D0 NOTE # 2046

DESCRIPTION OF THE D0 DATABASE LINKER PROTOTYPE

by

Homer A. Neal
Department of Physics
University of Michigan
Ann Arbor, Michigan 48109

March 31, 1994

General Goals of Project

This project was designed to provide collaboration members involved in data analysis with easy access to basic information about each D0 data run. It was undertaken at the request of D0 Spokesman Paul Grannis in the context articulated in his D0 News note of February 3, 1993, a portion of which is reproduced below:

> *"Work has been underway to develop useful data bases for luminosity, beam backgrounds, detector status, analysis status, beam x-y location etc.. I would like to promote the work on these subjects toward a common tool to be used to collect global information about the run. Homer Neal has agreed to convene a working group which will include the summary of luminosity, machine parameters, main ring backgrounds etc. (N. Amos), multiple interaction correction (R. Partridge), run-time conditions including status of detectors (H. Prosper, S. Chopra), production analysis data base (P. Bhat, L. Paterno), beam x-y (S. Zinchenko) — and whatever other run or store information is of general use. One of the main tasks of this group will be to coordinate and standardize these many pieces of information characterizing the run and to see that the appropriate standard interrogation tools and documentation are produced."*

Design Philosophy:

The database linker package was developed with the intent of providing the user with a single application entry point from which he or she could access information in all relevant databases, including the run summary database, the production database, various electronic logs, and the accelerator luminosity databases. In its initial form, one visits the various databases through a series of menu choices. Callable routines are available to provide direct access to specified tasks, though those would need to be customized for individual applications.

The term "database linker" was chosen for this application because it can access any of our standard databases, whether they be DBL3, RdB or indexed files. This global reach is made possible by the use of the DEC 4GL application RALLY. With RALLY one can open and use RdB databases directly, or call Fortran packages that access other databases. RALLY contains the forms management facilities, menu and form processing functions that are needed to provide a self-contained environment for collecting the basic parameters for each D0 data run.

Using the RALLY Application Development Language (ADL), a procedure was written which quickly builds a RdB database containing summary information about each run and makes it available for querying by the user. It is straightforward, for example, to obtain a listing of all runs with luminosities greater than a certain value and with a calorimeter run quality factor of "perfect", and with the beam x,y values within a specified range.

In this note, I will give some details of the design of the database linker and illustrate the various functions accessible to the user. Those desirous of additional information are encouraged to contact the author.

Included in this note is also a demonstration of the feasibility of using scanned logbook images to provide distant collaborators with access to any and all "original" experiment documents.

Though all of the goals originally established for this project have been met or exceeded, the product is still to be regarded as a flexible entity capable of being fashioned to be of the maximal utility to the user. For this reason, suggestions for improvement are welcomed. They could provide the basis for a very effective product for the latter parts of Run I and thereafter.

This package was originally released on August 18, 1993 as a D0 News posting, and is reproduced here as Appendix I. Most of the figures included in this note are actual screen captures.

Databases Accessed by the Linker:

The databases and files accessed by the Linker are described below.

Accelerator Luminosity Database:

The accelerator luminosity database has been developed by Norman Amos. It is a indexed file stored on DOLUM$OUT. Details of the parameters, and the corrections applied in determining the luminosity are presented in Appendix II. When one selects the accelerator luminosity database choice from the RALLY main menu, one is taken to the same package that has been routinely made available by Norm Amos to those requesting access to the luminosity data. With this package, one can extract the corrected accelerator luminosity for a run, for a list of runs, or for a range of runs.

Run Summary Database:

The Run Summary Database is a DBL3 database developed by S. Chopra that contains summary information about each D0 run. Fields include the run quality variable for each subdetector, the accelerator and min-bias luminosities, global summary file information, etc. The database is populated by the DBL3 server on ONLINE, D0FS and UNIX clusters. For Run 1A the database is on D0::DBONLINE:[DBL3.RSM]DBRUNSUM.DAT. For RUN IB the name has been changed to DBSUM$RSM.DAT. (Further details of the naming conventions and file locations can be provided upon request). When one accesses the Run Summary Database from the DBLINKER, it is straightforward to also display the global run summary data for the specified run. An example of the contents of the database and the global run summary display is given in the next section.

Production Database:

The Production Database is a RdB database developed and maintained by P. Bhat and colleagues (D0 Note # 1266). It contains detailed information about the file, trigger, luminosity, and processing status for each run. In the DBLINKER application, the Production Database is accessed by a spawned call to a set of utilities developed by Puspha Bhat. As will be seen later, a call to this package can return information about what runs were taken between selected times, their luminosities, triggers and filters for the run, etc. More details are provided in Appendix III.

Shift Summaries:

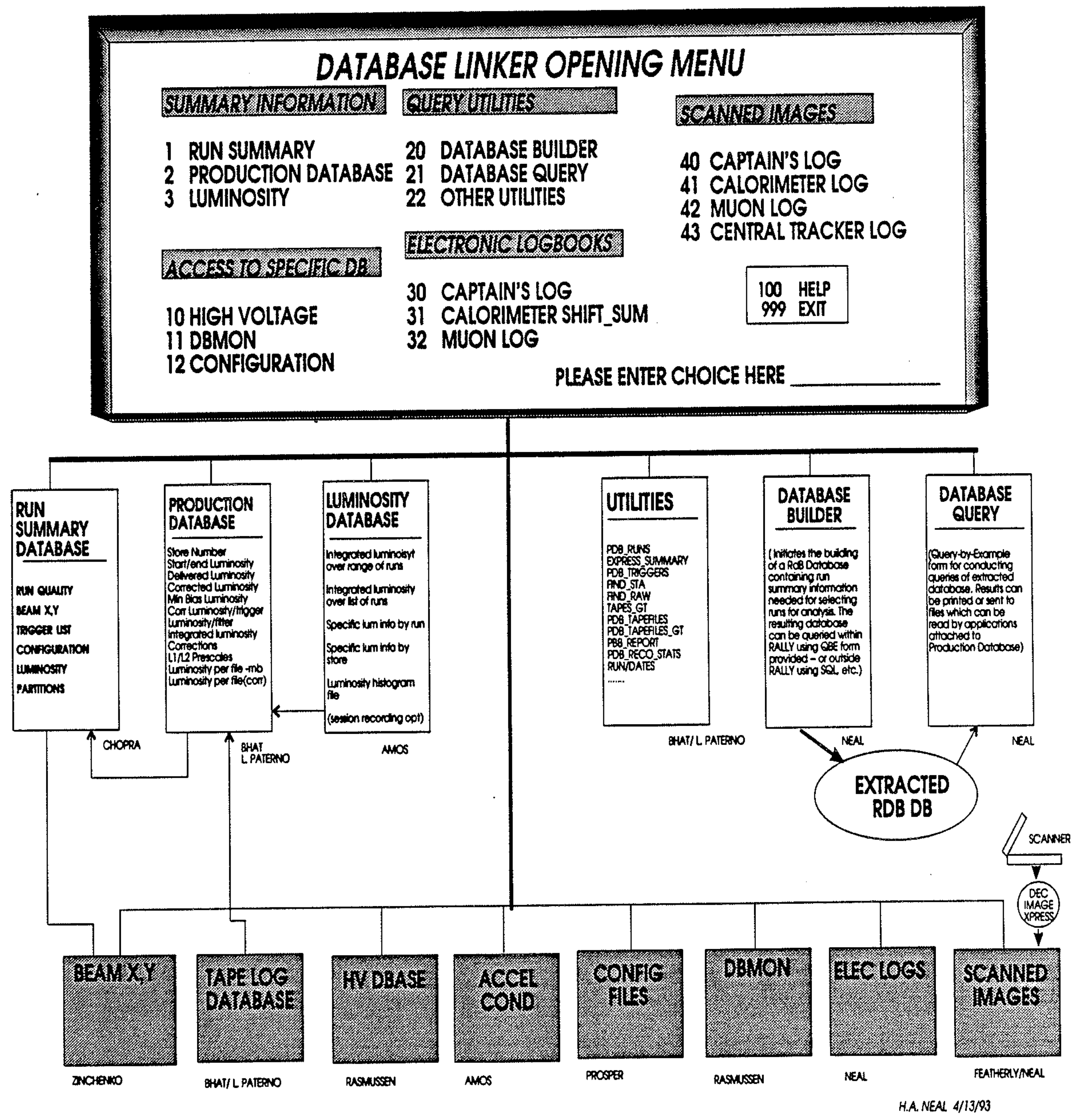
Several sub-systems in D0 maintain run-time electronic logs. In the present version of DBLINKER (v1.0), two of these logs are accessible -- the Calorimeter Shift Summary Log and the Muon Log. Basic characteristics of each run are recorded here by the physicists on the respective sub-system shifts. These logs (or rather their aliases) can be opened by making the appropriate menu choice from the DBLINKER main menu, and information about a given run is automatically located once the run number is specified. Normally, this information would not be accessed by a user after the run --- unless one was on a troubleshooting mission, A novel TPU program developed to make this feature possible is illustrated in Appendix IV.

Application Architecture:

Access to the various data sources is accomplished through a set of menu choices within RALLY. The organization of the Linker is illustrated below.. One moves from one level of the application to the next by selecting the appropriate choice from the current menu.

As implied by the chart, one can easily migrate to any of the second tier packages by making the appropriate choice from the opening menu. For example, one can visit the run summary database, the production database, the luminosity database, the production database utilities, or query or even build RdB run summary database on the fly, by simply making the desired menu choice.

The data sources listed across the bottom of the figure are not directly accessed from DBLINKER at this time, but in several cases they do form the base for the other packages which are linked to the DBLINKER.



H.A. NEAL 4/13/93

In the sections below we will review the choices in the various menus:

