

A VSAM/ICF Catalog White Paper

From Mainstar Software Corporation

Catalogs Do Break! — Part 3 – A Real-Life Story

By Ronald K. Ferguson, with technical assistance from Marty Hasegawa

Preface: In this real-life story, we'll examine a situation that recently occurred at the site of a new **Catalog RecoveryPlus** user - one who was just beginning to use the many features of this ICF catalog management tool. Several months earlier, they benefited from using the **Catalog RecoveryPlus** BACKUP/RECOVER commands to recover a damaged user catalog. The user was trying to build an alternate master catalog for protection in the event of a future failure with their master catalog. During this process it was discovered that it too was damaged. The BACKUP/RECOVER command logic in **Catalog RecoveryPlus** proved to be the only way this master catalog could successfully be repaired.

It's impossible to know what caused the failure in this master catalog (or even when it occurred), but it further supports our belief that ICF catalogs continue to break in unexpected ways, not always catastrophically, but with results that will hamper any number of required day-to-day functions that you wish to perform. In this White Paper, we'll detail what led up to the detection of the problem and how this master catalog was recovered. This specific situation may ever happen to your master catalog, but it illustrates the many and varied ways that ICF catalogs develop problems - problems that you need to identify and correct as early as possible.

Errors While Attempting To Create An Alternate Master Catalog

The origin of the incident described in the Preface was rather innocent. The SYSPROGs were performing a simulated **Catalog RecoveryPlus** MERGECAT COPYONLY command, in preparation for running the actual MERGECAT COPYONLY job, that would create an alternate master catalog. The command they were using is shown in *Figure 1*.

```
MERGECAT INBCS (SYSA.MASTER.CAT) -  
          OUTBCS (SYSA.ALTMAST.CAT) -  
          JOURNAL (RONF.MERGECAT.JOURNAL) -  
          COPYONLY -  
          SIMULATE
```

Figure 1: MERGECAT COPYONLY SIMULATE for alternate master catalog

For this discussion, the relevant messages from MERGECAT are shown in *Figure 2*. The user was understandably puzzled by these messages.

```
CAT12009I PROCESS STARTED FOR IXGLOGR.AX1C1.AX1C1.DFHLOG.A0000001 (SIMULATED)  
CAT12057W CATALOG ENTRY NOT FOUND IN SOURCE CATALOG:  
IXGLOGR.AX1C1.AX1C1.DFHLOG.A0000001  
CAT12009I PROCESS STARTED FOR IXGLOGR.AX1C1.AX1C1.DFHLOG.A0000001.DATA (SIMULATED)  
CAT12057W CATALOG ENTRY NOT FOUND IN SOURCE CATALOG:  
IXGLOGR.AX1C1.AX1C1.DFHLOG.A0000001.DATA
```

Figure 2: MERGECAT COPYONLY error messages

The user was no less puzzled when he read the message description for CAT12057W (see *Figure 3*), which seemed to imply that the dataset was present at one moment, but gone the next. He contacted Mainstar Technical Support for a further explanation.

```
CAT12057W CATALOG ENTRY NOT FOUND IN SOURCE CATALOG: xxxxxxxx
DESCRIPTION: Catalog entry xxxxxxxx was found in the source catalog during initial
processing, but was not found during the move phase. Processing continues.
The most likely reason is that a dataset was deleted in the interim. A second
possibility is that the source catalog has a corrupted index that does not point to an
entry that exists in the catalog's data component.
ACTION: If this was a SIMULATE run, then rerun the MERGECAT SIMULATE with
EXAMINE(DATATEST). Otherwise, use IDCAMS to perform an EXAMINE on the source catalog.
```

Figure 3: Message Description

From the information the customer was able to give Mainstar Technical Support, it seemed that the only answer was that someone was deleting the IXCLOGR datasets at the same time MERGECAT was running. The user responded that no activity was occurring on the system at that time, and therefore, it was extremely unlikely anyone was deleting these datasets. That's when things got interesting.

