

AS400 THEORY NOTES

<https://github.com/skill-at/AS-400-Training/wiki> --> From sir

[ashwin417/as400_myNotes](#) → from Ashwin

The Application system/400 was the result of the “**Silver Lake project**” at IBM’ - on **21st June 1988**

ADTS(Application Development Tool Set)

- PDM (Program Development Manager) . Command to start : STRPDM
- SEU (Source Entry Utility) . Command to start STRSEU
- SDA (Screen Design Aid) . Command to Start STRSDA
- RLU (Report Layout Utility) . Command to start STRRLU
- DFU (Data File Utility). Command STRDFU

System architecture

There are three categories of software on the AS/400:

- Application Software - PDM
- Operating System Software - CL
- System Licensed Internal Code (SLIC)

Client Access Software:

- Enable communication between client and server.
- IBM Personal Communication
- RENEX
- MOCHA SOFT (TN5250)
- RUMBA/400

ASP(Auxiliary Storage Pool)

The logical partition on AS/400 is called as Auxiliary Storage Pool (ASP). The ASP is like C drive D drive etc on Personal Computers except that the ASPs on AS/400 are numbered.

Within ASP are what are called as Libraries.

TYPES OF LIBRARIES

For one job max of 25 libs can be defined

- System
- Product
- Current library
- User library

STANDARD RECORD LENGTHS OF SRCPF

QRPGSRC, QDDSSRC, QDDL SRC, QCLSRC, QCMD SRC, etc - record length of 92

QRGLESRC - a record length of 112 (92 default + 20 bytes of comment)

Source physical file is an object. But the source member is not an object. When we compile the member, the object is created for that source.

AS400 LIMITS (refer github)

AS400 jobs (Batch job detailed)

A job is a piece of work that is done on AS400. Job name is composed of **Job Number/User/Job Name**.

Interactive jobs in AS400?

It requires user to be always interactive/signed in till the time job runs. It doesn't wait in line for system resources.

Batch job.

Batch job is a predefined group of processing actions submitted to the system to be performed with little or no interaction between the user and the system. Jobs that do not require user interaction to run can be processed as batch jobs. A batch job typically is a **low priority job** and can require a special system environment in which to run. Batch jobs run in the system background, freeing the user who submitted the job to do other work. Several batch jobs can be active at the same time.

Flow of a Batch Job

- ? **SBMJOB** command is run to start the batch job.
- ? Goes to Job Queue and waits for its turn with status **JOBQ**
- ? Become **ACTIVE** when its turns come/enters subsystem
- ? Goes to Output Queue (**OUTQ**) after getting finished.
- ? Spool file Moves to **Print Writer**
- ? Prints out on **Printer**

Job description:

It is AS400 object of type *JOB

Contains a specific set of job related attributes. Attributes determine how each job is run on the system. ? Values in the Job Description can be overridden during the run of commands SBJOB • Can specify: ? Initial library list ? Job Queue ? Job Priority: of jobs using this job description. ? Output Queue

Job queue :Whenever a batch job is submitted it goes to the place so called Job Queue where it waits for its turn to get processed based on its priority. This priority is called as job priority.

Values for job priority are from 1 to 9 , 1 being highest and 9 being lowest

Output Queue

Output queue contains spool files waiting for its turn to get printed. This is based on output priority 1 being highest and 9 being lowest

Run priority

This is priority which job uses when it is active Specifies the run priority for the job. Run priority is a value, ranging from 1 (highest priority) through 99 (lowest priority), that represents the priority at which the job competes for the processing unit relative to other jobs that are active at the same time.

Subsystem :

The subsystem is the work place for jobs on your system. All user work is done by jobs running in the subsystem and it is important to monitor this area for slow work performance.

Object types :- *sbsd (subsystem description) command : WRKSBSD SUBSYSTEME NAME *jobq (Job queue) command : WRKJOBQ JOBQUEUE NAME *outq Command : wrkoutq outq name

Concatenating Strings

Character strings can be combined into longer strings with three built-in concatenation operators: *CAT (concatenate), *BCAT (blank and concatenate) and *TCAT (trim and concatenate).

Operation	Description
*CAT or	Joins the two strings exactly the way they are (including any trailing blanks). If the first string has three trailing blanks, the resulting string would have those three blanks embedded in the middle.
*BCAT or >	Joins the two strings in such a way that only one blank is placed between the two strings (even if trailing blanks are found in the first string). *BCAT guarantees that there only will be one intervening blank.
*TCAT or <	Joins the two strings with the first string trimmed (without trailing blanks), TCAT results in a string that has no embedded blanks no matter how many trailing blanks were found in the first string.