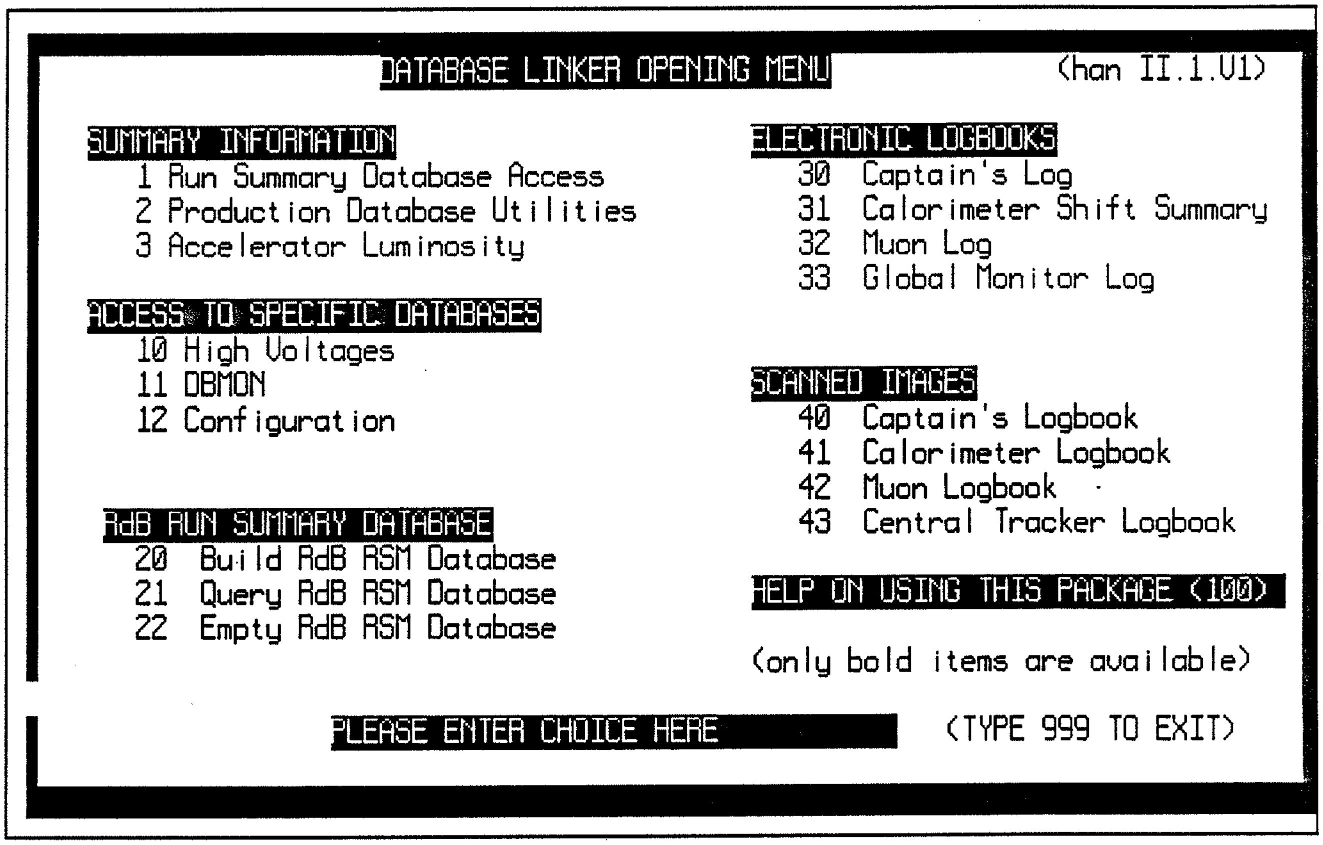
Main Menu:

The main menu provides a platform for accessing a variety of applications. One reaches the main menu by issuing the following sequence of commands from FNALD0 or D0:

```
$ SET DEF D0$BETA:[DBLINKER]
$ @SETUP_DBLINKER
$ DBLINK.
```

After responding to a "continuation query", one is presented with the screen below, which represents the main menu.

Examples of the choices available at this point are illustrated below:



```
                DATABASE LINKER OPENING MENU                (han II.1.V1)

SUMMARY INFORMATION                          ELECTRONIC LOGBOOKS
   1 Run Summary Database Access               30  Captain's Log
   2 Production Database Utilities             31  Calorimeter Shift Summary
   3 Accelerator Luminosity                    32  Muon Log
                                               33  Global Monitor Log
ACCESS TO SPECIFIC DATABASES
   10 High Voltages
   11 DBMON                                  SCANNED IMAGES
   12 Configuration                            40  Captain's Logbook
                                               41  Calorimeter Logbook
                                               42  Muon Logbook
RdB RUN SUMMARY DATABASE                       43  Central Tracker Logbook
   20  Build RdB RSM Database
   21  Query RdB RSM Database                HELP ON USING THIS PACKAGE (100)
   22  Empty RdB RSM Database
                                             (only bold items are available)

             PLEASE ENTER CHOICE HERE              (TYPE 999 TO EXIT)
```

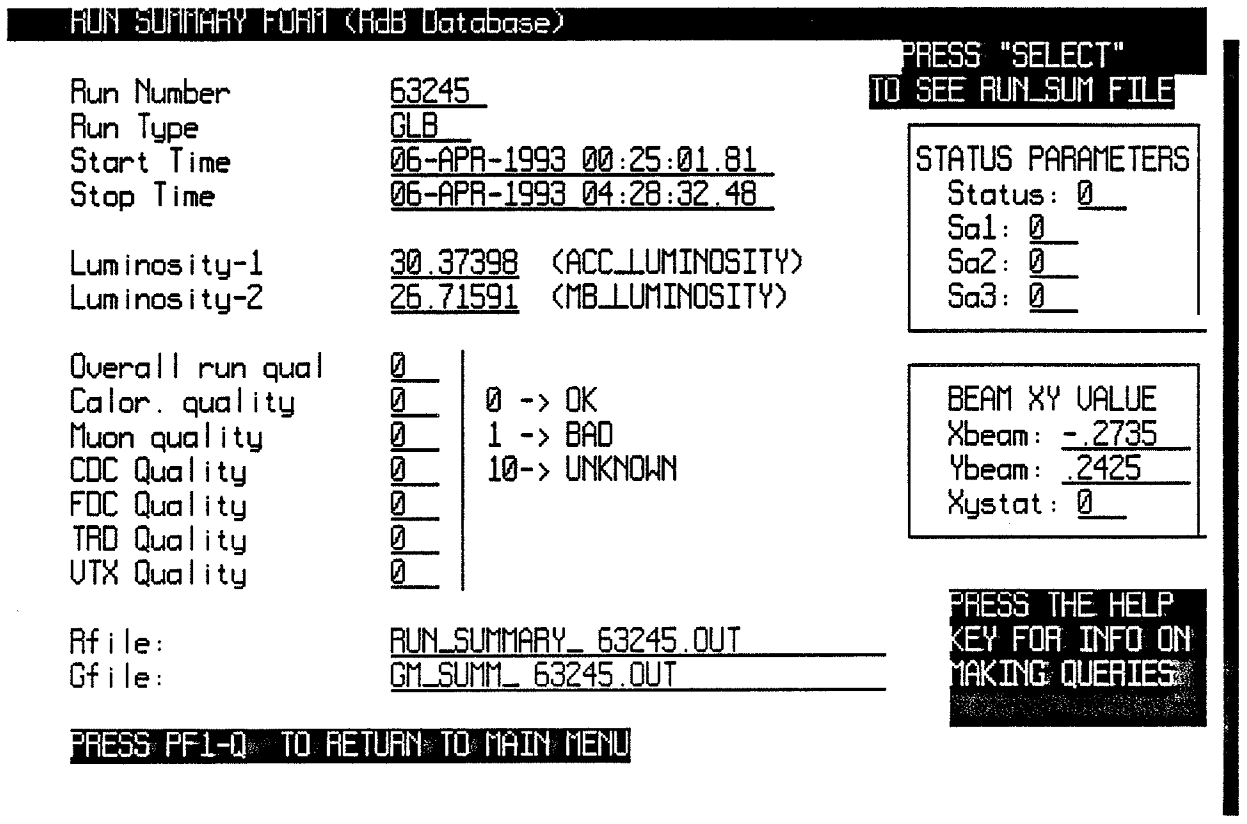
- the individual run summary sheets [choice #1], where one specifies a run number and is then taken to a sheet containing a variety of run quality, luminosity, date-time,..information. Also, from the run summary sheet an additional key sequence ("Select") takes one to the global monitor log for that run.

- the extracted run summary database [choice #21], permitting one to scan or conduct complex queries on an extracted RdB database. This database is practically the same as the Run Summary DBL3 database, except it is in a queryable, RdB form, making fast queries possible. The extracted database is assembled using the ADL (application development language) package which is a part of RALLY. That program loops over all data runs, extracting information about each run from DBL3 sources, and then writes the desired information into a standard RdB database. Normally this database building routine will be periodically executed by the system administrator, though it may eventually be made automatically self-updating.(An example of the ADL program used for building this RdB database is given in Appendix V)

- the calorimeter run summary file [choice #31], permitting one to specify a run number and then be taken, via the EVE editor, to any information in the run summary log pertaining to that run. One is left inside the file in EVE, and thus scanning, copying, logging, and printing operations are available, as well as further searching on keywords consisting of run numbers or other strings.
- the muon run summary file, permitting one to specify a run number and then be taken, via the EVE editor, to any information in the muon run summary log pertaining to that run.
- the experimental logbook image display[choice #41], illustrating what is possible should it be decided at a later time to scan individual logs. A later section of this note provides more details..

In the following section we present several of the second-tier forms and elaborate on their contents.

RUN SUMMARY DATABASE FORM (Main Menu Choice # 1)



```
RUN SUMMARY FORM (RdB Database)
                                                          PRESS "SELECT"
Run Number        63245                                   TO SEE RUN_SUM FILE
Run Type          GLB
Start Time        06-APR-1993 00:25:01.81                 STATUS PARAMETERS
Stop Time         06-APR-1993 04:28:32.48                   Status: 0
                                                            Sa1: 0
Luminosity-1      30.37398  (ACC_LUMINOSITY)                Sa2: 0
Luminosity-2      26.71591  (MB_LUMINOSITY)                 Sa3: 0

Overall run qual  0  |
Calor. quality    0  |  0 -> OK                             BEAM XY VALUE
Muon quality      0  |  1 -> BAD                            Xbeam: -.2735
CDC Quality       0  |  10-> UNKNOWN                        Ybeam: .2425
FDC Quality       0  |                                      Xystat: 0
TRD Quality       0  |
VTX Quality       0  |
                                                          PRESS THE HELP
Rfile:            RUN_SUMMARY_ 63245.OUT                  KEY FOR INFO ON
Gfile:            GM_SUMM_ 63245.OUT                      MAKING QUERIES

PRESS PF1-Q  TO RETURN TO MAIN MENU
```

One arrives at the Run Summary Database Form by selecting menu choice #1 from the main menu. After entering the run number of interest in the first field and, waiting a few seconds, while a Fortran routine opens the run summary database and fetches the relevant information, all fields are filled. These fields include variables such as the accelerator luminosity (Luminosity-1) and the minimum - bias luminosity (Luminosity-2), a list of the run quality parameters as entered into the global monitor log during the shift, with 0 representing a perfect run, 1-9 being increasing degrees of deficiency, and 10 representing a status of "unknown". The beam xy value is also given, along with a status parameter indicating if the xy value were calculated (Xystat =0) or if an average value were used (Xystat = 1).

