

## **Browsing MVS Control Blocks Using DDLIST/ISRDDN Under ISPF.**

I am a big advocate of ISRDDN/DDLIST under ISPF. However I only recently found out about a nice feature which I had never used before after reading the online help. The help states the following:-

*Browse modname {+offset}*

*Browse a load module as it exists in storage. modname may also be an absolute address or an indirect addressing string in TSO TEST syntax. Absolute addresses must be terminated with a period. Recursive brows (module or address) is supported.*

*While you are browsing storage, you can use any of the regular facilities of ISPF Browse. You can also scroll UP past the 'Top of Data' line until you reach the beginning of the readable contiguous storage.*

*When already browsing storage, you may use the following primary commands:*

- *BROWSE modname* Browse a module that is already loaded. If it is not loaded, you can use the *LOAD* command to explicitly load and browse it. You may also specify an address terminated with a period.
- If modname is not specified and the cursor is on an 8 character address, that address is browsed (similar to point and shoot).*
- *CANCEL* Leave all nested browse sessions.
- *REFRESH* Return to the original address that was used to start this browse session. Use the *REFRESH* command after you have scrolled up from the base address (at offset +0) to return to the +0 location.
- *LOAD modname* Explicitly load and browse a module.
- *FORMat* or *DUMP* View storage in dump format.
- *DATA* Browse unformatted storage with offsets.
- *RAW* Browse storage without any formatting.
- *WIDE* or *NARRow* Switch between wide and narrow display formats.

*This value will be saved in application profile variable ZDDVIEWWW as 'W' or 'N' and used as the display format for subsequent similar displays.*

- *DISASM ON* Disassemble the load module being browsed.
- *DISASM OFF* Cancel the *DISASM* function.
- *SETDATA offset* Specify that the data at the specified offset is not an instruction, while browsing a disassembly.

*The Browse command can be enhanced to view control blocks by name by allocating a sequential file to DD name *ISRDDN* and placing in that file a control block name and TSO *TEST* locator string on each line.*

*The first string on the line is a control block name. The second string is a locator string. Everything else on the line is ignored.*

*With the following file, you could type *B ASCB* to view the *ASCB*:*

*CVT 10.? Communications Vector Table*

*ASCB CVT?+C? Address Space Control Block*

This intrigued me so I did some digging. I allocated a sequential dataset with the following attributes on disk.

**MKTA                    Allocate New Data Set**

**Command ==>**

**More: +**

**Data Set Name . . . : MEERKAT.ISRDDN.CB**

**Management class . . .                    (Blank for default management class)**

**Storage class . . . .                    (Blank for default storage class)**

**Volume serial . . . .                    (Blank for system default volume) \*\***

**Device type . . . . .                    (Generic unit or device address) \*\***

**Data class . . . . .                    (Blank for default data class)**

**Space units . . . . . TRACK            (BLKS, TRKS, CYLS, KB, MB, BYTES  
or RECORDS)**

**Average record unit                    (M, K, or U)**

**Primary quantity . . 1** (In above units)  
**Secondary quantity 0** (In above units)  
**Directory blocks . . 0** (Zero for sequential data set) \*  
**Record format . . . . fb**  
**Record length . . . . 80**  
**Block size . . . . . 27920**  
**Data set name type** (LIBRARY, HFS, PDS, LARGE, BASIC, \*  
EXTREQ, EXTPREF or blank)  
**Extended Attributes** (NO, OPT or blank)

