**Mainframes Notes**

**SYSOUT** is always allocated and gets among other things all the output from the System level process (including any messages about the JCL itself, performance stats, error messages etc.)

**SYSPRINT** is just another DD which, by convention, is used by utility programs for thier output.

JCL related messages like allocation/deallocation messages, performance stats, error messages etc will be displayed JESMSG of JCL spool

The JES (Job Entry Subsystem) is a separate piece of software from the job initiator (which is what writes to SYSOUT) and its outputs are directed to JESMSG

**Directing System Output**   
([Next](http://www.simotime.com/jclone01.htm#Dddummt)) ([Previous](http://www.simotime.com/jclone01.htm#DDsetConcat)) ([Table-of-Contents](http://www.simotime.com/jclone01.htm#TOC          ))

The DD statement with a SYSOUT parameter is used to assign an output class to an output data set. The form of this parameter is:

//REPORT1 DD SYSOUT=A

//REPORT2 DD SYSOUT=\*

In the preceding example the REPORT1 output will be directed to SYSOUT=A which is usually the system printer. The REPORT2 output will be directed to the location specified in the MSGCLASS of the JOB statement. If MSGCLASS=0 then REPORT2 output will be directed to SYSOUT=0 which is usually placed in the JES Output Queue for a specified period of time *(such as 3 days)* before being purged.

if the ddname is SYSOUT (i.e. //SYSOUT ...) and SYSOUT=\* then the job class will be the same as that specified in the MSGCLASS in the job statement. Since the ddname is SYSOUT the DISPLAY in a COBOL program will be directed to the location specified in the MSGCLASS of the JOB statement. If MSGCLASS=0 then REPORT2 output will be directed to SYSOUT=0 which is usually placed in the JES Output Queue for a specified period of time *(such as 3 days)* before being purged. The following is a sample JCL statement with SYSOUT as the DD name.

//SYSOUT DD SYSOUT=\*

The following is a sample COBOL statement that uses DISPLAY.

DISPLAY 'This is a test...'

In the preceding example the text string "This is a test..." will be routed to the JES output queue.

**EXTENTS**

(SPACE=TRK,(100,100))

It means that the 100 tracks specified will be allocated in one extent if sufficient contiguous space exists, if not, the 100 tracks can be allocated in up to 5 seperate extents.

Well the size of track depends on the Device Type   
For eg.   
  
1.) For the DEVICE TYPE = 3390 number of bytes in one TRACK is 56664 BYTES.   
  
2.) For the DEVICE TYPE = 3390 number of TRACKS in one CYLINDER is 15 TRACKS.   
  
3.) BLOCKS are INDEPENDENT of the device type but it depends on the input files DCB parameter.   
  
The same goes for the 2ndary space, but the max extents allowed for a single **vol** DS is 16. So using expats example of a 5 extent primary, you are limited to 11 2ndary extents max. If you need more than 1 extent for any 2ndary the max is reduced further.   
  
BTW, if you alloc across multi vols you get 16 additional 2ndary extents on each vol after the 1st. Same rules apply.

for a non-vsam dataset it is 16 and for vsam it is 123  
what does exactly extents mean? how it will allocated the memory?  
can anyone tell me in detail plz?

When a file is allocated under z/OS, the system goes out and finds a disk pack that matches the specifications you provided. The system then looks for free space large enough to allocate the file. If it finds enough free space, the file is allocated on the disk drive and takes up one extent. If there is not enough free space to allocate the file in one extent, the system can use up to 5 free space areas that total enough space to allocate the file -- so it would be allocated in up to 5 extents (and yes these 5 count against the 16 total limit). If you start writing to the file and it runs out of space, the system will then attempt to allocate a secondary extent using the parameters you provided when you defined the file.  
  
So basically an extent has absolutely nothing to do with memory; an extent is an area on a disk pack that contains all or part of a file. Mainframes work very differently than PCs do -- for a PC the next extent pointer is part of the actual file, whereas the mainframe maintains a table of where the extents are located on the disk and therefore the table can hold 16 entries (123 for VSAM) -- once you fill it up, your file cannot allocate any more space on that disk no matter how much free space there is on the disk.

**Exception:** This routine produces the report on files which are crossing the safe limit. If issue is with space, more volumes can be allocated dynamically to these files to avoid space problems in the batch. If the issue is with number of extents, files need to delete / defined.

**How can we know issue is with space or extents?**

**What is data class of files?**

If issue is with volumes, check the data class of files, If the data class is set to allocate

the volumes automatically on need basis, no action is required. Else, L1 team needs

to add the volumes dynamically

**1) Can any one explain about INTRDR.**

Eg: sysout(\*,intrdr)

"An internal   
reader (INTRDR) is a special SYSOUT data set that other programs can use to submit   
jobs, control statements, and commands to JES2."

**2) What is NDM**

NDM is used within the financial services industry, government agencies and other large organizations that have multiple mainframes, Mid-Range, [Linux](http://en.wikipedia.org/wiki/Linux) or Windows systems

Network Data Mover (NDM), it was renamed to Connect:Direct in 1993, following the acquisition of Systems Center, Inc

NDM's primary advantage over [FTP](http://en.wikipedia.org/wiki/FTP) was that it made file transfers routine and reliable.

**3) What are some of the ABENDs generated, when not enough disk space is available and what do they mean?**

**ABEND SB37** : When the entire disk volume runs out of space, the ABEND **B37** is generated. The system allocated all the primary and secondary space it could, but it still could not write the output.

**ABEND D37 :** Primary disk space was exceeded and no secondary space has been specified, or it has been exhausted. Here, the possible solution is to increase both the primary and the secondary space, being allocated to the dataset.

- **ABEND E37 :** One of the more popular ABENDs. This often occurs when you try to add members to a PDS, and you get E37 ABEND. It occurs because the PDS Directory require more space for its members, than it was initially alloted. A possible solution is to **compress the PDS, to make more room for new members**. This can be done by 2 ways :   
1) In ISPF Menu, press Z against the **PDS** name.   
2) Run IEBCOPY with **INDD** and **OUTDD** pointing to the same PDS.