One can descend to a another level of detail by pressing the "Select" key on the keyboard. This opens, in EVE, the global summary file (GM_SUMM_.....), providing access to information such as illustrated on the next page. (Some current E-MAIL exchanged within the Global Monitor group bearing on the DBLINKER is reproduced in Appendix VI). *Note: the luminosity information in this summary is approximate and is not to be used for final calculations*

Example of GLOBAL RUN SUMMARY OUTPUT:

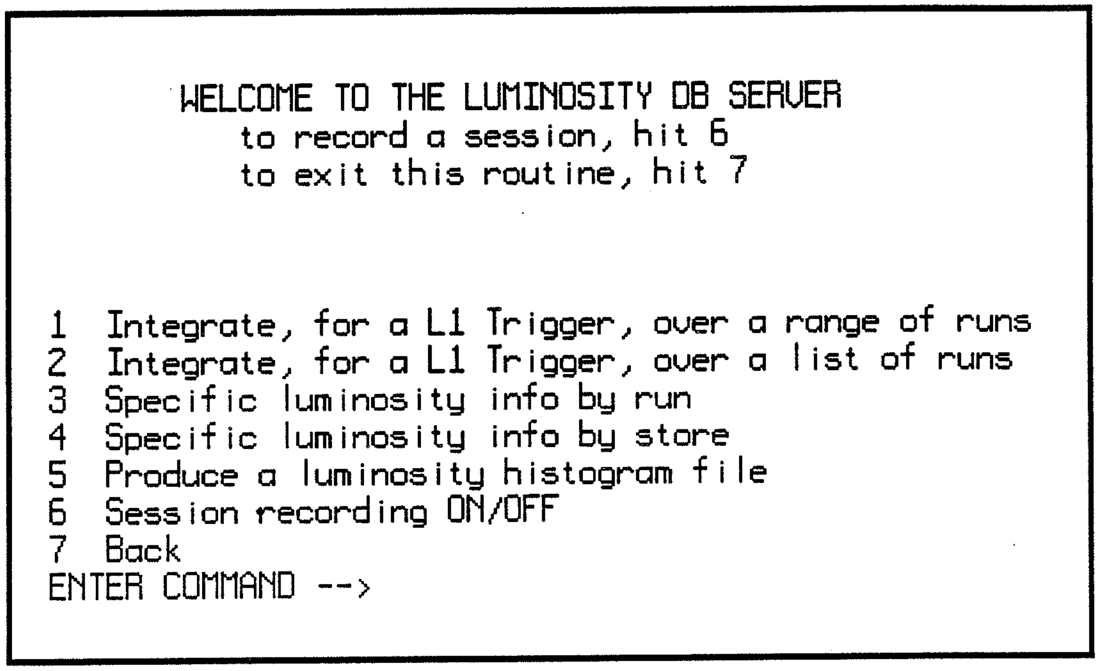
(Obtained by Pressing "Select" key while viewing Run Summary Form)

```
Run #  63245   Date:  6-APR-1993 00:25:01.81  Duration:     0 04:03:30.67
GL Log  5 page 152   Keyword: Beam-Beam
Configuration:  V72.GLB (V72-04E30)
Begin comment: v72-04e30                         5.4 e 30
Evaluation Good Run         End comment: Change prescales - retrig SUPR

 ACCELERATOR DATA
 TEV Store : 4340.       TEV COGSUM :  -55.95

 Toroid Current:  2433.38 SAMUS Current:   999.48

 BUNCH INTENSITIES
 P1    :  88.98 E9    A1    :  50.93 E9
 P2    :  95.71 E9    A2    :  57.89 E9
 P3    :  94.19 E9    A3    :  56.38 E9
 P4    : 100.98 E9    A4    :  53.41 E9
 P5    : 101.06 E9    A5    :  48.46 E9
 P6    :  99.46 E9    A6    :  38.83 E9
 P_AVE : 96.73 E9   A_AVE :  50.98 E9

 P HALO beg:  0.131 end:   0.121 kHz ->   1.303 Hz/E9
 A HALO beg:  0.009 end:   0.008 kHz ->   0.167 Hz/E9
 MR TOP beg:  0.852 end:   0.543 kHz
 MR BTM beg:  0.158 end:   0.096 kHz

 Starting Lum :  5.403 E30 /(CM2 SEC)      Ending Lum :  4.368 E30 /(CM2 SEC)
 Correction:  1.408 Delivered Lum: 70.857 /nBARN  Lum->Tape: 30.226 /nBARN

 DEADTIME SOURCES:
 L2*FEB*COOR*MRBS*uBLANK = 0.715*0.930*0.833*0.833*0.903 = 0.417

  Stream         Events    Parts (on tape)   Tapes
  ALL            14872      60     ( 60)     WM3627 WM3630 WM3628 WM3633 WM3632 WM
  EXPRESS        1256        5     (  5)     WM3631
  INSPILL        105         1     (  1)     WM3635

   Trigger          Presc Cnt  Rte Xsct      Filter           Presc Cnt  Pss Rte Xsct
                               (Hz)(b)                                   (%) *10 (b)
 1 ZERO_BIAS-V72   250k 7.2k 1.1 59.m    1 ZERO_BIAS-V72     100   75 1.0 .12 62.m
                                        32 MIN_BIAS           10  343 .47 .55 28.m
 3 MU_1_HIGH          5 367k 59. 60.u    3 MU_HIGH             1  854 <1c 1.3 141n
 4 MU_1_MAX           1 1.8M 295 60.u    4 MU_MAX              1 1.1k <1c 1.7 35.n
                                        33 MU_2_MAX            1  573 <1c .92 18.n
5 MU_2_LOW          1 137k 22. 4.5u    5 MU_2_LOW            1   44 <1c .07 1.4n

 6 MU_2_HIGH          1  84k 13. 2.7u    6 MU_2_HIGH           1  496 <1c .79 16.n
```

GLOBAL RUN SUMMARY OUTPUT (continued)

```
 7 MU_1_JET            5  27k 4.3 4.4u   7 MU_1_JET          1 2.0k .07 3.2 329n
 8 MU_EM_1             1 3.6k .58 120n   8 MU_ELE            1  162 .04 .26 5.3n
 9 MU_EM_2             1  23k 3.7 768n   9 MU_ELE_2          1  489 .02 .78 16.n
10 MU_JET_HIGH         1  23k 3.7 776n  10 MU_JET_HIGH       1  550 .02 .88 18.n
                                        34 MU_JET_MAX        1  170 <1c .27 5.6n
                                        35 MU_JET_MED        1   19 <1c .03 628p
11 MU_JET_LOW          1  21k 3.4 700n  11 MU_JET_LOW        1 3.7k .17 5.9 123n
12 SCALAR_ET           1 3.0k .48 99.n  12 SCALAR_ET         1  218 .07 .35 7.2n
13 MISSING_ET          1  22k 3.5 737n  13 MISSING_ET        1  225 .01 .36 7.4n
                                        36 MISS_ET_MON      10   42 .01 .06 13.n
14 JET_3_MISS          1  16k 2.5 516n  14 JET_3_MISS        1   88 <1c .14 2.9n
15 JET_1_MISS          2  35k 5.6 2.3u  15 JET_MISS_TAU      1   44 <1c .07 2.9n
16 JET_2_GAP           1  54k 8.7 1.7u  16 JET_GAP_MED       1  370 <1c .59 12.n
17 JET_2_END           5  11k 1.7 1.8u  17 JET_END_MED       1  189 .01 .30 31.n
18 JET_1_LOW        1.0k  25k 4.0 832u  18 JET_MIN          10  104 .04 .16 34.u
                                        37 JET_LOW           1  139 <1c .22 4.5u
19 JET_2_HIGH         10 4.9k .78 1.6u  19 JET_MEDIUM        1  320 .06 .51 105n
20 JET_3_HIGH          1  36k 5.7 1.1u  20 JET_HIGH          1  525 .01 .84 17.n
21 JET_4_MED           1  46k 7.3 1.5u  21 JET_MAX           1   85 <1c .13 2.8n
                                        38 JET_MULTI         1  467 .01 .75 15.n
                                        39 JET_MULTI_X       1   60 <1c .09 1.9n
                                        40 SCALAR_ET_JT      5  202 .02 .32 33.n
23 EM_1_MED            1 165k 26. 5.4u  23 ESC_HIGH          1  166 <1c .26 5.4n
                                        41 GAM_HIGH_ISO      1  490 <1c .78 16.n
                                        42 ELE_HIGH          1 2.4k .01 3.8 78.n
                                        43 ELE_HIGH_TRK      1  727 <1c 1.1 24.n
24 EM_1_HIGH           1 140k 22. 4.6u  24 ELE_MAX           1   89 <1c .14 2.9n
```

ACCELERATOR LUMINOSITY DATABASE FORM:
(MAIN MENU Choice #3)


```
        WELCOME TO THE LUMINOSITY DB SERVER
           to record a session, hit 6
           to exit this routine, hit 7



1  Integrate, for a L1 Trigger, over a range of runs
2  Integrate, for a L1 Trigger, over a list of runs
3  Specific luminosity info by run
4  Specific luminosity info by store
5  Produce a luminosity histogram file
6  Session recording ON/OFF
7  Back
ENTER COMMAND -->
```

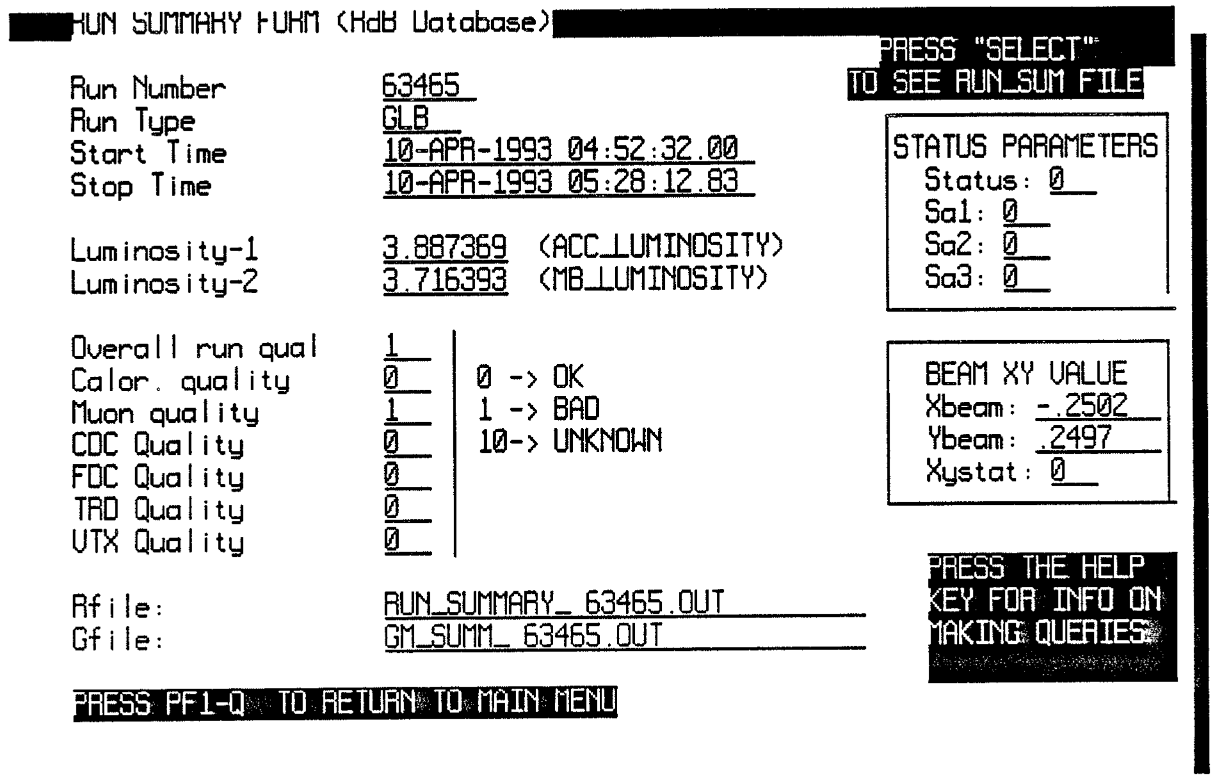
This screen is presented to the user after the selection of menu choice #3. At this stage one can select any of the options for utilizing the accelerator luminosity package developed by Norman Amos. Selection #1 provides the user with an integrated luminosity for a L1 trigger covering a range of runs. In order to obtain similar information for a list of runs, one chooses menu selection #2. Choices #3 and #4 provide specific luminosity information by run and by store, respectively. Menu choice #5 gives the user a luminosity histogram file. The entire session of engagement with the package can be logged to a file by selecting menu choice #6. By entering a 7, one is then taken back to the main menu.

More details about this package are given in Appendices II.

