstar's. (This particular installation now

licenses **CR+**.)

Watch Out for Fingerchecks

Define a 'fingercheck' as one of those times when you're doing something totally innocuous, something without risk, and something you've done many times before – and then just as you hit the Enter key, you wish you hadn't.

Surprisingly, this is the single most frequent cause of unplanned catalog outages. The stories are legion and varied. All are about situations that simply should not have occurred, and none of them will likely ever occur to you, right?. Instead, when you least expect it, the worst thing you can imagine will occur – and you'd better be able to either correct it, or cover your tracks very well.

The best advice we can give is: don't ever lose your vigilance and attention to detail when you're doing anything related to an ICF catalog. It's a complex structure, you probably know less than you should about the environment, most likely you haven't practiced much with the tools at hand, you're focused on something else, and, worst of all, it's doubtful a backup of the catalog was taken immediately before you hit the Enter key.

This is the time when you'll wish you had followed the recommendations described here for increased ICF catalog reliability.

CR+ Is Vital for ICF Catalog Reliability

CR+ is the most widely used and up-to-date ICF catalog tool available. Its thorough support of ICF catalogs is evidenced by the recent IBM z/OS 1.7 Software Announcement (dated July 27, 2005): "IBM recommends Mainstar Catalog RecoveryPlus (5620-FGY)", which declares that **CR+** is "best of class". IBM remarkets **CR+** and many other Mainstar products.

Reliability is a cornerstone of **CR+** design. **CR+** includes 27 commands to monitor, tune, repair, and recover every aspect of your ICF catalog environment. Specific to reliability are the following commands:

- **BACKUP BCS/VVDS** performs high speed backups of a BCS or VVDS, ensuring that all of your ICF catalog structures are backed up to a single backup file. The methodology used in BACKUP design focuses on ultra-high integrity in the backup process, guaranteeing a correct backup. For very large catalog environments, a PARALLEL feature utilizes multitasking logic, backing up multiple catalogs at a time, speeding the backup process by 2 to 2.5 times. Aliases assigned to each BCS are automatically backed up from the system's master catalog.
- **DIAGNOSE** has four subcommands, enabling a tailored diagnostic process. For all DIAGNOSE commands, fixes are created to correct all identified error conditions. Frequent scheduling of the DIAGNOSE process is the single best way to improve ICF catalog reliability, enabling you to keep your environment clean and error-free.
  - **DIAGNOSE ALIAS** helps you maintain reliability of your 'alias structure', diagnosing within a single master catalog, or between two master catalogs. In today's z/OS systems, user catalog 'order of search' logic *only* utilizes the alias matching technique, whereby the high-level qualifier of the data set to locate matches the alias of the user catalog the data set is cataloged in.
  - **DIAGNOSE BCS-VVDS** and **VOL-BCS** are two commands for a bi-directional check of record consistency throughout the ICF catalog environment. Extensive multi-volume support – very important in today's z/OS systems – identifies many types of error situations that are virtually impossible to identify and correct.
  - **DIAGNOSE VVDS-VTOC** is a specialized diagnostic process designed to ensure the integrity and reliability of DASD extent information between the VVDS and VTOC. If extent information becomes out-of-synch, serious DASD volume errors may occur on critical volumes.
- **MERGECAT** is the key to an alias balancing project, enabling you to move or copy records from one catalog to another, either by entire alias level, or individual entry. The **CR+** MERGECAT command executes 17 times faster than IDCAMS REPRO MERGECAT, with significantly greater reliability through a single-step process that is greatly simplified over any other facility. Catalog safety is paramount, so MERGECAT provides a SIMULATE, RESTART, and BACKOUT feature in the event errors are encountered during execution.
- **REORG** allows you to dynamically reorganize an open BCS, thereby cleaning up

performance bottlenecks in the catalog, and, more importantly, repairing identified errors. Extensive keywords allow you to alter virtually any attribute of an existing catalog; eliminating the need to take the catalog out of use – and therefore, eliminating disruptions to applications using the catalog.

Conclusion

Your ICF catalog environment is one of the most critical structures within your z/OS system, enabling you to access the many, many data sets that are essential to your installation. Reliability, availability, and visibility (RAV) of your ICF catalog is of paramount importance. If you're willing to spend a bit of time and resources to maintain a clean and workable ICF catalog environment, you will significantly improve the RAV factor for your systems.

Ronald K. Ferguson - Founder, President & CEO of Mainstar Software Corporation

Ron Ferguson has a technical background in large-scale z/OS systems. As a software instructor for 20+ years, he has presented over 600 courses on VSAM and ICF catalogs, and is recognized worldwide as an expert in these areas. Ron travels widely, meeting with customers and presenting at national and international conferences.