

DFSMS – What's New with the Catalog Search Interface

Stephen Branch
IBM

March 6, 2015
17103



#SHAREorg



SHARE is an independent volunteer-run information technology association that provides **education, professional networking and industry influence.**



Introduction

- Recognized need to provide catalog information retrieval
 - CAMLST was insufficient
 - IDCAMS invocation by program is complex
 - Invoke from high-level programs
- Need for faster interface than one entry at a time
- Need for Filtering Capabilities
- Resulted in the creation of the Catalog Search Interface
 - Integrated into the base product DFSMS 1.4

Catalog Search Interface

- Provides a callable interface (GUPI) to retrieve selected catalog attributes.
- Supports generic filtering and returns data for one or entries at a time.
- Resume feature for large queries
- Documented in Managing Catalogs, Chapter 11 Catalog Search Interface User's Guide

How is CSI Invoked?

- The CSI consists of a single load module
 - Located in SYS1.LINKLIB (IGGCSI00)
 - May be LOAded and called
 - May be LINKed to
 - May be linked in user load module
 - Be aware of maintenance implications
- May be entered:
 - 24 or 31-bit mode
 - Any protection key
 - Supervisor/Problem program state

Invocation Restrictions

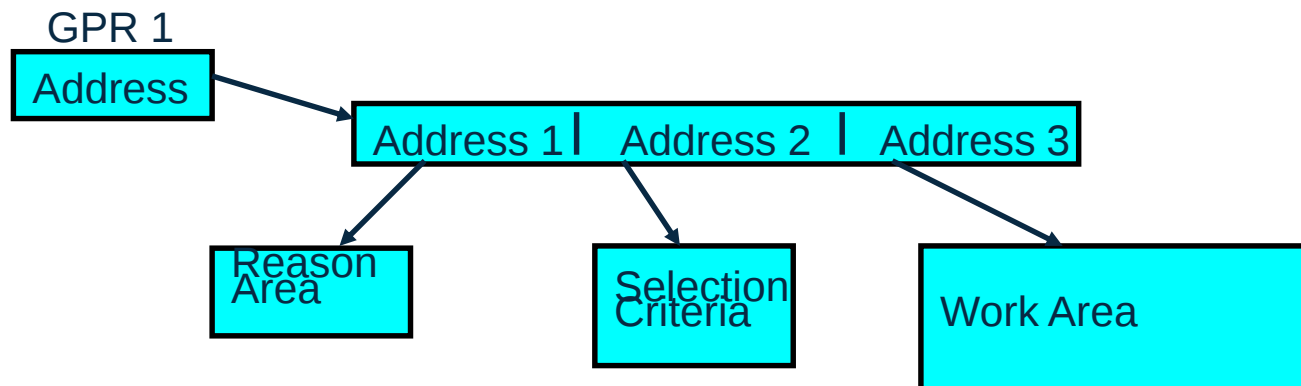
- May not be used in certain environments
 - In a dynamic allocation exit
 - During an OPEN/CLOSE/EOV exit
 - In cross-memory mode
 - In SRB mode
 - In disabled mode

Register Usage

- Register usage on entry:
 - Standard Linkage conventions
 - Three word parameter list pointed to by R1
- On exit:
 - R15 contains a return code indicating success/failure of the request
 - Represents either Catalog or CSI return code

Input Parameter List

- Pointed to by General Purpose R1
- Consists of three addresses:
 - Address of location to put 4-byte return info
 - Used when R15 = 4 or 8
 - Address of structure containing selection criteria
 - Address of work area that will contain returned results



Reason Area

- 4 byte field filled in on return from CSI if R15 is 4 or 8
 - Contains 2 byte Catalog module code, 1 byte reason code and 1 byte return code
 - Reason and return codes are documented under message IDC3009I except return codes 100 and 122.
 - 100 Catalog has detected an error
 - reason 4 – at least one data set entry in error
 - Reason 8 – at least one catalog error
 - 122 invalid filter key