Once allocated I scanned the web where I found this file:-



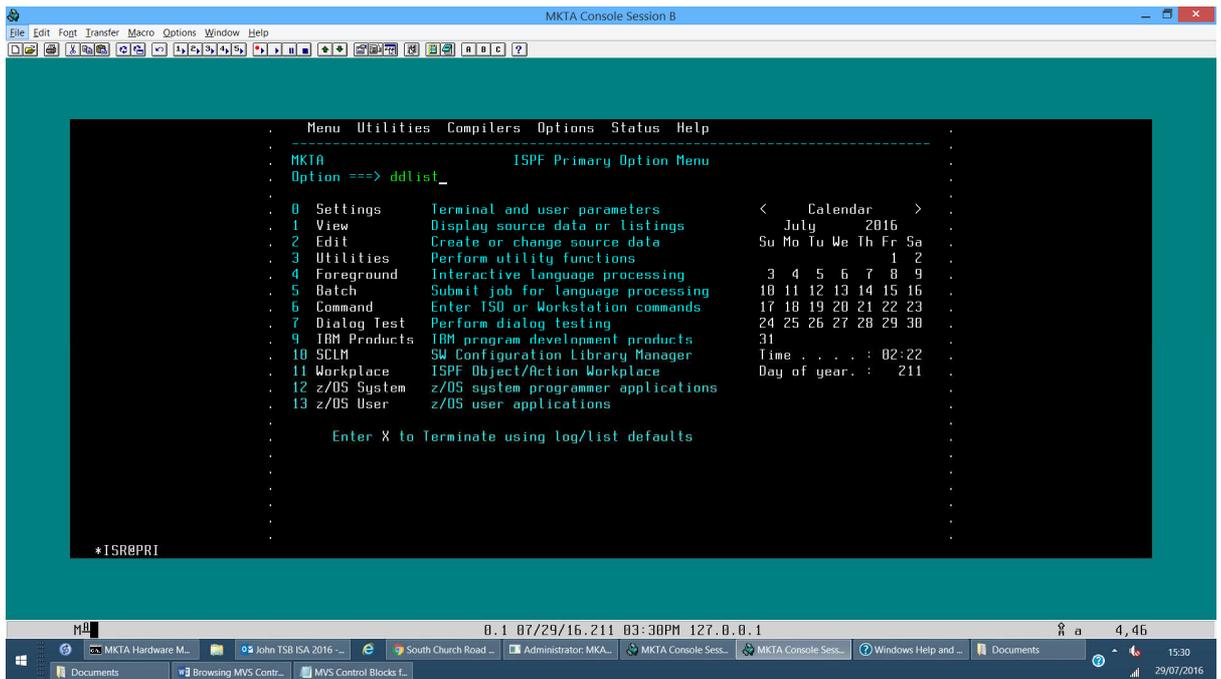
MVS Control Blocks for ISRDDN.txt

It contains the main MVS control blocks. Here is a snippet of the contents:-

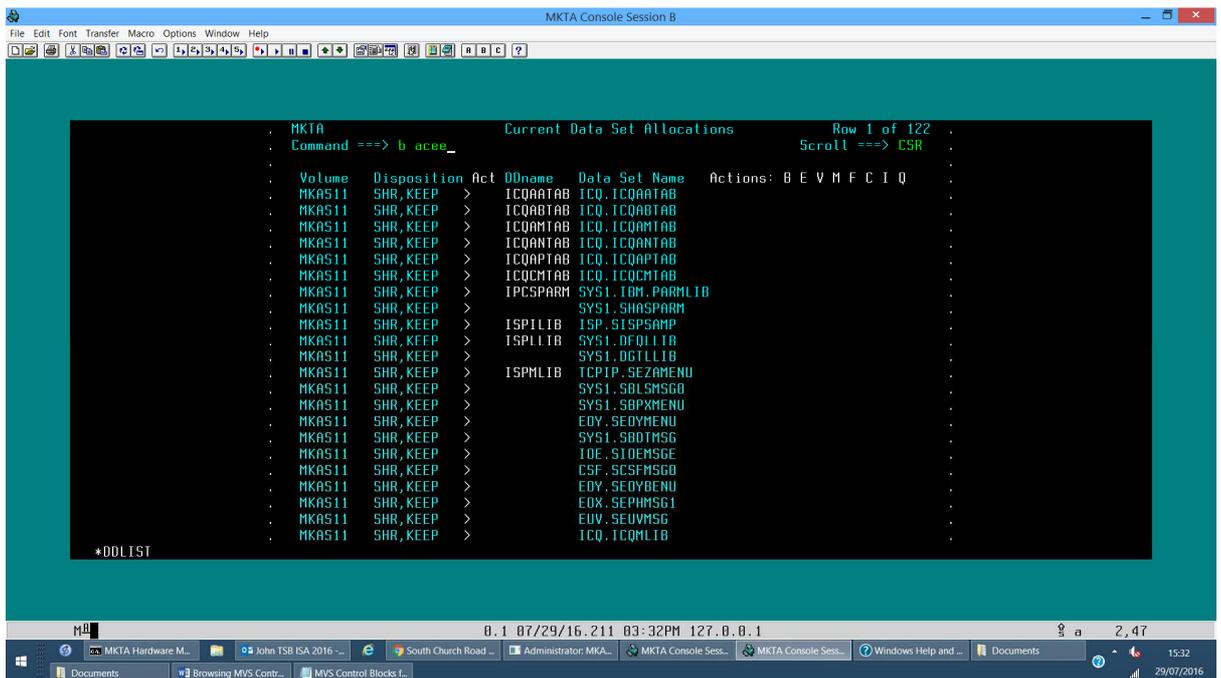
*ACEE ASXB+C8? Accessor Environment Elt (current) LSQA IHAACEE*  
*ACEX ACEE+98? Acee extension (current) LSQA*  
*APHT CSVT+C?+8? APF List*  
*ASCB PSA+224? Address Space Control Block (current) SQA IHAASCB*  
*ASCBJNI ASCB+AC? ASCB (Current) Batch Jobname (or zero)*  
*ASCBJNS ASCB+B0? ASCB (Current) Start/Mount/Logon*