Example of Output from Luminosity Program
(Actions Invoked from the Accelerator Luminosity Form)

```
ENTER COMMAND -> 2
WARNING *** CHECK TO CONFIRM ALL RUN #'s ARE PRESENT
ENTER L1 SPECIFIC TRIGGER (0:31) -> 30
ENTER STARTING RUN # -> 63000
ENTER STOPPING RUN # -> 64000
RUN     TAPE     TOTAL
63026   29.78    29.783
63027   24.83    54.611
63028   23.66    78.273
63030    7.31    85.581
63033    4.20    89.783
63034    0.95    90.730
63038    1.71    92.444
63042    1.38    93.822
63057   17.12   110.938
63066   29.04   139.975
63068   13.79   153.765
63069   29.91   183.672
63070   20.54   204.210
63071   22.87   227.077
63072   12.46   239.542
63084    0.00   239.542
63092   16.44   255.985


1  Integrate, for a L1 Trigger, over a range of runs
2  Integrate, for a L1 Trigger, over a list of runs
3  Specific luminosity info by run
4  Specific luminosity info by store
5  Produce a luminosity histogram file
6  Session recording ON/OFF
7  Back


ENTER COMMAND -> 3
ENTER RUN -> 63245
RUN    STORE  START    STOP    START_L STOP_L  DELVR  CHECK  TAPE
63245  4340 462.018 462.187    5.4     4.4     70.6    71.2    30.4
 L2,FEB,COOR,MRBS,uBLANK
 0.74 0.94 0.85 0.85 0.91
MULT, uBLANK, CENTZ, SINGZ
 1.415 0.926 0.340 0.630
MEASURED PRESCALES
0.000E+00 0.308E+06 0.000E+00 0.502E+01 0.100E+01 0.100E+01 0.100E+01 0.507E+01
0.100E+01 0.100E+01 0.100E+01 0.100E+01 0.100E+01 0.100E+01 0.100E+01 0.200E+01
0.100E+01 0.501E+01 0.939E+03 0.103E+02 0.100E+01 0.100E+01 0.000E+00 0.100E+01
0.100E+01 0.100E+01 0.100E+01 0.496E+01 0.100E+01 0.100E+01 0.100E+01 0.297E+07
EXPECTED PRESCALES
0.500E+05 0.250E+06 0.000E+00 0.500E+01 0.100E+01 0.100E+01 0.100E+01 0.500E+01
0.100E+01 0.100E+01 0.100E+01 0.100E+01 0.100E+01 0.100E+01 0.100E+01 0.200E+01
0.100E+01 0.500E+01 0.100E+04 0.100E+02 0.100E+01 0.100E+01 0.000E+00 0.100E+01
0.100E+01 0.100E+01 0.100E+01 0.500E+01 0.100E+01 0.100E+01 0.100E+01 0.000E+00
```

RUN SUMMARY FORM (RdB Database) - Main Menu Choice #21)

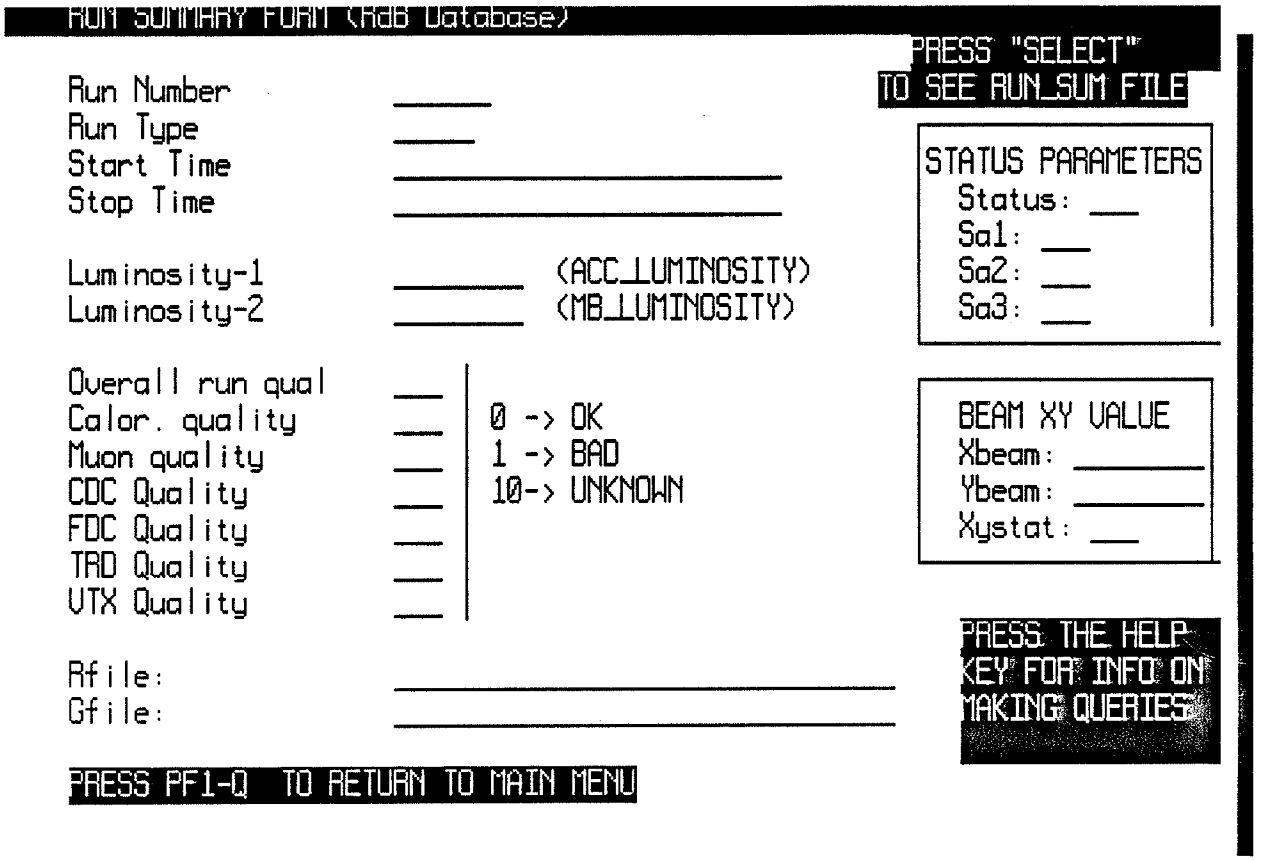


```
RUN SUMMARY FORM (RdB Database)
                                                          PRESS "SELECT"
Run Number        63465                                TO SEE RUN_SUM FILE
Run Type          GLB
Start Time        10-APR-1993 04:52:32.00               STATUS PARAMETERS
Stop Time         10-APR-1993 05:28:12.83                 Status: 0
                                                          Sa1: 0
Luminosity-1      3.887369  (ACC_LUMINOSITY)              Sa2: 0
Luminosity-2      3.716393  (MB_LUMINOSITY)               Sa3: 0

Overall run qual  1  |                                  BEAM XY VALUE
Calor. quality    0  |  0 -> OK                         Xbeam: -.2502
Muon quality      1  |  1 -> BAD                        Ybeam: .2497
CDC Quality       0  |  10-> UNKNOWN                    Xystat: 0
FDC Quality       0  |
TRD Quality       0  |
UTX Quality       0  |                                  PRESS THE HELP
                                                        KEY FOR INFO ON
Rfile:            RUN_SUMMARY_ 63465.OUT                MAKING QUERIES
Gfile:            GM_SUMM_ 63465.OUT

PRESS PF1-Q  TO RETURN TO MAIN MENU
```

This form, which is presented after one selects menu choice #21 from the main menu, has all of the appearances of the run summary database form reached by menu choice #1, but there is an important difference. The above form is actually coupled to a constructed RdB database containing the run summary information. The RdB database presently is built by the database administrator, using main menu choice #20, which invokes a repetitive call to the basic run summary package, each time extracting information about the run and writing that information into the RdB database. (The initial program developed by the author for this purpose is presented in Appendix V).

The added benefit to the user of accessing the RdB database, rather than making single inquiries via menu choice #1, is that the rapid queries and browsing through numerous runs is much easier. The details of how to make a query are presented on the next page. A description of the various parameters above can be found in the discussion of the run summary previously given..

Run Summary Query Form (Main Menu Choice #20)

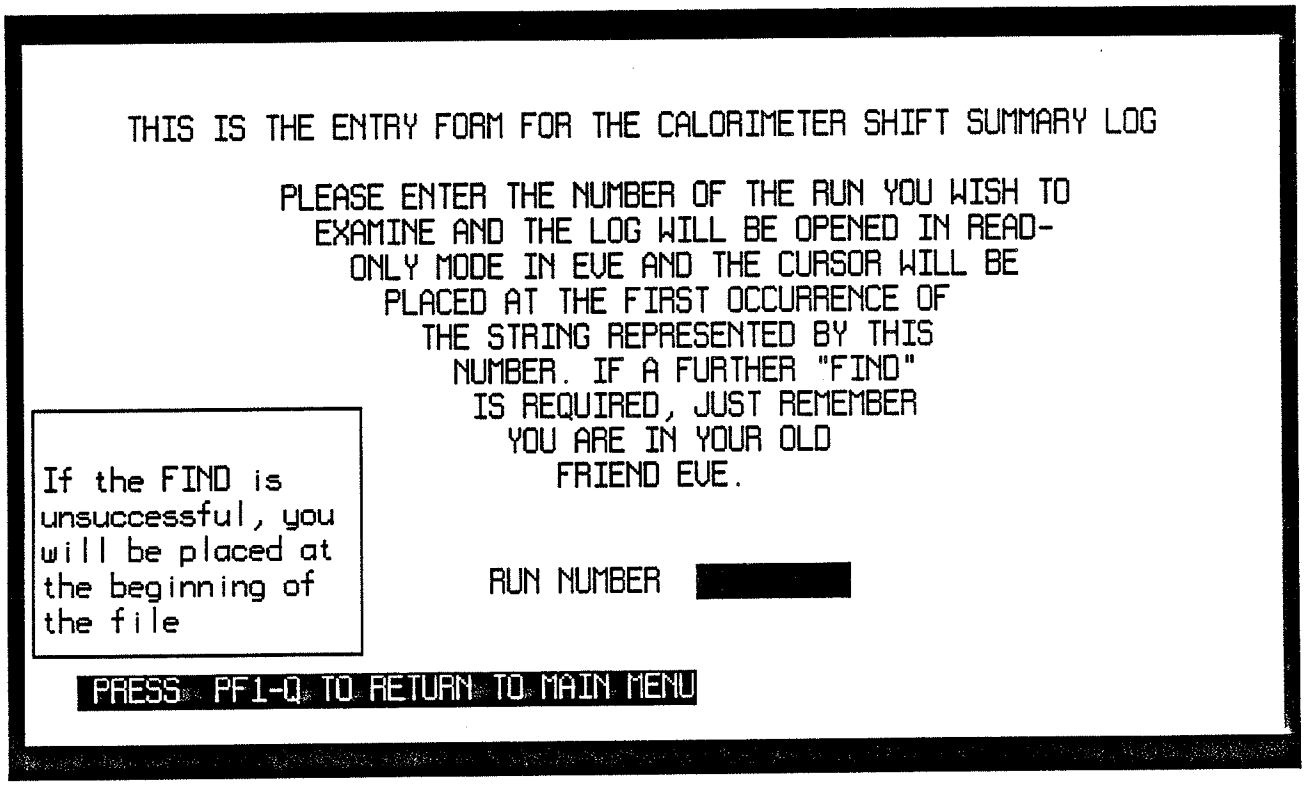


```
RUN SUMMARY FORM (RdB Database)
                                                    PRESS "SELECT"
Run Number          ____                           TO SEE RUN_SUM FILE
Run Type            ___
Start Time          _______________                STATUS PARAMETERS
Stop Time           _______________                  Status: __
                                                     Sa1: __
Luminosity-1        _____  (ACC_LUMINOSITY)          Sa2: __
Luminosity-2        _____  (MB_LUMINOSITY)           Sa3: __

Overall run qual    __                               BEAM XY VALUE
Calor. quality      __   0 -> OK                     Xbeam: _____
Muon quality        __   1 -> BAD                    Ybeam: _____
CDC Quality         __   10-> UNKNOWN                Xystat: __
FDC Quality         __
TRD Quality         __
VTX Quality         __                               PRESS THE HELP
                                                     KEY FOR INFO ON
Rfile:              __________________               MAKING QUERIES
Gfile:              __________________

PRESS PF1-Q  TO RETURN TO MAIN MENU
```

From the previous form (Run Summary RdB Form, Choice #21), one presses the "Do" key to get to the command line, and then types "Set Query" to enter the query mode. Then the form above appears, with all fields empty. At this point, one uses the tab keys to move to the field of interest and then enters the values to be used in conducting the query. For example, to get a listing of all runs that are "perfect", one moves to the "Overall run qual" field and types a "0". Then one again presses the "Do" key to get to the command line, and then types "Execute Query". At this point the record stream available to the user for browsing (use arrow keys to go from one run to the other), or for printing, is the set of all runs for which the field "Overall run qual" is 0.