Tracking Down The Error

An IDCAMS DIAGNOSE ICFCATALOG was run against this master catalog, and it completed with return code zero. There were 135 (or so) IXCLOGR datasets identified from the DIAGNOSE, none showing any errors, and the specific cluster and data component name listed in the CAT12057W message from MERGECAT was shown. An IDCAMS LISTCAT on the cluster and data component also did not show them to be in the catalog. Finally, an IDCAMS EXAMINE command was run (see *Figure 4*), and things started to make sense.

```
EXAMINE NAME(SYSA.MASTER.CAT) DATATEST
IDC01700I INDEXTEST BEGINS
IDC01724I INDEXTEST COMPLETE - NO ERRORS DETECTED
IDC01701I DATATEST BEGINS
IDC11728I DATA FOUND IN EMPTY CI
IDC01713I DATA CONTROL INTERVAL DISPLAY AT RBA 129 FOLLOWS
000000 00CD0055 C3006900 2DC9E7C7 D3D6C7D9 4BC1E7F1 C3F14BC1 E7F1C3F1 4BC4C6C8
-- continuation of line above -- *....C....IXGLOGR.AX1C1.AX1C1.DFH*
000020 D3D6C74B C1F0F0F0 F0F0F0F1 40404040 40404040 40000021 26000008 E2E3C1D5
-- continuation of line above -- *LOG.A0000001 .....STAN*
000040 C4C1D9C4 0007D3E2 F1F0D4C5 C70008E2 E3C1D5C4 C1D9C400 1401FFFF FFFFFFFF
-- continuation of line above -- *DARD..LS10MEG..STANDARD.....*
000060 FFFF0801 124F0000 0C010000 30C40062 0029C9E7 C7D3D6C7 D94BC1E7 F1C3F14B
-- continuation of line above -- *.....|.....D....IXGLOGR.AX1C1.*
000080 C1E7F1C3 F14BC4C6 C8D3D6C7 4BC1F0F0 F0F0F0F0 F14BC4C1 E3C10000 1401FFFF
-- continuation of line above -- *AX1C1.DFHLOG.A0000001.DATA.....*
0000A0 FFFFFFFF FFFF0801 124F0000 0C010000 1E0400E6 D6D9D2F4 F6301020 0F820000
-- continuation of line above -- *.....|.....WORK46.....*
0000C0 00200000 00000000 00000000 00006100 34E30000 C42DC9E7 C7D3D6C7 D94BC1E7
-- continuation of line above -- *...../..T..D.IXCLOGR.AX*
0000E0 F1C3F14B C1E7F1C3 F14BC4C6 C8D3D6C7 4BC1F0F0 F0F0F0F0 F14BC4C1 E3C14040
-- continuation of line above -- *1C1.AX1C1.DFHLOG.A0000001.DATA *
000100 40400000 2B030001 0024C9E7 C7D3D6C7 D94BC1E7 F1C3F14B C1E7F1C3 F14BC4C6
-- continuation of line above -- * .....IXGLOGR.AX1C1.AX1C1.DF*
000120 C8D3D6C7 4BC1F0F0 F0F0F0F0 F1000000 00000000 00000000 00000000
-- continuation of line above -- *HLOG.A0000001.....*
Lines 000140 through 000FE0 Contain All Zeroes
IDC01714I ERROR LOCATED AT OFFSET 00000FFC
IDC01727I REFER TO INDEX CONTROL INTERVAL DISPLAY FOR RBA/CI 1
IDC01714I ERROR LOCATED AT OFFSET 0000006B
IDC21702I MINOR ERRORS FOUND BY DATATEST
IDC01708I 251 CONTROL INTERVALS ENCOUNTERED
IDC01710I DATA COMPONENT CONTAINS 5052 RECORDS
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 4
```

Figure 4: EXAMINE DATATEST on Master Catalog

Identifying The Error

As you can see from the EXAMINE INDEXTEST portion of the output, no errors were detected in the index component - in other words, all index records were found to be pointing correctly, (horizontally and vertically), to their expected counterparts. There are no apparent 'gaps' in the index information, and there doesn't appear to be any missing index records.

Yet, when the EXAMINE DATATEST was run, message IDC11728I indicated that a data component CI was found that seemingly contained valid data. If you interpret the first dataset name (in *Figure 4*) within the dumped CI that follows the message, sure enough, it's -

```
IXGLOGR.AX1C1.AX1C1.DFH LOG.A0000001
```

- the same data component name found to be in error during MERGECAT processing, causing it to terminate.

So, what is the problem? Well, it would be impossible to categorically state the cause, but our supposition is that, at the time one of the datasets whose name is in this CI was being created, some event occurred that prevented the catalog index from being updated. It would be nearly impossible to identify what this event was, but to mention just a few possibilities, it could have been due to:

- an ENQ or RESERVE that didn't properly occur.
- a system crash.
- a hardware failure.
- a catalog address space abend.

Whatever it was, it apparently went undetected at the time. We were told by the user that these IXGLOGR datasets were part of a parallel sysplex test that had occurred a year earlier, so presumably, that was the time when the problem occurred.

The master catalog has apparently been functioning well enough since that time, and is now only causing the problem when the MERGECAT is required for creation of an alternate master catalog, preventing that MERGECAT from executing. Effectively, this is a master catalog that runs perfectly well on a day-to-day basis, but nevertheless, it's broken. Worse still, the broken aspect of it prevents this user from creating the alternate master catalog that would be needed in the event a further failure occurs.

Suggesting A Fix

Because of the design logic of the **Catalog RecoveryPlus** BACKUP and RECOVER command process, the specific type of index error that this catalog suffers from, normally can be corrected by simply backing up and restoring. This is because our BACKUP command bypasses the catalog's index, accessing the data component of the catalog as if it was an ESDS, and backs up every CI that it encounters. When RECOVER is run, the old, damaged catalog will be deleted, a new one defined, the records on the backup file sorted, and as they are loaded into the new catalog, a brand new index is constructed from scratch.