Once in place I decided to try browsing the said control blocks via DDLIST. The following is a step by step rundown of how to do this:-

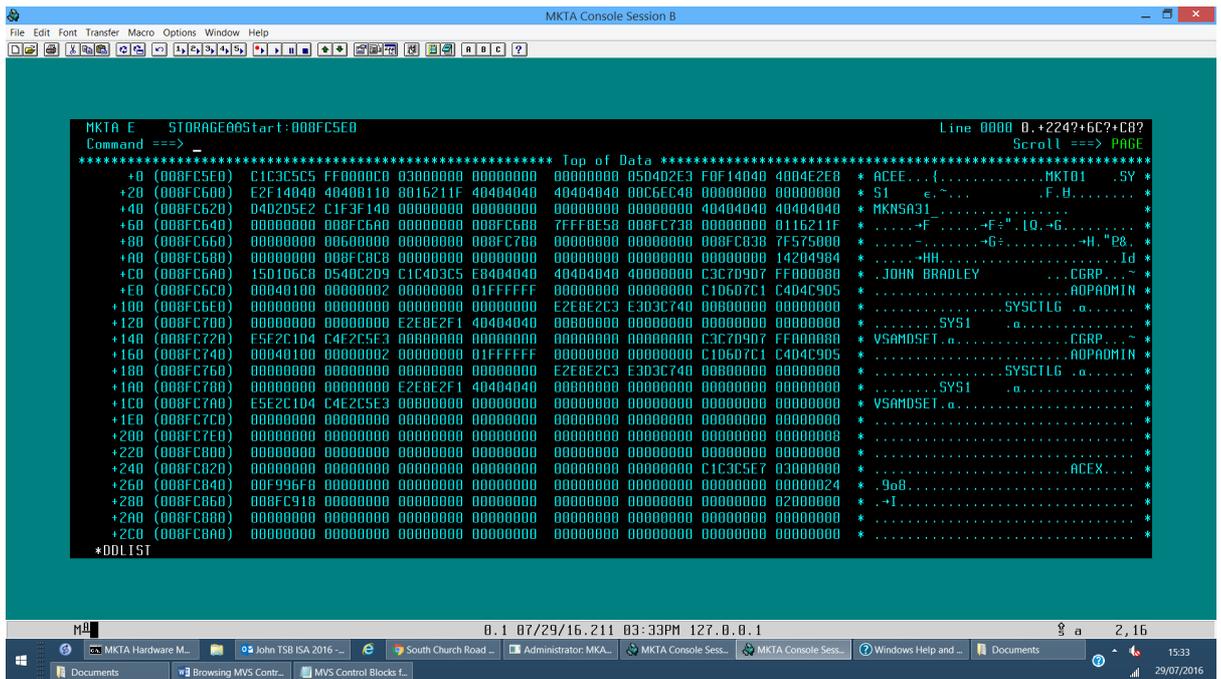
1. In ISPF go to option 6 and issue the following command:-  
**ALLOC F(ISRDDN) DA('MEERKAT.ISRDDN.CB') SHR**
2. Then issue DDLIST to get into the ISRDDN utility:-