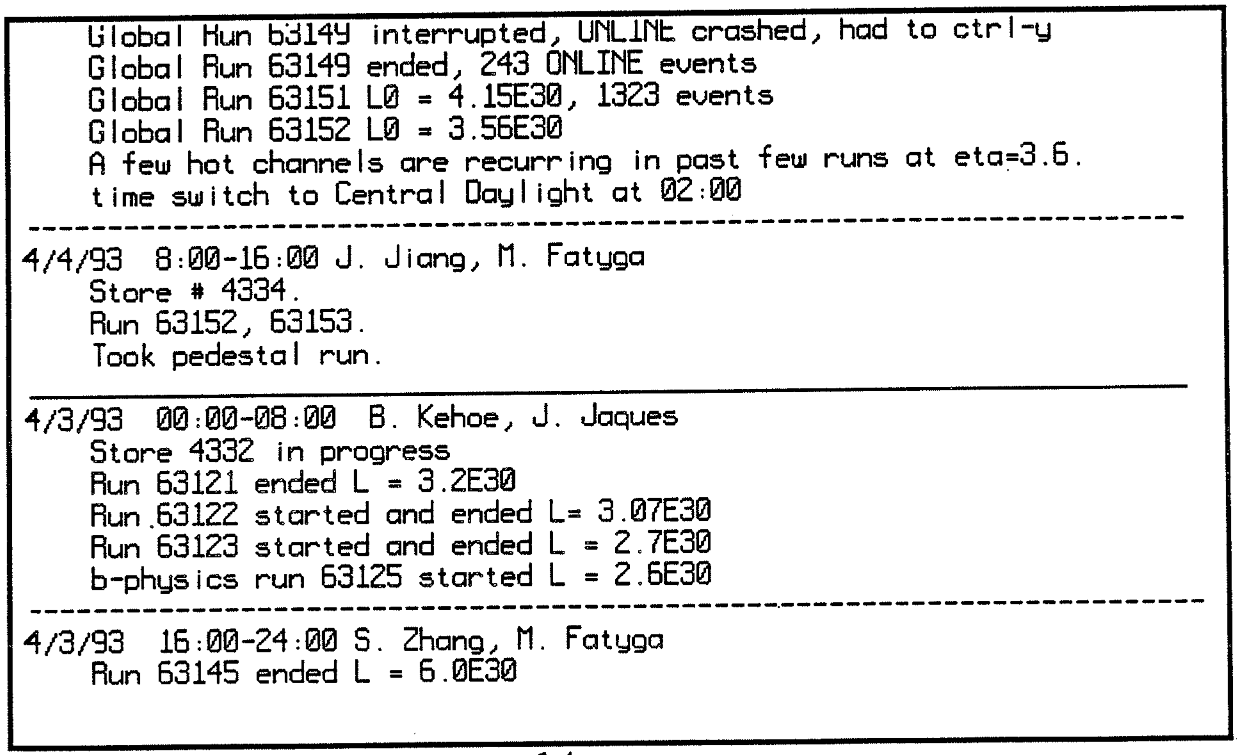
For more complicated queries, where not only equality can be demanded, but all of the standard Boolean conditions as well, one goes through the same process as above, except when a condition is to be applied to a field, one moves the cursor to the field in question, goes to the command line ("Do") and types "Extend Query", and then types the conditions to be applied, using <,>, etc. as appropriate. This powerful capability allows ones to select out runs on the basis of their quality (overall, or for particular sub-systems), luminosities, beam positions, etc.

Calorimeter Shift Summary Form (Main Menu Choice #31)



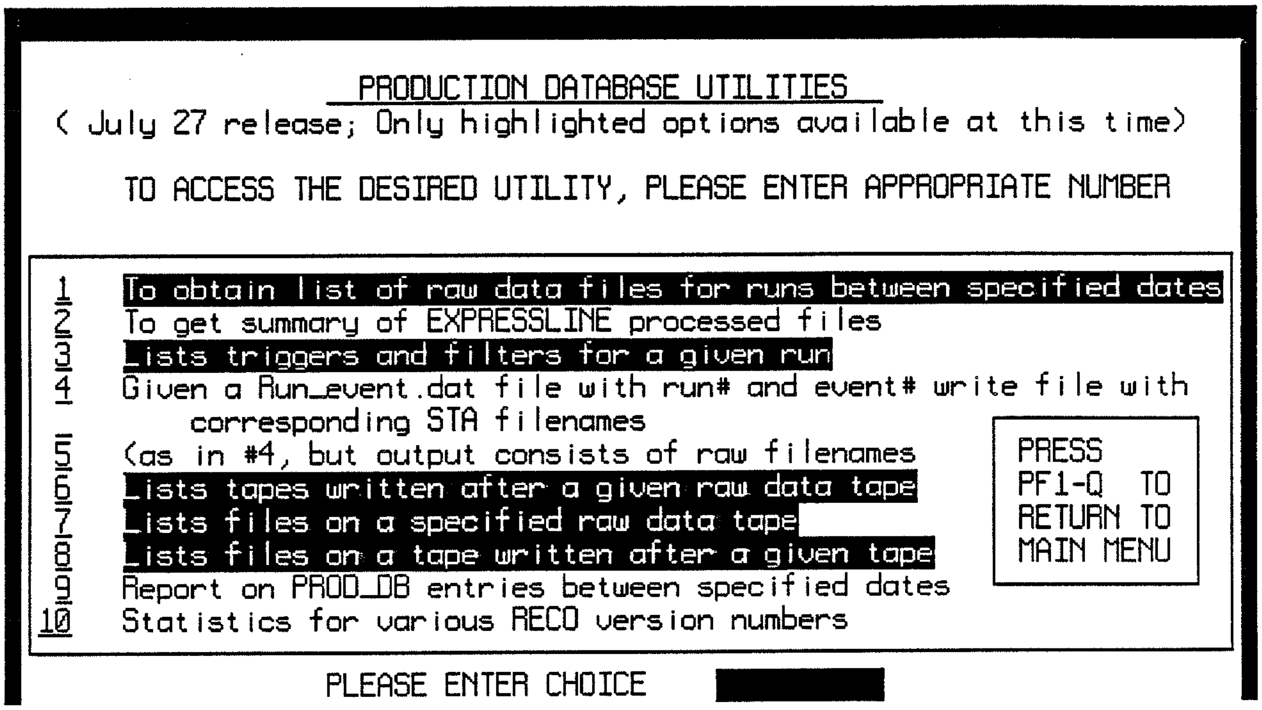
```
THIS IS THE ENTRY FORM FOR THE CALORIMETER SHIFT SUMMARY LOG

          PLEASE ENTER THE NUMBER OF THE RUN YOU WISH TO
           EXAMINE AND THE LOG WILL BE OPENED IN READ-
            ONLY MODE IN EVE AND THE CURSOR WILL BE
             PLACED AT THE FIRST OCCURRENCE OF
              THE STRING REPRESENTED BY THIS
               NUMBER. IF A FURTHER "FIND"
                IS REQUIRED, JUST REMEMBER
                 YOU ARE IN YOUR OLD
                   FRIEND EVE.

If the FIND is
unsuccessful, you
will be placed at
the beginning of           RUN NUMBER
the file

PRESS PF1-Q TO RETURN TO MAIN MENU
```

After entering the run number 63151 above and pressing return, the following information is presented about the run from the calorimeter shift log.



```
    Global Run 63149 interrupted, ONLINE crashed, had to ctrl-y
    Global Run 63149 ended, 243 ONLINE events
    Global Run 63151 L0 = 4.15E30, 1323 events
    Global Run 63152 L0 = 3.56E30
    A few hot channels are recurring in past few runs at eta=3.6.
    time switch to Central Daylight at 02:00
------------------------------------------------------------------------
4/4/93  8:00-16:00 J. Jiang, M. Fatyga
    Store # 4334.
    Run 63152, 63153.
    Took pedestal run.
________________________________________________________________________
4/3/93  00:00-08:00  B. Kehoe, J. Jaques
    Store 4332 in progress
    Run 63121 ended L = 3.2E30
    Run 63122 started and ended L= 3.07E30
    Run 63123 started and ended L = 2.7E30
    b-physics run 63125 started L = 2.6E30
------------------------------------------------------------------------
4/3/93  16:00-24:00 S. Zhang, M. Fatyga
    Run 63145 ended L = 6.0E30
```

Production Database Utility Form (Main Menu Choice #2)



```
                PRODUCTION DATABASE UTILITIES
( July 27 release; Only highlighted options available at this time)

     TO ACCESS THE DESIRED UTILITY, PLEASE ENTER APPROPRIATE NUMBER

 1   To obtain list of raw data files for runs between specified dates
 2   To get summary of EXPRESSLINE processed files
 3   Lists triggers and filters for a given run
 4   Given a Run_event.dat file with run# and event# write file with
        corresponding STA filenames
 5   (as in #4, but output consists of raw filenames         PRESS
 6   Lists tapes written after a given raw data tape          PF1-Q  TO
 7   Lists files on a specified raw data tape                 RETURN TO
 8   Lists files on a tape written after a given tape         MAIN MENU
 9   Report on PROD_DB entries between specified dates
10   Statistics for various RECO version numbers

              PLEASE ENTER CHOICE
```

By selecting menu choice #2 from the Main Menu, one arrives at the form presented above. From this form one can access all of the production database utilities released by Puspha Bhat (et. al). This platform permits one to extract the list of raw data files for all runs between designated dates, a list of triggers and filters for a given run, tapes written after a given raw data tape, information about the various RECO version numbers, and so on. The information is written to a file designated by the user.

More details about the production database can be found in Appendix III.

Comments on the 4GL Package RALLY:

Individuals wishing to learn more about RALLY should consult the appropriate DEC documentation. Some comments are included in this section, however, about special features of RALLY that were utilized in developing the current application.

First, it should be noted that one is immediately placed into the world of RALLY by typing $RALLY CREATE Filename. This statement creates a RALLY file "Filename.RGA" which will contain all components of the application. After this command is executed, one is presented with an intial screen from which RdB databases can be created, forms and menus can be created and linked to databases, ADL procedures can be written, etc.