Selection Criteria

- Consists of fields describing:
 - Options that control CSI processing
 - Filtering criteria
 - Types of catalog entries to be returned
 - Generic (or specific) filter key for entry names to be processed
 - Catalog field names to be returned

Selection Criteria Fields

Offset	Type	Length	Field Name	Description
0(0)	VL			CSIFIELD CSI selection criteria fields
0(0)	Character	44	CSIFILTK	Generic filter key
44(2C)	Character	44	CSICATNM	Catalog name or blanks
88(58)	Character	44	CSIRESNM	Resume name or blanks
132(84)	Character	16	CSIDTYPD	Entry types
132(84)	Character	1	CSIDTYP S	Entry types to be returned. All types = blanks
148(94)	Character	4	CSIOPTS	CSI Options
148(94)	Character	1	CSICLDI	Return data or index, Y or blank
149(95)	Character	1	CSIRESUM	Resume, Y or blank
150(96)	Character	1	CSIS1CAT	Search 1 catalog only, Y or blank
151(97)	Character	1	CSIOPTNS	An F entry means to use fullword lengths
152(98)	Fixed	2	CSINUMEN	Number of entries in table
154(9A)	Character	VL	CSIENTS	Variable length table containing field names
154(9A)	Character	8	CSIFLDNM	Field name (1st one)

CSI Options

- CSIDTYPs - Up to 16 entry types to be processed
 - ex: A = non-VSAM, C = Cluster, X = Alias
 - If left blank, all possible types are eligible to be processed, except L, W, and Z
 - Results may include data/index entries, but they cannot be requested directly in CSIDTYPs
 - Specified as flag in selection criteria
 - May be specified in any order
 - Does not affect order of entries returned

CSI Options (*continued*)

- CSICLDI
 - Indicates whether data/index component info will be returned if cluster name matches filter key - "Y" or "N"
- CSIOPTNS
 - Determines length of length fields in returned workarea - either 2 (blank) or 4-bytes ("F")
 - Strongly recommended you always specify "F" (4-byte lengths)

CSI Options (*continued*)

- CSIRESUM - Must be blank to start.
 - Set to "Y" if more entries remain but workarea is full
 - Set to blank when no more entries remain
- CSIRESNM - name where resumed request will continue (Do not change!)
- CSIS1CAT - Indicates whether only one ("Y") or all eligible catalogs (blank) should be searched

Field Names

- Are indicated by a halfword count of field names, followed by the 8-byte field names themselves
 - Field names and their format are listed in "Managing Catalogs", Appendix C
 - Examples: ENTYPE – entry type - Char
VOLSER – volume serial - Char
VSAMTYPE – KSDS, ESDS, etc.
Bitstring
HARBA - Fixed
 - Currently 75+ catalog field names,
 - 9 Volcat Library Entry field names,
 - 15 Volcat Volume entry field names
 - 3 Catalog Control Block field names (new in V2R2)

Filter Key

- Determines which catalog entries will be processed
 - Special characters expand search
 - "*" indicates one or more characters in a qualifier will match
 - "**" - zero or more qualifiers will match.
 - Must be preceded by period or blank
 - Must be followed by a period or blank
 - "%" - one alphanumeric/national character matches
 - "%%.." - 1 to 8 alphanumeric/national match

Filter Key Examples

- TEST.**
 - All data sets beginning with 'TEST'
- TEST.*
 - All two-level data sets beginning with 'TEST'
- TEST.*.EXEC
 - All three-level data sets with TEST as first qualifier and EXEC as last
- TEST.*.*.LISTING
- TEST.ABC%.*.TEMP

Limiting the Catalogs Searched

- Unless otherwise specified, all catalogs that could match the filter key will be searched
 - SYS*. ** will search all catalogs that have an alias of SYS and up to 5 characters in the high-level qualifier
 - When a generic is in the first level, a list of catalogs is built
 - Matching entries in all catalogs in the list are returned
 - ** will search ALL catalogs and ALL data set names will match
 - PDF 3.4 (ISPF) verifies you *really* want to do this