All three concatenation operators leave the second string as is. If the string contains any leading blanks, those leading blanks are always included in the resulting string.

Comparison	Description
*EQ or =	Equal to
*NE or ≠	Not equal to
*LT or <	Less than
*LE or ≤	Less than or equal to
*NL or ≮	Not less than
*GT or >	Greater than
*GE or ≥	Greater than or equal to
*NG or ≮	Not greater than

SUBROUTINES IN CL

SUBR (Subroutine) command is used in CL program to begin the subroutine block.

ENDSUBR (End Subroutine) command is used in CL program to end the subroutine block.

CALLSUBR (Call Subroutine) command is used in CL program to execute/call a subroutine.

RTNSUBR command is used to return a value and exit the subroutine when **RTNVAL** is optional to use.

CRTDUPOBJ V/S CPYF

In CRTDUPOBJ for a logical file the created duplicate file will be also logical file and for a physical file the created file will also be a physical file. Even the record format identifier will also be the same. While in case of CPYF, if we are copying a logical file then the created file be a physical file not a logical file.

Commitment control is a function that ensures data integrity. It defines and processes a group of changes to resources, such as database files or tables, as a transaction.

A physical file is composed of three components, namely,
Record Format
Access Path
Data Member

ENTRIES LEVELS IN PHYSICAL FILE

There are four levels of entries that the physical file contains in a specific order as described below:

File Level – UNIQUE, FIFO, LIFO, FCFO, REF, REFFLD

Record Format Level – FORMAT, TEXT

Field Level –

ALIAS, ALWNULL, CMP/COMP, COLHDG, DATFMT, DATESEP, REFFLD, TEXT, TIMFMT, TIMSEP, VALUES, VARLEN

Key Field Level-ABSVAL, DESCEND, DIGIT, SIGNED, UNSIGNED

A **Logical file** (LF) with Keys are Access path (INDEX) over the physical file (PF)

A Logical file (LF) with no key are VIEWS over the physical file (PF).

If there is/are Logical files (LF) present for a physical file (PF) then we cannot delete Physical file (PF) until and unless we delete all the dependent Logical files (LF) over that physical file (PF). But Logical file (LF) can be deleted without deleting the Physical file (PF) first.

TYPES OF LF

Non-Join Logical File

Single Record Format Logical File

File level keyword -- PFILE

Multiple Record Format Logical File

Join Logical File (LF)

File level – JDFTVL(LEFT OUTER JOIN)

Join level – JDUPSEQ, JOIN, JFLD

Record – JFILE

Field – JREF

JOURNALING

Four basic journal entry categories

- The most common journal entries fall into four basic categories (J, F, R, C).
- Within each category there are number of different journal entry types represented by a two-character entry code (e.g. PR, NR for journal entry J).

Journal and journal receiver operations (J). These include such things as references to the previous receiver (PR) or the next receiver (NR) in a chain. Also, at IPL-time, an entry is made (e.g., an IN entry for IPL after normal end) marking a critical chronological boundary in the file activity.

File operations (F). This category includes file opens (OP) and file closes (CL).

Record operations (R). Record updates (UP), deletes (DL), and new records written (PT and PX) all fall into this category.

Commitment control (C). Anything related to commitment control falls into this category. Some examples are begin commitment control (BC), start a commit cycle (SC), commit operation (CM) and rollback operation (RB).

To get journal entries use DSPJRN command , either take output to OUTFILE or display results.

Example : - DSPJRN JRN(JRNFILE) FILE((EMPPF)) RCVRNG(*CURCHAIN) OUTPUT(*OUTFILE)
OUTFILFMT(*TYPE3) OUTFILE(PRAF12121/JRNOUTPUT)

RPGLE built in functions

- 1) %SUBST
- 2) %SCAN
- 3) %CHAR
- 4) %XLATE
- 5) %LEN
- 6) %TRIM,%TRIML,%TRIMR
- 7) %CHECK
- 8) %CHECKR
- 9) %DEC
- 10) %DATE
- 11) %DAYS
- 12) %MONTHS
- 13) %YEARS
- 14) %TIMESTAMP

SQL → RCDFMT → RENAME

To allow lowercase values, go to edit change source settings "Change session defaults" or F13. Then change uppercase input only to N.