3. Then as an example lets browse the ACEE associated with the logged on user:-



4. You will get a display of the actual control block:-



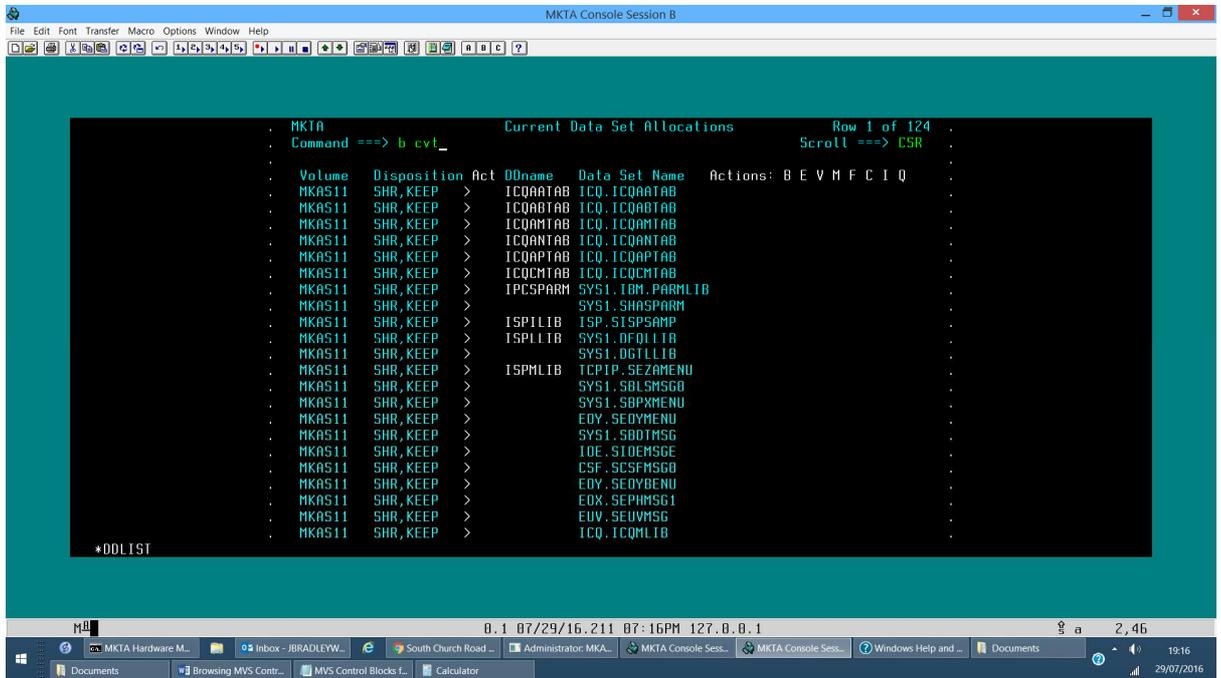
- Next if you want to browse another control block from here just enter “B controlblockname” on the command line and so on. To go back to the previous display just press PF3.

This facility is useful for debugging and to see what switches and features are set within your session and can be much easier than writing REXX execs to retrieve data about the system. It is possible to chain to other control blocks by positioning the cursor on addresses returned as part of the initial display.

So as an example here is how to find the Master Catalog name for the system you are logged onto via control blocks in storage.