Limiting the Catalogs Searched (*continued*)

- You can limit the search by filling in CSICATNM and possibly CSIS1CAT
 - If CSIS1CAT is set and CSICATNM is non-blank, ONLY that catalog is searched
 - If CSIS1CAT is not set and CSICATNM is non-blank, that catalog and the master will be searched
 - If CSICATNM is blank, any catalog with an alias that matches the high-level qualifier of the filter key, and the master, will be searched

Duplicate Alias Reference

- A specific catalog will only be searched once, regardless of how many times an alias is found for it that matches the filter key
- Example
 - Filter key = 'ABC*.*'
 - Alias ABCD of Catalog TEMP.CATALOG
 - Alias ABCE of Catalog TEMP.CATALOG
 - TEMP.CATALOG is only searched once

MLA and Catalog Searching

- MLA level affects catalogs that may match
 - MLA level = 2
 - Filter key = 'ABC*.**'
 - Alias ABCD of Catalog TEMP.CATALOG
 - Alias ABCD.X of Catalog TEMPX.CATALOG
 - Alias ABCE.Y of Catalog TEMPY.CATALOG
- You may get duplicate data set names

Duplicate Returned Entries

- What appear to be duplicate or unwanted names may be returned
 - The filter key is matched against the master catalog aliases
 - Usercats pointed to by those aliases may be searched
 - Catalogs searched use the filter key for entries returned

Duplicate Returned Entries (*continued*)

- Example
 - Filter key of A*. **
 - Alias AB for Catalog CATALOG.X
 - Alias AC for Catalog CATALOG.Y
- CATALOG.X is NOT searched only for entries starting with AB
- CATALOG.X is NOT searched only for entries starting with AC
- Any entries found in either catalog starting with “A” will be returned

Work Area Return Information

- The information retrieved is returned in the work area pointed to by the third parameter
- Size of passed workarea input in first word
- CSI will return size of work area filled with data (third word)
- Minimum size is 1K (1024), maximum is (currently) 8M (8,388,600)
 - We recommend at least 512K area, depending on the results you expect

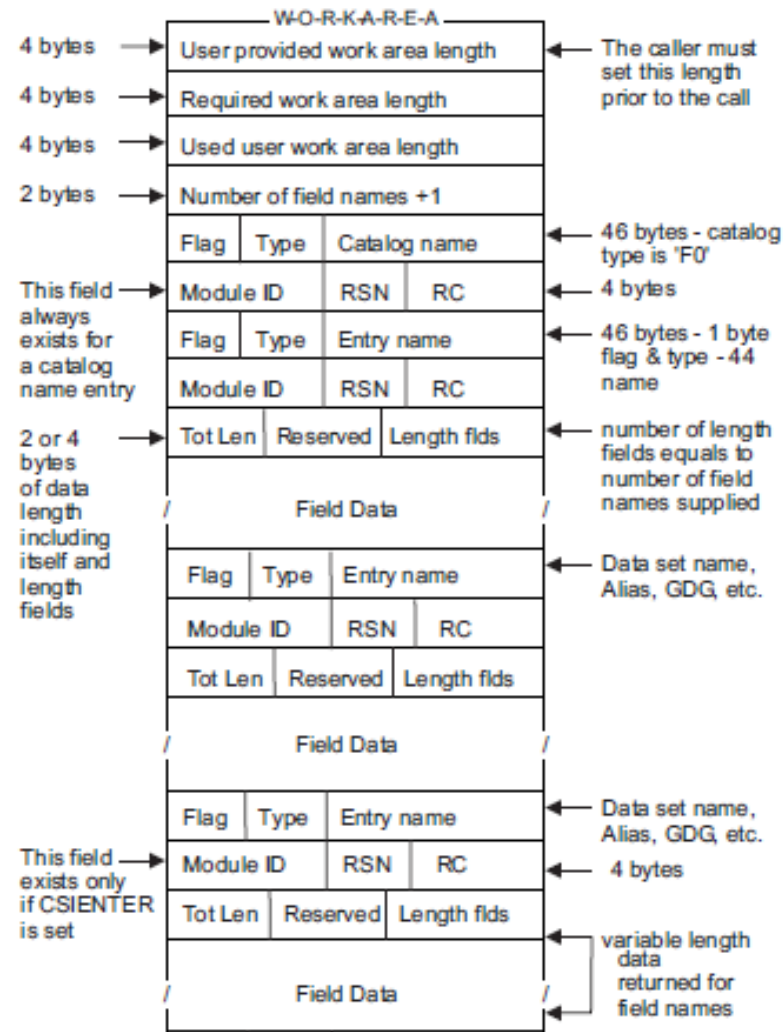
Work Area Return Information (*continued*)