An existing application can be edited at a later time by simply typing $RALLY EDIT FILENAME. And, as indicated earlier in the note, an application can be run by typing $RALLY RUN Filename where, in each case, "Filename" is the name of the application. The next section discusses some of the utilities available to the user when running a RALLY application.

If there is any non-trivial part of the development of a RALLY application the first time, it is in manuering around the intricacies of forming external links and the shareable images needed to permit a RALLY action to invoke an external routine. Once the details are mastered, however, they can be repeatedly applied with ease.

To create an external program link it is necessary to create the external routine, create a formal external program link and then assign the external program link to a RALLY action site. The fields of a form or menu might represent such a site. Thus, one can have an external Fortran program called when the cursor is moved to a certain field of a form -- and the contents of various fields on the form can be passed as values to the external routine. Results from the calulcations of the external routine can also be passed back to the form and cause the filling of a different (or the same) set of fields on the form -- even if the fields are elements of a RdB database. Much of the power of RALLY is buried in this very feature. The ease of creating and linking forms and menus adds to the overall utility of the package.

Another item worthy of note is need to make sure that the external routines are linked as shareable images. Moreover, every PSECT in the routine must be given the attribute NOSHR. For a massive library, such as the CERN Library, which has not been written to be shareable, the user must take the tedious steps to include a statement changing the attribute for a number of PSECTS during linking. This tedium can be moderated by making a link map and using a TPU package to search for the PSECTs in question and making the appropriate attribute change via a "Replace" command.Though this discussion is verging on become quite technical, one more piece of advice may be of use to the first-time application-developer. In placing routines into shareable images one should write a macro file to define the vectors for each routine, and then a link command file should be prepared to set up any linker libraries, to assemble the macro code which forms the routine vectors within the shareable image, and to link the image. The author would be pleased to provide any interested collaboration member with the details of this process if desired.

RALLY Utilities:

If one could only get at the RALLY forms via the screen display, their utility would be limited. But the fact is that many powerful report features are available to the user by simply going to the command line by pressing the "Do" key. I will summarize a few of these and other utilities, and refer the reader to the RALLY manuals for a more complete description.

Print Screen (Gold-KP1)
Sends a copy of the current screen image to the system printer

Spawn (Gold-$)
Creates a VMS subprocess of the current RALLY operation.

Write Screen
Sends a copy of the current screen image to a file the user specifies on the command line

Print Report
Sends a copy of the current form to the system printer (i.e., all records in the stream)

Print Rest_of_report
Sends a copy of the remaining pages of the current form/report to the system printer.

Write Report
Sends a copy of the form to a user specified file.

Write Rest_of_report
Sends a copy of the remaining pages of a form to a file specified by the user.

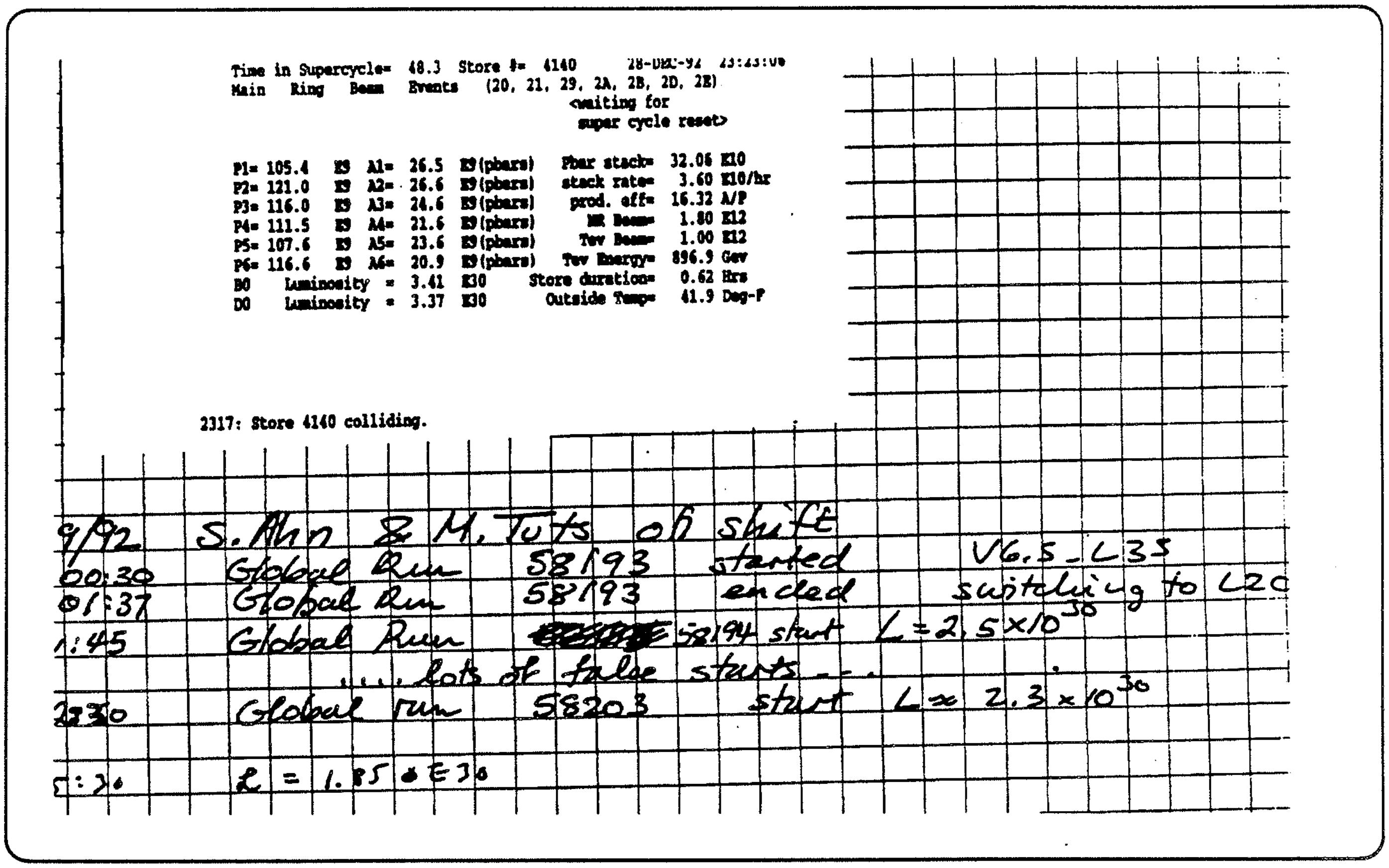
There are literally dozens of other commands, most of which the D0 user would never require. We are only using a small fraction of the features of RALLY, which permits multiple tasks running in different windows, etc.

Image Scanning Options:

During discussions with members of the collaboration we learned that many desired a method to remotely access logbooks. The author and John Featherly looked into the possibility of actually scanning each log book, entering each image into a properly indexed database, and then using the Database Linker to access images upon demand. Though this challenge seemed daunting at the outset, the fact is that the goal is entirely reasonable. Advancements which bring such a task to within reason include, the availability of the DEC product Image Express, which fully manages the scanning of images into an indexed database, along with the very low cost of reliable disk storage and the very efficient packing of the DEC DDIF format.

A page out of a log book requires approximately 150 blocks of storage for a 300 dpi resolution. A hundred log books would fit on a $2500 disk.

The image below results from the selection of menu choice #41 from the main menu.



```
Time in Supercycle=  48.3  Store #=  4140      28-DEC-92  23:43:00
Main   Ring   Beam   Events   (20, 21, 29, 2A, 2B, 2D, 2E)
                                           <waiting for
                                            super cycle reset>

P1= 105.4   E9  A1=  26.5  E9(pbars)   Pbar stack=  32.06 E10
P2= 121.0   E9  A2=  26.6  E9(pbars)   stack rate=   3.60 E10/hr
P3= 116.0   E9  A3=  24.6  E9(pbars)    prod. eff=  16.32 A/P
P4= 111.5   E9  A4=  21.6  E9(pbars)      MR Beam=   1.80 E12
P5= 107.6   E9  A5=  23.6  E9(pbars)     Tev Beam=   1.00 E12
P6= 116.6   E9  A6=  20.9  E9(pbars)   Tev Energy=  896.9 Gev
B0    Luminosity  =  3.41  E30     Store duration=   0.62 Hrs
D0    Luminosity  =  3.37  E30       Outside Temp=  41.9 Deg-F

2317: Store 4140 colliding.
```

9/92 S. Ahn & M. Tuts on shift

00:30 Global Run 58193 started V6.5_L35

01:37 Global Run 58193 ended switching to L2C

1:45 Global Run ~~[illegible]~~ 58194 start $L = 2.5 \times 10^{30}$

.... lots of false starts ...

2330 Global run 58203 start $L \approx 2.3 \times 10^{30}$

5:30 $\mathcal{L} = 1.85 E30$

The syntax of the command used to fetch an image is very straightforward. In the package provided by DEC with the DECImageExpress application sample display programs are included. After properly setting up the display on an X-terminal, the issuance of a sequence similar to ...

```
$ivp2 :== sys$common:[syshlp.examples.ibs]ivp.exe
$ivp2 drawer_name, index_1,index_2,index_3, index_4...
```

will generate the image corresponding to the drawer and indices specified. For us, the drawer names could be Run Ia, Run Ib,..... The second index could be "captain's log", "calorimeter log", ... The third and fourth indices could be logbook volume and page number, which could both be supplied by a relations table driven by the run number.

Thus, a collaborator in Paris could call up the Database Linker, specify that he or she wanted to see what was written in the calorimeter logbook about run 63245 and almost immediately be presented with a screen showing a high quality image of the relevant pages, with scrolling and copying options available.

We have tested the feasibility of this from various nodes around the country and found it to work and to be fast.

We have no present intention to proceed with the actual implementation of this project, but merely wanted to demonstrate its feasibility. It is feasible, but management attention would have to be provided by someone at the lab to oversee hired help doing the scanning, and conducting the necessary quality control checks.

Summary Remarks:

I trust that this summary provides the reader with a glimpse of what is possible using the RALLY 4GL package. It can provide a unifying force in a system employing many different types of databases with only a very small programming overhead, since every database must eventually have a subroutine that can access it in any case. RALLY can literally attach to these routines, drag variables into its own environment, do calculations involving those variables, and then pass out selected fields into forms, screens, or brand new databases.

This study over the past several months has permitted us to master RALLY, to generate or collect the routines needed to access the Run Summary Database, the Accelerator Luminosity Database, the beam x-y position, and to access the various electronic logbooks and shift summaries. And we have demonstrated the feasibility of integrating important optically scanned images into our workstation environment.

This work should provide the basis for deciding on what aspects of the above system should be incorproated into standard D0 operations.

=====

Acknowledgements:

Special thanks are in order to John Featherly, Sailesh Chopra, Norman Amos, and Chip Stewart. I also want to acknowledge the contributions of Pushpa Bhat, Lauri Paterno, A. Zinchenko, and Harrison Prosper.