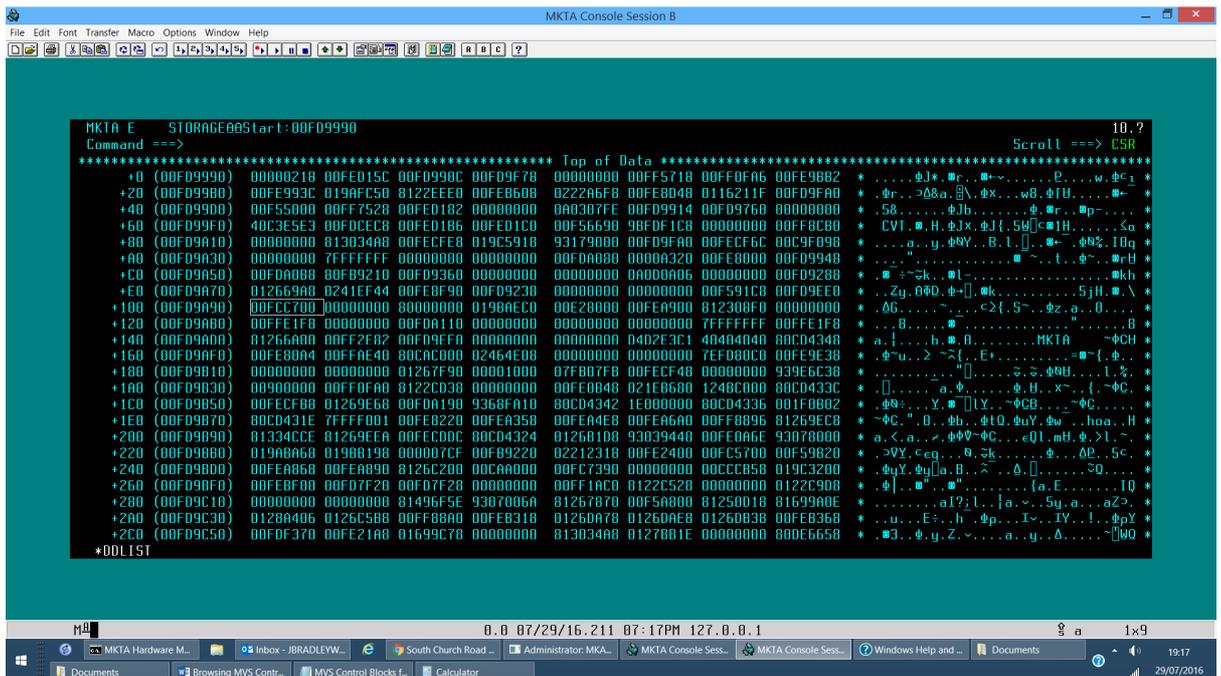
The chain of control blocks to get to the Master Catalog name in storage is as follows:-