- Entries are returned by catalog searched
 - There is an entry for a catalog that was searched followed by all of the entries found in that catalog
 - This repeats for every catalog that was searched
 - There may be no entries found in a catalog
 - Each catalog entry has a return/reason code indicating success of the search
 - Each individual entry may have a return/reason code as well

Work Area Return Information (*continued*)

- After each individual entry is
 - a total length of the data returned for this entry
 - length and data for each field specified in the field name list
- You must use the total length field to determine when you have reached the end of the data for an entry.
- The next entry (catalog or data set) will follow

General Format of a CSI Workarea



Format of a CSI Workarea Catalog Entry

Flag	Type	Catalog name	
Modid		Rsn	Rc

Flag - indicates all, partial, or none of catalog processed

Type - X'F0' indicates a Catalog entry

Modid - two character Catalog module issuing Rc/Rsn

Rsn - one byte reason code, documented in IDC3009I

Rc - one byte return code, documented in IDC3009I

Format of a Data Entry

Flag	Type	Entry name	
Modid		Rsn	Rc

Flag - X'80' - primary entry (e.g. Cluster),
X'40' - Error detected, info in MODID/RSN/RC
X'20' - Data is returned for this entry

Type - one-byte indicating type of entry

Following only exist if X'40' is on in Flag:

Modid - two character Catalog module issuing Rc/Rsn

Rsn - one byte reason code, documented in IDC3009I

Rc - one byte return code, documented in IDC3009I

Returned Data for an Entry

Total Len	Reserved
Length of field 1 data	
Length of field 2 data	
Length of field 3 data	
Etc...	
Data for field 1	
Data for field 2	
Data for field 3	

Returned Data (*continued*)

- Rules for values of data determined by length field:
 - -1 - security data suppressed by RACHECK
 - 0 - no data found, normally not applicable to entry type
 - 2/4 - for variable length, field does not exist, value of data will be zero
 - n - fixed length field of length 'n'.
 - If data contains high-values (e.g. X'FFFF'), field does not exist

Example of a Normal Return

```

00000400 00000000 000003E4 000600F0 8E *.....U...0*
E2E8E2F1 4BD4E5E2 D9C5E24B D4C1E2E3 *SYS1.MVSRES.MAST*
C3C1E340 40404040 40404040 40404040 *CAT*
40404040 40404040 40404040 00000000 *.....*
A0C30000 00000000 00000000 00000000 *.C.....*
00000000 00000000 00000000 00000000 *.....*
00000000 00000000 00000000 00000000 *.....*
00260000 00000000 00000000 00000000 *.....*
00020000 00040000 00040000 FFFFFFFF *.....*
FFFFFFFF 20C4E2E8 E2F14BD4 E5E2D9C5 *.....DSYS1.MVSRE*
E24BD4C1 E2E3C3C1 E3404040 40404040 *S.MASTCAT*
40404040 40404040 40404040 40404040 *.....*
40400000 00300000 00000000 00060000 *.....*
00040000 00020000 00040000 0004E4E2 *.....US*
D9D7C1D2 3010200F 0000000A F000000A *RPAK.....0...*
F00020C9 E2E8E2F1 4BD4E5E2 D9C5E24B *0..ISYS1.MVSRES.*
D4C1E2E3 C3C1E34B C3C1E3C9 D5C4C5E7 *MASTCAT.CATINDEX*
40404040 40404040 40404040 40404040 *.....*
0000003A 00000000 0000000C 00000008 *.....*
00000002 00000004 00000004 E4E2D9D7 *.....USR*
C1D2E4E2 D9D7C1D2 3010200F 3010200F *AKUSRPAK.....*
00000000 04000000 0400A0C3 D4C1E2E3 *.....CMAS*