SQL - CREATE OR REPLACE TABLE(not null with default, primary key), CREATE VIEW...AS..[CAST,ROUND], CURRENT DATE TIME, INDEX(create unique index indexname on tablename(key1,key2)), DROP(equivalent to dltf or wrkobj filename 4), Alter table(equivalent in dds -- chgpf), flatfiles(files without str), DELETE FROM(equivalent to clrpfm clear phy member

COMMAND LIST

1. CRTLIB
2. CHGCURLIB
3. DSPLIBL
4. DSPLIB
5. DLTLIB
6. EDTLIBL
7. ADDLIBL
8. WRKLIBPDM
9. DSPSYSVAL/QSYSLIBL
10. DSPSYSVAL QUSRLIBL
11. RMVLIBLE
12. WRKMBRPDM
13. STRDBG
14. DSPFD
15. DSPFFD
16. CRTSRC PF

17. WRKJOB
18. SBMJOB
19. SBSDB
20. WRKJOBQ
21. WRKACTJOB
22. WRKSBSDB
23. WRKOUTQ
24. WRKACTJOB SBS(SUBSYSTEM Name)
25. WRKUSRJOB
26. WRKSBSJOB
27. WRKSBS
28. WRKSBMJOB
29. DSPMSG
30. DSPMSG QSYSOPR
31. WRKOBJLCK
32. DSPRCDLCK
33. CRTDUPOBJ
34. RCVF – CPF0864 - original program model (OPM) program
35. CLOF
36. SNDRCVF
37. DCLF – RCDLMT
38. CPYF – DROP,MAP,NOCHK REPLACE,ADD CRTFILE(YES)→CURLIB

Copy records where EMPCTY starts with 'H' (First character is 'H') # INCCHAR parameter

CPYF FROMFILE(*LIBL/EMPPF) TOFILE(*LIBL/EMPPFBK) FROMMBR(*FIRST) TOMBR(*FIRST)
MBROPT(*REPLACE) CRTFILE(*NO) FROMRCD(*START) INCCHAR(EMPCTY 1 *EQ 'H')

Field test (INCREL) parameter using relational operator

39. MONMSG
40. STRCMTCTL
41. ENDCMTCTL
42. STRRLU
43. CRTDTAARA
44. RTVDTAARA
45. CHGDTAARA
46. DSPDTAARA
47. DLTDTAARA
48. CRTPF
49. CHGPF
50. CRTLF
51. DSPDBR
52. ADDPFM – MAXMBRS(While CHGPF)
53. OVRDBF
54. STRSQL
55. RUNSQLSTM
56. OPNQRYF
57. DLTOVR
58. SNDMSG
59. SNDUSRMSG

60. SNDPGMMMSG
61. SNDBRKMSG
62. CLRPFM
63. RMVM
64. CPYFRMIMPF
65. CPYTOIMPF
66. CRTJRNRCV
67. CRTJRN
68. STRJRN
69. ENDJRN
70. ADDPFTRG
71. RUNQRY
72. WRKQRY
73. UPDDTA
74. CHGCURDIR
75. CHGOWN
76. MOV
77. RMVLNK
78. WRKAUT
79. WRKLNK
80. ENDJOB
81. ENDSBS
82. HLDJOBQ
83. RLSJOBQ
84. SBMJOB
85. STRSBS
86. CHGJOB
87. DSPJOBLOG
88. WRKSYSSTS
89. CRTSAVF
90. RSTOBJ
91. SAVOBJ
92. SAVLIB
93. CHGAUT
94. EDTOBJAUT
95. GRTOBJAUT
96. RVKOBJAUT
97. CRTOUTQ
98. CRTDEVPRT
99. STRRMTWRTR
100. WRKOUTQ
101. WRKSPLF
102. DLTSPLF
103. WRKWTR
104. CRTUSRPRF
105. WRKUSRPRF
106. CRTBNDDIR
107. WRKBNDDIR

- 108. CRTSRVPGM
- 109. UPDSRVPGM
- 110. STRPGMEXP
- 111. ENDPGMEXP
- 112. DSPSRVPGM
- 113. CRTPRTF
- 114. DSPSYSVAL
- 115. DLTF
- 116. DSPPGMREF
- 117. CRTDSPF
- 118. DSPPGM
- 119. RCLACTGRP
- 120. FNDSTRPDM
- 121. CPYSPLF
- 122. ADDPFCST