Appendices:

Appendix I: D0 News Article

RELEASE OF DATABASE LINKER PROTOTYPE; August 18, 1993; H. Neal

==

The inital prototype for the Database Linker has been released. With this package, which is built around the DEC RALLY 4GL Application Development System, one can have direct access to:

the Run Summary Database (developed by S. Chopra)

the Accelerator Luminosity Database (developed by Norman Amos)

the Production Database Utilities (developed by Puspha Bhat and the PDB Group)

the Calorimeter Shift Summary Log

the Muon Run Summary Log

an extracted RdB database containing all Run Summary information for Run 1A (developed by H. Neal)

example of scanned log book images (developed by J. Featherly and H. Neal)

This product is the result of a request by Paul Grannis in February, 1993 that a working group be formed to identify how the key parameters for each run could be easily accessed by D0 collaborators. The challenge has been that these parameters have lived within the domains of various separate databases which do not communicate with each other. Some databases are RdB, some are DBL3, and some are indexed files. By going to a level above all of the existing databases and utilizing a package that is able to penetrate any type of database and extract the desired information, we have attempted to produce an interface that transparently provide users with the data they need to carry out their intended D0 analysis, without having to worry about the precise origin of the information.

After typing "1" to get by the welcoming menu, one is presented with a main menu from which access can be gained to any of the desired applications. A D0 Note is in preparation that describes each step in some detail. For purposes of this D0 News posting, it is probably most efficient to simply give the following examples illustrating how one would go about accessing specific items of information about the experiment.

===

Task 1:

Suppose the user wishes to know all of the run summary information about Run # 64086.

Action:

Type DBLINK to start RALLY. After the welcome screen appears type 1 to get to the main menu. At the main menu type 1 for Run Summary. Wait for the Run Summary screen, and then type in the run number (64086). The form is automatically filled in with all of the relevant information on the luminosities, detector status, etc. for this run. At this point the user can type in another run number to get information about that run, write the form to a file or print the form. (Keep in mind that all runs are not global runs!). To accomplish the write or print operations, simply press the "Do" key, and at the command prompt, type "write report filename" or "print report". To ascend once again to the main menu type <PF1>-q. From here, any of the other main menu options are again available.

===

Task 2:

The user wants to have an interactive session with Norm Amos' luminosity program.

Action:

From the main menu, type 3. After a few seconds a connection is established to Norm's program, where one can get information about luminosiites per run, luminosities integrated over a series of runs, etc. All of the output for a session can be logged to a local file for the user's later use. The program accessed is precisely the one that Norm has been providing collaborators to date. When the interactive session is complete, the user should press 7 and be returned to the main menu.

==============================

Task 3:

The user wants to find out what runs have been taken between date-1 and date-2, or get the list of triggers for a given run, or a list of tapes written after a given run, or a list of files on a specified raw data tape....

Action:

Go to the main menu, select # 2 (for Production Database Utilities), and then the user is taken via an additional RALLY menu to the utility package developed by Pushpa Bhat, where the user can ask any of the above questions (and others) and have the resultant answers written to a specified file.

==================================

Task 4:

The user wants to look at the Calorimeter Shift Summary log section relating to a specified run.

Action:

From the main menu, select 31. The user is taken to an explanatory form and is asked for the run number of interest. RALLY then opens the shift summary log alias in EVE and positions the cursor at the first occurence of the run number search string (run number). If this is not the spot the user wishes to be, then he or she can just do another "Find" in EVE. The access to EVE is READ/ONLY. When finished viewing, an exit from EVE will return the user back to the RALLY form for the logbook, from whence another logbook entry can be searched, or the user can return to the main menu (<PF1>-q).

(an identical process allows one to search the muon run log)

=================================

Task 5:

The user may have heard that prototype work has been done toward making it possible for the viewing of actual scanned images of logbook pages. That is, for a given run number, the relevant pages of the captain's logbook, the calorimeter logbook, etc. could be brought onto the screen, even if you were in Paris. (Actually, we have proven that this is possible — and even practical. The disk space required using DEC's DDIF format is reasonable, and the actual scanning can be done directly into a DEC-provided RdB database (DEC Image Express). John Featherly and I do not intend to press forward with this project, however, unless there is significant user interest). But....to see what a page would look like....

Action:

From the main menu, press 41 (Scanned Images, Calorimeter Logbook). If there is a display problem, keep in mind that a Set Display may be required. To remove an image from the screen, just 'click-and-drag'.

==================================

Task 6:

The user wants to step through all of the Run Summary information for all global runs in Run 1A. That is, by pressing an arrow key, the user wants to see information about the luminosities, beam xy position, detector quality, etc. for each run in turn. He or she may even want to print this information or write it to a file.

Action:

From the main menu, press 21 (For Run Summary Database Query). Then press the down(up) arrow to see each run record in sequence. Displayed will be the run number, luminosities, quality variable for the calorimeter, TRD, FDC, muon system, ..., the beam xy position, etc. Again, all of this information can be printed or written to a file using the procedures outlined above.

The user can also view the Run Summary file, by pressing the "Select" key while the Database Query form is being displayed.

====================================

Task 7:

The user wishes to do some heavy-duty querying of the run_summary parameters associated with the global runs from Run 1A. That is, he or she may want to develop a record stream of runs that satisfy some set of conditions on subdetector quality, luminosities, beam xy, etc.

Action:

There are actually two paths to consider here. If the query is a simple one based on a field in the run summary database having a definite value (i.e., a given run number, a given sub-detector quality, etc.), then one course of action is appropriate. If the conditions are complex (involving <,>, OR, AND, or cross field calculations) then another approach is required.

For the simple query — go to the main menu, select 22 (for RdB database query), press the “Do” key to get to the command line, type “Set Query” and press return. The form then clears, except for the field names. At this point the user types in the values he or she wishes to impose on the fields directly on the form next to the relevant fields, goes to the command line by pressing “Do”, and then types “Execute Query”. The new record stream selected now conforms to the restriction placed by the user (e.g., all runs for which the calorimeter was judged to be working perfectly). This selection of records can be scrolled, written to a file, or printed.

For the more complex query, again get into the query mode by pressing the “Do” key and typing “Set Query”. Then move the cursor to the field to which a condition is to be applied. Then press the “Do” key again and type “extend query”. Enter the conditions desired*, and press return. Then, finally, press the “Do” key and enter “Execute Query”. At this point the user will have selected a new record stream consiting only of records (runs) satisfying the conditions which he or she imposed. As in earlier examples, this record stream can be printed or written to a file for later use. Of course, it can also be scrolled interactively on the spot.

* (Example of condition: > 63000 AND <64000)

To return to the main menu, press <PF1>-q.

==================================

These examples are meant to show the capabilities of the package. It has other powers that we intend to keep in Jurassic Park, unless there are some who would like to be taken on a

Appendix II : Description of the Accelerator Luminosity Program

(provided by N. Amos)

The calculation of the integrated luminosity depends upon both the specific trigger conditions and the bunch crossing under consideration. A general calculation, which allows for both, is performed using scalers within the current Level 1 framework and information from the Accelerator Division.

A luminosity server was written which acquires data from the daq pool, from the Level 1 trigger control computer and from the Accel erator Division. This data is stored in a local data base and al lows for a full luminosity calculation without the need to spin tapes.

Corrections are made for experimental dead time, prescales, multiple interactions, main ring vetoing configurations and beam related Level 1 conditions. Further corrections for offline reconstruction efficiencies are performed. The D0 user is presented with a luminosity normalization based upon his selection of events per run, filter and reconstruction stream.

The integrated luminosity for a D0 run is actually measured through one of the Level 1 specific trigger bits (bit 30), which actually counts live luminosity. In this way, all sources of experimental live time are correctly accounted for crossing by crossing. The only correction to this is that due to multiple interactions. This correction is measured by another set of Level 1 scalers, and is applied separately for each of the six bunch crossings. Additional corrections for the measured Level 1 prescale factors and main ring vetoing configurations are likewise applied, and are based on Level 1 scalers readings. Level 2 prescales are applied based on the input prescale factors. Final corrections for offline reconstruction efficiencies are made from data in the production data base. Here, corrections for lost events or lost partitions are made based upon the number of events which were fully reconstructed.

The output of the luminosity data bases give the measured corrections for multiple interactions, central vertex cut, single interactions cut, main ring veto and the experimental prescale factors. Furthermore, the sources of dead time are broken down by source: front end busy, coor disable, level 2 disable, MRBS_LOSS or MICROBLANK. It is important to note that these corrections have already been applied when you request integrated luminosity by Level 1 trigger or by Level 2 filter. They are for your reference only and should not be reapplied by the user.

Appendix III: Production Database and Utilities
(E-MAIL Note from P. Bhat)

From : D0SFT::BHAT
Posted : 16-AUG-1993 10:06:18, Expires : 31-AUG-1993 10:02:52
Production Database and Utilities:

At the last OCPB meeting, the physics group representatives were asked some questions about theProduction Database and its usage. Note that it is a bit too late to ask what theProduction Database should contain and what it should be used for,in a broad sense. I had asked these questions one and a half years ago (see D0News dated 28 Jan. 92) when the databasewas being designed. Based on the answers and ideas that we came up with, we designed the database and have been storing a variety of information about raw and processed data. It has been and being used for certain applications like, for picking STA events (by Pick Event Facility), for getting luminosities for runs and files, periodicallyfor production- related information and statistics. Some time ago, we released some Production Database utilities which can be run by executing $D0SETUP PROD_DB. (Laura Paterno has written and released some Luminosity utilites.) Typing PDBHINT will give you a list of symbols and explanations for various utilities. These utilties will also beavailable via Homer Neal's DBLINKERinterface as well. At the present time, a word of caution however - we are preparing to make newreleases of these and many other new utilities. So, you will be better off if you can wait for a couple of weeks. Also, the database itself is being worked on in terms of making it complete. If you need something done immediately please contact me.

Recently, more emphasis and priorities have been given to this project by allocating more human resources. So, I would like users to explore and exploit the enormous amount of information that resides in this database. I provide here, some important items of information that are available in the database, which would enable and encourage physics groups /users tocontemplate using the database in a variety of ways.

The fundamental entities in this database are Runs and Files - raw and processed (reconstructed, streamed, filtered, re-reconstructed,...). (See D0Note #1266.), and Tapes.

For all runs written to tape, the following are available:

Run Number
Run-date (begin and end time)
Triggers and Filters for the run
- bit numbers and names
- pre-scales
Data origin (Collider, Calibration etc)
DAQ configuration filename
Accelerator Store Number

Instantaneous Luminosity (At begin and end of run)
Integrated Luminosity, delivered.
Integrated Luminosity from Min-bias triggers written to tape
Integrated Luminosity per Trigger (written to tape)
Various corrections for Luminosity
Number of Files/Run
Number of Events
Information about Files: (Raw and Processed)

Filename
FATMEN Generic Filename (via a utility)
First Event
Last Event
Number of Events
File Creation Date
Integrated Luminosity/file (Min-bias and Trigger based)
File_size (in Mb)
Tape Label
Sequence Number of file on tape
Production Process version number (for processed files)

There are separate records for RAW, STA,DST, streamed-STA, streamed-DST, filtered-STA, filtered-DST files. For processed files, a full mapping of which parent files were used to create the output files are kept. This helps provide luminosities for streamed/filtered DSTs.