```

Decoding an Entry

```
00000400 00000000 000003E6 000600F0 8E *.....0*
E2E8E2F1 4BD4E5E2 D9C5E24B D4C1E2E3 *SYS1.MVSRES.MAST*
C3C1E340 40404040 40404040 40404040 *CAT *
```

X'0400' - length of work area caller provided

X'03E6' - length of work area filled by CSI

X'0006' - # of fields + 1. Fields requested were:

VOLSER

DEVTYP

STORCLAS

HURBADs

HARBADS

Decoding an Entry (*continued*)

```

00000400 00000000 000003E6 000600F0 8E *.....0*
E2E8E2F1 4BD4E5E2 D9C5E24B D4C1E2E3 *SYS1.MVSRES.MAST*
C3C1E340 40404040 40404040 40404040 *CAT*
40404040 40404040 40404040 00000000 *.....*

```

X'**00**' - Flags indicate no errors, all applicable data has been returned for this catalog

X'**F0**' - this entry is for a catalog

X'**00000000**' - No error information has been set for this catalog

Decoding an Entry (*continued*)

A0C30000	00000000	00000000	00000000	*.C.....*
00000000	00000000	00000000	00000000	*.....*
00000000	00000000	00000000	00000000	*.....*
00260000	00000000	00000000	00000000	*.....*
00020000	00040000	00040000	FFFFFFFF	*.....*

FFFFFFFF

X'A0' - Primary entry, data is returned

X'C3' - ('C') Entry is a Cluster entry

X'00000026' - Total length of data

X'00000000' - Reserved

X'00000000' - Length of 1st field (VOLSER) doesn't apply

X'00000000' - Length of 2nd field (device type) doesn't apply

X'00000002' - Length of 3rd field (storage class)

X'00000004' - Length of 4th field (HURBADS)

X'00000004' - Length of 5th Field (HARBADS)

X'0000' - STORCLAS value (2-byte len of zero = no STORCLAS)

X'FFFFFFFF' - HURBADS - high-values = not available/applicable

X'FFFFFFFF' - HARBADS - high-values = not available/applicable

Decoding an Entry (*continued*)

```

      20C4E2E8 E2F14BD4 E5E2D9C5
E24BD4C1 E2E3C3C1 E3404040 40404040
40404040 40404040 40404040 40404040
40400000 00300000 00000000 00060000
00040000 00020000 00040000 0004E4E2
D9D7C1D2 3010200F 0000000A F000000A
F000

```

```

*.....DSYS1.MVSRE*
*S.MASTCAT          *
*                   *
* .....*
* .....US*
*RPAK.....0...*

```

X'20' - Not primary entry, data returned

X'C4' - ('D') Entry is data component

X'00000030' - length of returned data

X'00000000' - reserved

X'00000006' - Len of 1st field (VOLSER)

X'00000004' - Len of 2nd field (DEVTYP)

X'00000002' - Len of 3rd field (STORCLAS)

X'00000004' - Len of 4th field (HURBADs)

X'00000004' - Len of 5th field (HARBADS)