Important note: Do not attempt this type of recovery with an IDCAMS EXPORT/IMPORT, as the records in the empty CI will be 'dropped' during the backup. There isn't any information about the empty CI within the index records, and therefore, that CI will never be referenced.

But, don't forget that the master catalog isn't just any 'normal' catalog - you can run BACKUP on your master catalog any time you wish, but you cannot RECOVER it when it is in use as a master catalog. To recover a master catalog from backup, you can only run the recovery job from a system that was IPL'd using a different catalog than this as its master catalog, and the corrupted master catalog to be recovered must be connected to it as a user catalog. This recovery process can be performed from another LPAR (that isn't sharing the corrupted master catalog), or a rescue system that you've created just for this type of occurrence, or an alternate master catalog on the same system as the corrupted master catalog.

Also, don't forget that when the error on this master catalog was first detected, the user was trying to create an alternate master catalog, to be used on this system in the event the master catalog became corrupted or inoperable. Well, at this time, that alternate master catalog hasn't yet been created.

After talking it over with the user, we advised that there were two ways to go about the repair:

Method #1: Create an alternate master catalog directly from the damaged primary master catalog (using **Catalog RecoveryPlus** MERGECAT COPYONLY), then IPL with the alternate master catalog, in order to repair the primary master catalog. The reason this technique works in this particular instance, is that MERGECAT COPYONLY will ignore the records identified in the CAT12057W message we encountered up in

Figure 1, and it was determined that none of the records in the 'unaddressed CI' were critical to the IPL and short-term execution, since we were only going to be doing a repair.

Method #2: Take down the system(s) running the damaged master catalog, then connect it as a user catalog to the master catalog on another system, and repair it from the other system.

Keep in mind, that when planning for either of these methods, an IPL of the system(s) with the damaged master catalog is unavoidable, so you're likely going to have to schedule this with great care.

Assuming method #2 is chosen, here's what you'll have to do:

1. Import Connect the damaged master catalog to the master catalog of the system from which the repair is to be done (if it isn't already).
2. Shut down the system(s) that are using the damaged master catalog.
3. Using **Catalog RecoveryPlus**, run a BACKUP BCS on the damaged master catalog from the repair system. As mentioned above, an IDCAMS EXPORT will not work, as it will not 'see' the records in the damaged CI, and you'll lose them. If you're nervous, you might also want to get a volume-level dump of the volume on which the master catalog resides.
4. Run an IDCAMS DELETE RECOVERY on the damaged master catalog. This physically deletes the catalog, without 'touching' any of the datasets that are cataloged in it. It is imperative that you code RECOVERY, and not FORCE.
5. Run an IDCAMS DEFINE USERCATALOG, to physically allocate a new version of the master catalog.
6. Run a **Catalog RecoveryPlus** RECOVER BCS, using the backup file created in step 3, to restore the full set of records into the new master catalog. You should specify NOALIAS, NOVVDUPDATE, and INTOEMPTY on this command.

Note: The RECOVER command could re-allocate the master catalog for you, and if so, you don't have to run steps 4 and 5 (and you also don't specify INTOEMPTY on RECOVER), but SMS can get in the way and place the catalog on the wrong volume. It is generally considered a bad idea to create master catalogs on SMS managed volumes, and with this technique, you can ensure that it doesn't happen.

7. The NOALIAS parameter tells RECOVER not to change any of the existing alias entries in the current master catalog that the command is running under. This isn't a big point, as the catalog that we're restoring is actually a master catalog, defined temporarily to the current system as a user catalog, and there shouldn't be any aliases defined to it anyway.
8. NOVVDUPDATE is specified so that RECOVER doesn't update any of the catalog name back-pointers. They should already be correct, and there's no reason to do that.

Conclusion

This latest true story should stress just how true this maxim is: **Catalogs do break!** Too many people still think otherwise, but we continue to find more and more ways that catalogs are breaking - and with symptoms that might seem baffling. Once again, the broken catalog was running without noticeable problems, but when some out-of-the-ordinary operation needed to be done on it, the failure unexpectedly showed up.

As this White Paper has once again illustrated, if you don't have **Catalog RecoveryPlus** for that recovery, your chances for succeeding aren't as high as you would want them to be.

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

Ron Ferguson has a technical background in large-scale OS/390 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. Ferguson travels widely, meeting with customers and presenting at national and international conferences.

Marty Hasegawa – Software Developer for Mainstar Software Corporation.

Marty Hasegawa had been a Systems Programmer for large-scale MVS systems for 23 years prior to joining Mainstar in 2000. His highly regarded expertise includes a wide range and depth of technical knowledge in operating system components, applications platforms, and information security. Marty enjoys the challenge of making technology work for people.

Mainstar® is a registered trademark of Mainstar Software Corporation. Catalog RecoveryPlus™ is a trademark of Mainstar Software Corporation.