- a) X'10' point to CVT



- b) X'100' point to AMCBS control block

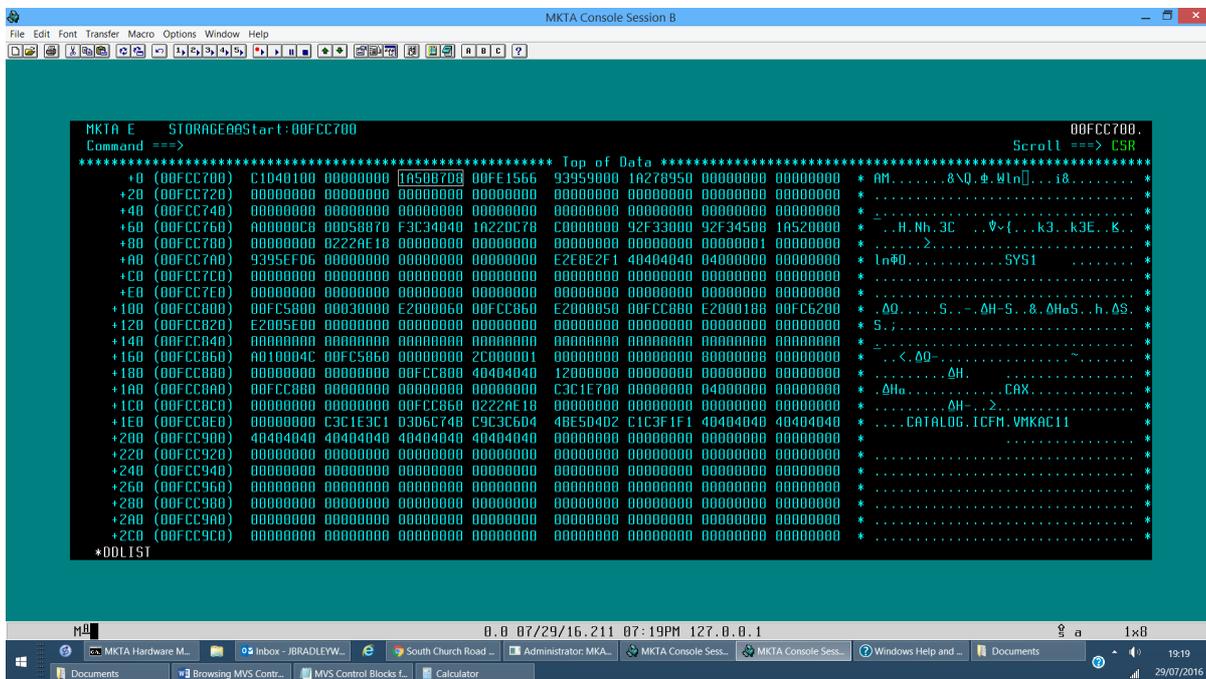
On the resulting display you place the cursor at X'100'.



Hit ENTER to be taken to that address.

c) X'08' point to Master Catalog ACB

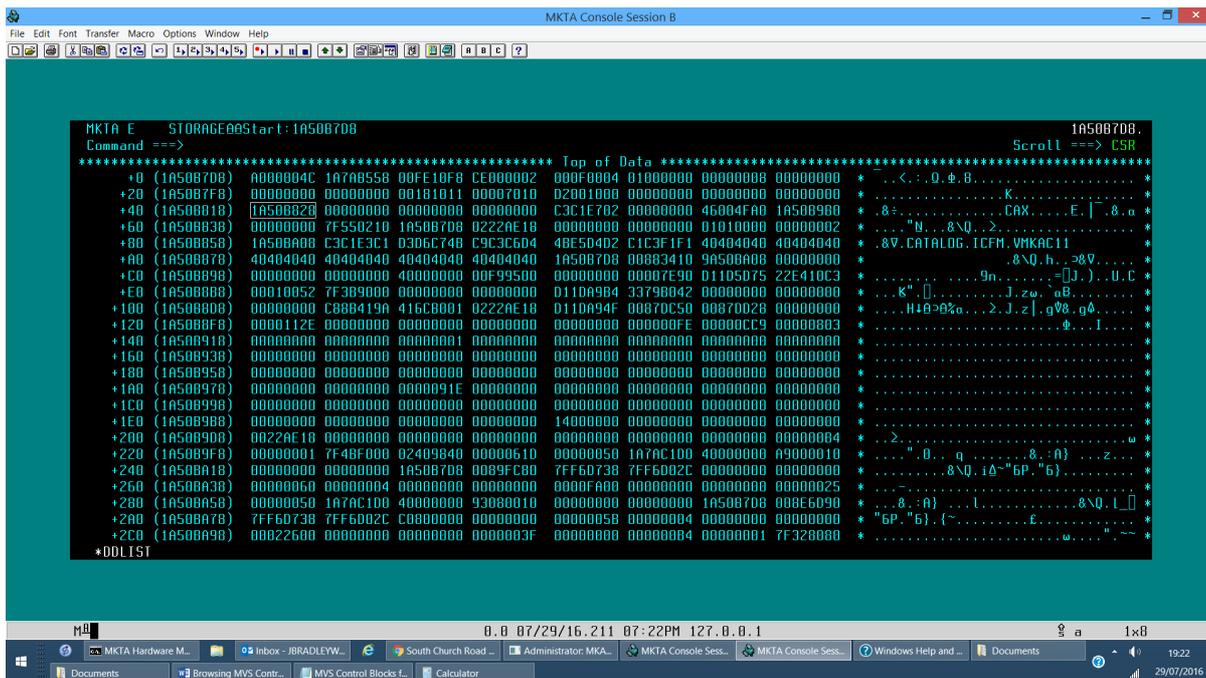
Next place the cursor at byte 8 in the display.