Decoding an Entry (continued)

20C4E2E8 E2F14BD4 E5E2D9C5	*.....DSYS1.MVSRE*
E24BD4C1 E2E3C3C1 E3404040 40404040	*S.MASTCAT *
40404040 40404040 40404040 40404040	* *
40400000 00300000 00000000 00060000	* *
00040000 00020000 00040000 0004E4E2	*US*
D9D7C1D2 3010200F 0000000A F000000A	*RPAK.....0...*
F000	

X'E4E2D9D7C1D2' = C'USRPAK' - Value of VOLSER

X'3010200F' - Value of DEVTYP

X'0000' - Value of STORCLAS

X'000AF000' - Value of HURBADs

X'000AF000' - Value of HARBADs

Entries and Their Fields

- Field data is specific
 - Applies to a specific level in the data set
 - What type of entry would show HARBA?
 - Cluster?
 - Data?
 - Index?
 - Non-VSAM?
- Best way to know what entry (C, D or I) would contain the field is to look at a LISTCAT output of the entry

Coding Hint

Catalog Entry 'F0'x

Entry (A, C, D, I, etc.

Field data

Entry (A, C, D, I, etc.

Field data

Entry (A, C, D, I, etc.

Field data

Catalog Entry 'F0'x

Entry (A, C, D, I, etc.

Field data

Possible resume point

Think of processing the work area with 3 loops. The outer most loop is for catalogs, then the next loop is for entries, then innermost loop is for field data. When field data processing is finished, move to the next entry, if the entry is not 'F0'x, process the field data for that entry. If it is 'F0'x, process next catalog. Do until out of work area. Check for possible resume point.

How to get started

- Sample programs in SYS1.SAMPLIB
 - IGGCSILC - BAL program does something similar to LISTCAT NAME
 - IGGCSIVG - BAL program to identify unused space at end of VSAM data set
 - IGGCSIVS - BAL program listing data sets on a volume and their associated catalog
 - IGGCSIRX - REXX EXEC returns data set name(s), type, and volsers for a prompted filter key

CATSRCHI – Even better still!!!

- Improved REXX exec based on IGGCSIRX – offered “as is”
- CATSRCHI – Allows filter key input for name and any field names
- Available for download:
 - <ftp://public.dhe.ibm.com/servers/storage/support/software/dfsms/cattools/>
 - II14316.CATSRCHI.TR
 - II14316.CATSRCHI.JCL.TR
- Two files:
 - CATSRCHI REXX exec
 - JCL to run exec in batch

CATSRCHI – FTP Download JCL

```
//FTPGET    JOB ,
//          MSGCLASS=H,MSGLEVEL=(1,1),TIME=(1440),
//          USER=&SYSUID,NOTIFY=&SYSUID,CLASS=B
//*****
//*
//*  FTP COPY OF CATSRCHI FROM FTP SERVER
//*
//*****
//DUMMY     EXEC PGM=IEFBR14
//DD1       DD   DSN=YOURID.II14316.CATSRCHI.TR,UNIT=SYSDA,
//  DISP=(,CATLG,CATLG),SPACE=(CYL,(700,300),RLSE),
//  DCB=(RECFM=FB,LRECL=1024,BLKSIZE=6144,DSORG=PS)
//DD2       DD   DSN=YOURID.II14316.CATSRCHI.JCL.TR,UNIT=SYSDA,
//  DISP=(,CATLG,CATLG),SPACE=(CYL,(700,300),RLSE),
//  DCB=(RECFM=FB,LRECL=1024,BLKSIZE=6144,DSORG=PS)
//FTPbatch  EXEC PGM=FTP,REGION=4M
//SYSPRINT  DD   SYSOUT=*
//OUTPUT    DD   SYSOUT=*
//INPUT     DD   *
public.dhe.ibm.com
anonymous
branchs@us.ibm.com
binary
cd servers/storage/support/software/dfsms/catttools/
get II14316.CATSRCHI.TR +
'YOURID.II14316.CATSRCHI.TR' (replace
get II14316.CATSRCHI.JCL.TR +
'YOURID.II14316.CATSRCHI.JCL.TR' (replace
dir
quit
/*
//
```

Note: Case Counts!