Pushpa

APPENDIX IV: LISTING OF EVE/TPU PROGRAM (to open named file)

```
       SUBROUTINE EVE_OPEN(FILE_NAME)
C------------------------------------------------
C-
C-   Inputs   : FILE_NAME
C-   Outputs  : NONE
C-   Controls:
C-
C-   Created  31-JUL-1993   Homer A. Neal
C-
C------------------------------------------------
      IMPLICIT NONE
C------------------------------------------------

      INTEGER NLINE
      PARAMETER( NLINE = 13 )
      CHARACTER *80      A(NLINE)
      CHARACTER *(*)     FILE_NAME
      CHARACTER *(6)     STRING

      INTEGER  TPU$TPU
      INTEGER  LIB$SPAWN
      EXTERNAL TPU$TPU
      INTEGER  TPU$STATUS
      INTEGER  ISTATUS,I


      A(1) = ' LOCAL text_pattern, text_marker_range; '
      A(2) = ' text_pattern := "' //STRING //'";'
      A(3) =
     &  ' new_buffer := CREATE_BUFFER("temp_buf",
     &  "'//FILE_NAME(1:30)//'");'
      A(4) = ' SET (NO_WRITE,new_buffer);'
      A(5) = 'MAP(current_window,new_buffer);'
      A(6)=
     &  ' text_marker_range := SEARCH_QUIETLY(text_pattern,FORWARD);'
      A(7) = 'IF text_marker_range <> 0'
      A(8) = 'THEN'
      A(9) = '   POSITION (text_marker_range); '
      A(10) = 'ELSE'
      A(11)= 'MESSAGE ("Text not found" );'
      A(12)=
     & 'MESSAGE("You are in EVE -Please continue search or exit");'
      A(13)= 'ENDIF;'


       ISTATUS = LIB$SPAWN('$DELETE/LOG
     &  D0$BETA:[DBLINKER]EVE_PROG.TPU;*'  )


       OPEN (UNIT=10,
     &  FILE='D0$BETA:[DBLINKER]EVE_PROG.TPU', STATUS='NEW')

      WRITE (10,100)(A(I), I=1,NLINE)
      CLOSE (UNIT=10)
  100 FORMAT(1H ,A80)
C---

      ISTATUS=TPU$TPU('TPU/READ_ONLY/
     &   COMMAND=D0$BETA:[DBLINKER]EVE_PROG.TPU')


  999 RETURN
      END
```

Appendix V: LISTING OF ADL PROCEDURE TO BUILD RdB RUN SUMMARY DATABASE

_(author H. Neal)

```
=====================

PROCEDURE ADL_DB_MAKER;

 VAR
       ir       :                     NUMBER;
       rt       :                     CHAR (4);
       astart   :                     CHAR (23);
       astop    :                     CHAR (23);
       lum1     :                     NUMBER;
       lum2     :                     NUMBER;
       qual     :                     NUMBER;
       cqual    :         NUMBER;
       mqual    :                     NUMBER;
       cdcqual  :                     NUMBER;
       fdcqual  :                     NUMBER;
       trdqual  :                     NUMBER;
       vtxqual  :                     NUMBER;
       xbeam    :                     NUMBER;
       ybeam    :                     NUMBER;
       xystat   :                     NUMBER;
       rfile    :                     CHAR (30);
       gmfile   :                     CHAR (30);
       status   :                     NUMBER;
       sa1      :                     NUMBER;
       sa2      :                     NUMBER;
       sa3      :                     NUMBER;
           istop        :                  NUMBER;
           first_run    :                  NUMBER;
           stop_run     :                  NUMBER;
           loop         :                  NUMBER;
           id           :                  NUMBER;
           stat_code    :                  NUMBER;

BEGIN

        loop := -1;
        istop := 1;
        first_run := 63500;
        stop_run  := 64000;

                DB_OPEN(rs_dsd,id,stat_code);
                        IF (stat_code <>0) THEN
                                BEGIN
                                        ERROR(stat_code);
                                        RETURN;
                                END;

                WHILE (istop > 0) DO
                BEGIN

                  loop := loop + 1;
                  ir := first_run + loop;

    CALL RUN_SUM_HAN_extlnk1 (
                     ir,rt,astart,astop,
                     lum1,lum2,qual,cqual,mqual,cdcqual,
                     fdcqual,trdqual,vtxqual,xbeam,ybeam,
                     xystat,rfile,gmfile,status,sa1,sa2,sa3);

    IF ((status = -10) OR (ir > stop_run) ) THEN istop := 0;

        IF (istop > 0 ) THEN

            BEGIN

                        rs_dsd.I1               :=      ir;
                        rs_dsd.rt               :=      rt;
                        rs_dsd.start            :=      astart;
                        rs_dsd.stop             :=      astop;
                        rs_dsd.lum1             :=      lum1;
                        rs_dsd.lum2             :=      lum2;
                        rs_dsd.qual             :=      qual;
                        rs_dsd.cqual            :=      cqual;
                        rs_dsd.mqual            :=      mqual;
                        rs_dsd.cdcqual          :=      cdcqual;
                        rs_dsd.fdcqual          :=      fdcqual;
                        rs_dsd.trdqual          :=      trdqual;
                        rs_dsd.vtxqual          :=      vtxqual;
                        rs_dsd.xbeam            :=      xbeam;
                        rs_dsd.ybeam            :=      ybeam;
                        rs_dsd.xystat           :=      xystat;
                        rs_dsd.rfile            :=      rfile;
                        rs_dsd.gfile            :=      gmfile;
                        rs_dsd.status           :=      status;
                        rs_dsd.sa1              :=      sa1;
                        rs_dsd.sa2              :=      sa2;
                        rs_dsd.sa3              :=      sa3;


            IF( (rt = 'GLB') AND (status=0)) THEN

                     BEGIN

                              DB_INSERT (id,stat_code);

                                   IF( stat_code <> 0 ) THEN
                                       BEGIN
                                       ERROR (stat_code);
                                       RETURN;
                                   END;

                     END;
                END;                                   {jumps here on last event}


                   END;                                {ends WHILE DO}

                DB_COMMIT (stat_code);

                IF (stat_code <> 0) THEN

                     BEGIN
                      ERROR (stat_code);
                      RETURN;
                     END;

                DB_CLOSE (id, stat_code);

                IF(stat_code <> 0 ) THEN
                      ERROR (stat_code);


END;
```

Appendix VI: Preliminary Proposal for Global Monitor Summary Definition

```
1790
   #1790        2-DEC-1993 16:31:53.93                                  93B
From:D0SFT::KLIMA          "Boaz Klima (708)840-2323"
To:@GM_ALL,@CONVENORS
CC:KLIMA
Subj:Proposal for Global Monitor Summary Definition
```

Fermilab, 2-DEC-1993

Dear GM and/or physics group convenor,

The Global Monitor Summary (GM_SUM) combined with the Rally-based combined database can be used by D0 data analyzers to determine possible bad runs for Run 1b analyses. We would like to ensure that the decisions made by the Global Monitors (GM's) in filling out the summary are most useful.

I formed an ad-hoc committee led by Jeff Bantly to discuss the way the GM_SUM tool should be used by the GM's. The committee included the following people: Jeff Bantly (chair), Krzysztof Genser, Norman Graf, Jae Yu and Daria Zieminska. They discussed this question both representing the analysis (users) needs and from the GM's (providers) point of view. They came up with the following document as a proposal to accomplish this goal.

Please read it and send comments to us as soon as possible, December 10th at the latest. Comments are solicited both from GM's and from physics group convenors.

Boaz

Proposal for Gm Summary Program Data Entry
==

The Global Monitor Run Summary program is a tool designed to record problems or lack thereof for the global and physics data runs taken at D0. Information from the summary files is accessed by the Rally-based utility from Homer Neal's group. Users are able to select good or bad runs through selections made on the information in the database. Towards this end, the proposal presented below is an attempt to provide guidelines to the Global Monitors on shift so that their input into the summary is uniform for all shifters for the entire run.

The GM Summary Program has several generic entry items such as run number, date stamps, and the names of the GMs inputting information. For each detector, a rating box allows the GM to mark the detector as 'good'(green), 'maybe' (yellow), 'bad'(red), or 'no information'(no color). There is also a text box available for the shifter to describe any problems that would explain a non-good rating. The current detector list includes VTX, TRD, CDC, CAL, MUO, and FDC. There are also rating boxes and text spaces for the Level 0, Level 1, Level 2, Global Examine, and Expressline results. Finally, a Global rating box with text is available for an overall run rating. This combination of entries allows for a detailed representation of each run using all of the tools available to the GM including information available from the other individuals on shift.

The global decision between good, maybe, and bad should be made on the following basis. A good run is one that can be used by any physics group. There are no known problems with the run. A bad run is one that can be used by no physics group. Problems with the run would be deliniated in the text boxes. A maybe run is one in which some physics groups would be able to use the run. An initial search for bad runs might include all bad runs and use a series of cuts on individual detectors to decide the maybe runs.

The detector decision between good, maybe, and bad should be made based on the functionality of the detector throughout the run. A good mark indicates the detector was fully functionable for the entire run. This would include paused run HV resets. A bad mark indicates the detector was inoperative or defective for a significant part of the run (>10%). A maybe mark indicates an uncertainty of a problem. The detector output does not match the expected results but no problem with the detector is known at the time.

The triggering decisions are more individualized. Level 0 bad indicates the detector is not functioning (no way to measure luminosity). Level 0 maybe indicates a few channels are not functioning (introducing an increased luminosity error for the run). Level 1 and Level 2 bad indicate wrong downloads, wrong trigger menus, nodes are processing events improperly, etc. A maybe indicates an exceptionally large or small trigger rate that is not explainable.

The data analysis decisions are based on output histograms. A bad Global Examine indicates a bad condition in a detector was seen. A maybe would indicate either the program bombed in the analysis or a problem is seen only by the Global Examine and there is no evidence from other sources. The examine analyzes a sample of events from the Global Shared Common and therefore picks up the higher rate triggers. The text would explain the problems seen. The GM Expressline indications would be the same. The expressline examines several of the low rate triggers and a greater quantity large-sized events.

The overall strategy for decision-making is given above and, to aid the GM further, a review of the major bad run lists of the experiment has been done. The various reasons listed in the two known lists are all categorized below.

Detector/System Bad:
- BLS problems, bad octant
- detector readout problems, HV set wrong, unnoticed HV trip
- detector missing
- wrong trigger menu downloads, bad downloads, wrong RCPs, DAQ problems
- testing prescales, modifying triggers, events not passing triggers
- magnet off
- test run not meant to be recorded, <.02 nb-1 recorded
- bad beam, scraping not complete, very high beam losses

Detector/System Maybe:
    hot cell problems, few channels not functioning properly
    magnet current fluctuating

Oddly enough, this is the entire list of problems cited for bad runs by the varioius physics groups. It is not very long but encompasses many possibilities. Not all of these reasons are sufficient to mark a run as bad but all would mark at least one system as bad or at least a maybe.

The following are suggestions made by us to improve the GM Summary program. First, the muon system should be split from MUO into WMU and SMU. Some physics groups do not care if SAMUS is functioning whereas others require it. Second, one input should be made to indicate that the run is a special physics run, BMUONS and GAMMA for example. Anything that does not use the standard trigger menus currently in use should be marked as a special run.

Two further problems arise that must be addressed. First, very short runs where almost no detector or examine information is available are difficult to assess. If the reason for the short run is Tevatron beam abort or some non-physics affecting ending of the run, it shall be marked similarly to the previous run. However, if the run was started after a change in triggers or a fix to detectors or was ended prematurely for some problem then the run will be marked as bad. Appropriate explanations in the text area should be provided. Second, sometimes a good or maybe run will be found out later to have been bad. If the GM summary files have already been migrated to the Rally database, the update of the information will occur there as opposed to changing the original GM summary database. Otherwise, the change in status of the run might be lost.

Proper use of the GM Summary database via the Rally package will enable a future physics analysis to generate a bad run list in a simple manner. If there are any doubts as to how to mark a run, the guidelines listed above should solve most problems. If there is still doubt, the run should be marked pessimistically and reported to the experts (DOSFT::KLIMA,FNALDO::BANTLY) along with sufficient details.