Again hit ENTER.

d) X'40' point to CAXWA control block

Place the cursor at X'40' and hit ENTER.



e) X'34' to start of Master Catalog name

Finally go to offset at X'34' and the next 44 bytes are the Master Catalog name.

```
MKTA E STORAGE00start:1A50B820 1A50B820.
Command ==> Scroll ==> CSR
***** Top of Data *****
+0 (1A50B820) C3C1E702 00000000 46004FA0 1A50B900 00000000 7F550210 1A50B708 0222AE18 * CAX.....E.[]&....."N...&\Q..>.*
+20 (1A50B840) 00000000 00000000 01010000 00000002 1A50BA08 C3C1E3C1 0306C748 C9C3C604 * .....&V.CATALOG.ICFM.*
+40 (1A50B860) 4BE50402 C1C3F1F1 40404040 40404040 40404040 40404040 40404040 * .VMKAC11.*
+60 (1A50B880) 1A50B708 00883410 9A50BA08 00000000 00000000 00000000 40000000 00F99500 * .&\Q.h...>V.....9n.*
+80 (1A50B8A0) 00000000 00007E90 01105075 22E410C3 00010852 7F3B9000 00000000 00000000 * .....=[]J)...U.C...K"[].....>.*
+A0 (1A50B8C0) 011DA94F 0087DC50 0087DD28 00000000 00000000 C88B419A 416C8001 0222AE18 * J.n.n...b.....H10>0%a...>.*
+E0 (1A50B8E0) 00000000 000000FE 000000C9 00000003 00000000 00000000 00000001 00000000 * J.z[]g%8.g0.....>.*
+100 (1A50B920) 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 * .....&.....>.*
+120 (1A50B940) 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 * .....&.....>.*
+140 (1A50B960) 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 * .....&.....>.*
+160 (1A50B980) 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 * .....&.....>.*
+180 (1A50B9A0) 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 * .....&.....>.*
+1A0 (1A50B9C0) 14000000 00000000 00000000 00000000 0022AE18 00000000 00000000 00000000 * .....&.....>.*
+1C0 (1A50B9E0) 00000000 00000000 00000000 00000000 00000001 7F4BF000 02409840 00000610 * .....&.....>.*
+1E0 (1A50BA00) 00000050 1A7AC1D0 40000000 A9000010 00000000 00000000 1A50B708 0089FC80 * ..&.:A}...z.....&\Q.i0~.*
+200 (1A50BA20) 7FF60738 7FF6082C 00000000 00000000 00000050 00000004 00000000 00000000 * "6P."6}.....&\Q..>.*
+220 (1A50BA40) 0000FA00 00000000 00000000 00000025 00000050 1A7AC1D0 40000000 93080010 * .....&.....>.*
+240 (1A50BA60) 00000000 00000000 1A50B708 008F6D90 7FF60738 7FF6082C C0000000 00000000 * .....&\Q..[]>6P."6}..>.*
+260 (1A50BA80) 00000056 00000004 00000000 00000000 00022600 00000000 00000000 0000003F * .....&.....>.*
+280 (1A50BA00) 00000000 000000B4 00000001 7F328000 02409158 00002838 00000000 000000B4 * .....&.....>.*
+2A0 (1A50BAC0) 00000001 7F31F000 02409000 00002830 00000000 000000B4 00000001 7F320040 * .....&.....>.*
+2C0 (1A50BAE0) 02408FC0 0000283F 00000000 000000B4 00000001 7F321000 02408E80 00002841 * ..+{.....&.....>.*
*DDLIST
```

In this case for our MKTA system the name is CATALOG.ICFM.VMKAC11.

This facility can be useful if you need to check various variables on your system or to check what a user's address space is doing and the type of authority they have when logged on.