CATSRCHI – Unterse JCL

```
//UNTERSE JOB ,  
//      MSGCLASS=H,MSGLEVEL=(1,1),TIME=(1440),  
//      USER=&SYSUID,NOTIFY=&SYSUID,CLASS=B  
//STEP1 EXEC PGM=AMATERSE,PARM=UNPACK  
//SYSPRINT DD SYSOUT=*  
//SYSUT1 DD DISP=SHR,  
//      DSN=YOURID.II14316.CATSRCHI.TR  
//SYSUT2 DD SPACE=(CYL,(5,10),RLSE),  
//      VOL=(,,,1),  
//      DISP=(NEW,CATLG,DELETE),UNIT=SYSDA,  
//      DSN=YOURID.II14316.CATSRCHI  
//STEP2 EXEC PGM=AMATERSE,PARM=UNPACK  
//SYSPRINT DD SYSOUT=*  
//SYSUT1 DD DISP=SHR,  
//      DSN=YOURID.II14316.CATSRCHI.JCL.TR  
//SYSUT2 DD SPACE=(CYL,(5,10),RLSE),  
//      VOL=(,,,1),  
//      DISP=(NEW,CATLG,DELETE),UNIT=SYSDA,  
//      DSN=YOURID.II14316.CATSRCHI.JCL
```

CATSRCHI – Execution JCL

Batch JCL:

```
//CSIRXJCL JOB , 'Your Name ', CLASS=A,
//          MSGCLASS=H, NOTIFY=&SYSUID
// *
//STEP1     EXEC PGM=IKJEFT1B
//SYSPROC   DD DISP=SHR, DSN=yourlib.CLIST.CLIST
// Point to your CLIST data set.
//SYSPRINT   DD SYSOUT=*
//SYSTSPRT  DD SYSOUT=*
//SYSTSIN   DD *
//          %CATSRCHI filter key FIELDNAMES() TYPES(ABC...) -
//          CAT(catalog name)
// *
//
```

CATSRCHI – Parameters

%CATSRCHI filter key TYPES(ABC...) FIELDNAMES() CAT(catalog name)

- CATSRCHI is the member name found in yourlib.CLIST.CLIST
- All parameters are optional except the filter key, ex: A.B%%.C*. **
- TYPES()
 - Can be specified with no spaces, spaces or commas
 - TYPES(ABC) TYPES(G P) (X,H)
 - If not specified, you get ABCGHRUX by default
 - Types Z, L, and W must be specified to get these selected
- FIELDNAMES()
 - If not specified, VOLSER ATTR1 XHARBADS XACIFLAG EXTENT XHURBA are the defaults
 - List of field names in Managing Catalogs, Chapter 11
- CAT()
 - Name of a catalog to be searched
 - If specified, limits the filter search to that catalog only
 - If not specified, filter key is used to select catalog(s)

CATSRCHI – Even better still!!!

- Example using SMS.KSD%

```
%CATSRCHI SMS.KSD%
```

```
Catalog Search Interface is being called with filter key: SMS.KSD%  
and catalog field name(s): VOLSER ATTR1 XHARBADS XACIFLAG EXTENT XHURBA
```

```
Catalog  SMSCAT
```

```
CLUSTER  SMS.KSDS
```

```
ATTR1    = x'FF'
```

```
XHARBADS = no data
```

```
XACIFLAG = no data
```

```
EXTENT   = no data
```

```
XHURBA   = no data
```

```
DATA      SMS.KSDS.DATA
```

```
On volser(s) 338001
```

```
ATTR1      = x'40'
```

```
XHARBADS   = 122880
```

```
XACIFLAG   - Data set is not EA (extended addressing)
```

CATSRCHI – Even better still!!!

- Example using SMS.GDG%.** TYPES(B H) FIELDNAMES(GDGLIMTE VOLSER)

```
%CATSRCHI SMS.GDG%.** TYPES(B,H) FIELDNAMES(GDGLIMTE VOLSER)
```

```
Catalog Search Interface is being called with filter key: SMS.GDG%.**  
and catalog field name(s): GDGLIMTE VOLSER  
Requesting catalog entries: BH
```

```
Catalog  SMSCAT  
GDG      SMS.GDGE  
GDGLIMTE = x'03E7'
```

```
GDS      SMS.GDGE.G0001V00  
GDGLIMTE = no data
```

```
On volser(s) 1P0302
```

```
GDS      SMS.GDGE.G0002V00  
GDGLIMTE = no data
```

```
On volser(s) 1P0301
```

CATSRCHI – Even better still!!!

- Example using SMSCAT TYPES(Z) FIELDNAMES(CATFLAGS)

```
%CATSRCHI SMSCAT TYPES(Z) FIELDNAMES(CATFLAGS)
```

```
Catalog Search Interface is being called with filter key: SMSCAT
and catalog field name(s): CATFLAGS
Requesting catalog entries: Z
```

```
Catalog SYS1.MVSRES9.MASTCAT
```

```
CAXWA SMSCAT
```

```
CATFLAGS = x'A000'
```

```
BIT STRING = b'101000000000000000'
```

```
First Byte explains catalog is ....
```

```
1xxxxxxx open
```

```
x1xxxxxx the master catalog on this system
```

```
xx1xxxxx active in In-Storage Cache
```

```
xxx1xxxx active in VLF
```

```
xxxx1xxx ECS-active
```

```
xxxxx1xx open in RLS mode
```

```
xxxxxx1x deleted
```

```
xxxxxxx1 locked
```

New CSI Fields

- The CSI Fields are documented in Chapter 11 of the *DFSMS Managing Catalogs* manual.
- New fields for CAXWA in private
 - A new entry type of “Z” (used in the CSIDTYP5 field) is used to tell the CSI that the information desired is dynamic data from CAS control blocks rather than information stored in the catalog.
 - No other types are allowed to be specified with the Z entry type
 - The master catalog name is returned for every Z entry request
 - For Z entry requests, the catalog name is specified in the filter key field (CSIFILTK); the catalog name field (CSICATNM) should be blank

New CSI Fields (con't)

- CATACT: A 2-byte field containing the catalog activity count (the number of address spaces that have the catalog allocated)
- CATFLAGS: A 2-byte field containing flags that indicate the following:
 - Catalog is open
 - Catalog is the master catalog on this system
 - Catalog is active in In-Storage Cache
 - Catalog is active in VLF
 - Catalog is active in ECS
 - Catalog is open in RLS mode
 - Catalog is deleted
 - Catalog is locked
- CATUCB: A 4-byte field containing the address of the UCB for the volume the catalog resides on

New CSI Fields (con't)

- Fields for GDG Extended and GDG PURGE
 - GDGLIMTE: A 2-byte fixed number of the number of GDSs in the GDG. Meant to replace GDGLIMIT.
 - GDGATTR: not new but new setting added.

no	Bitstring	1	GDGATTR	Generation data group attributes
	0...			Delete oldest GDS when GATLIMIT exceeded
	1...			Delete all GDSs when GATLIMIT exceeded
	.0..			Do not scratch data set when rolled off
	.1..			Scratch data set when rolled off if volume mounted
	..0.			Allocate GDSs in LIFO order
	..1.			Allocate GDSs in FIFO order
	...0			GDS NOPURGE
	...1....			GDS PURGE
	 0...			GDG Classic
	 1..			GDG Extended

The End

If you have questions in the future, please contact

Stephen Branch

branches@us.ibm.com