A Diagnostic Tool for Console Applications of the Fermilab Accelerator Control System

Junye Wang, Brian Hendricks
Fermilab
Beam Division Accelerator Controls Department
MS 347 P.O. Box 500
Batavia, IL 60510 USA

Abstract

The Fermilab accelerator console software consists of console manager programs, console applications, and a library that has more than one thousand entry points which is named CLIB. CLIB provides support for data acquisition, user interface, error handling, memory management, database access, file access, network access, and many other things.

The CLIB Peeker is a network capable diagnostic tool that allows programmers to view information about CLIB routines in a running application. CLIB Peeker runs as an application program. It can be used to inspect any application program running on any active console, and it can be set up to automatically update any diagnostic information. CLIB Peeker is an efficient tool that helps users debug and improve programs.

1 The purpose of clib peeker

The CLIB (Console LIBrary) of the accelerator controls system at Fermilab includes more than one thousand routines for application programmers. Such a wealth of CLIB routines is very helpful and convenient for users, but at the same time makes it hard for users to develop a mastery of necessary routines. CLIB Peeker can be used to find syntactically correct but improper usage of these routines.

2 CLIB peeker user interface

Users can peek at programs on any console since the information is obtained via network (ACNET) requests. To select which program to peek into, interrupt on the field 'CNSxxx Slot x' (fig.1). A node selection menu pops up and lets the user select any console or central node. After choosing a console or node, a menu of currently running processes will pop up, and the user can choose the process that is to be viewed.

Users can choose the information update rate by typing a number in the brackets of the '< 0> one shot' field (fig. 1). The value the user enters will be interpreted as the number of seconds between peeks. A value of zero can be used for a one shot peek.

The CLIB facility selection can be made via the "Pgm_Tools" menu (fig.1). There are presently fourteen of these facilities supported by CLIB Peeker. These facilities include 'Process information', 'Memory management', 'Data acquisition', 'Window management', 'Graphic window management', 'Error handling', 'Database access', 'Network access', 'File management', and more.

General statistics information such as current status, last status, number of function calls, etc. are available for all facilities. A menu bar controls whether these general statistics information windows are displayed or not.

You can click on the 'Scaled' button (fig. 1) to choose between "Raw" and "Scaled" data displays. In "Raw" data display mode, all information is displayed in numerical digits except text strings. For example, time will be displayed in clinks (the number of seconds since January 1, 1972), logical values will be either '0' or '1', and devices will be displayed by device index. If the user selects 'Scaled' data mode, all information is displayed in text strings except function call counts. For example, time will be displayed as 'dd-mmm-yyyy hh:mm:ss' (date string format), and logical values will be either 'False' or 'True', and devices will be displayed by name.

Most sub-peekers have their own specific diagnostic information commands. Users can access these special commands by clicking on the 'Command' button. Most of the windows are movable and resizable. Therefore users may configure the interface to their needs.

3 Diagnostic information support

A set of inter-program communication routines supports the CLIB peeking activities. CLIB Peeker sends a request, which consists of the user selected peek node, peek task, and a typecode which has been assigned to the user-selected facility, to the CLIB facility via ACNET (Accelerator Control NETwork). Then a reply returns through ACNET with the statistics embedded in the CLIB facility. In general, the returned information includes numbers of function calls, errors and status, and other miscellaneous information.

4 The peekers

CLIB Peeker allows users to peek into "Data Acquisition", "Database Access", "Screen Management", "Graphic Window Management", "Error Handling/Logging", "File Access", "Memory Management", "Network Access", "Process Information" and many more.

"DIO Peeker" provides access to data acquisition activity. Data acquisition involves input and output to accelerator hardware as well as reading database information concerning that hardware. It involves reading, setting, controlling, and scaling values as well as handling alarms and miscellaneous device attributes. The "DIO Peeker" will display counts of function calls and errors. It will also show the number of devices and lists of devices being read and set as well as the frequency of retrievals. In addition, timing statistics are shown for the data accesses being performed. "DIO Peeker" will be introduced as an example later.

"Data Pool Peeker" and "Database Peeker" look at low level data acquisition routines. These two peekers display the numbers of data base manager and data pool manager function calls used in the peeked application. They also display the errors of the last functions called.

"Window Manager Peeker" shows each window manager function call counts and errors. It also displays the attributes of existing windows, switch information, slider information, menu bar information, select field information, and help field information.

"Graphic Window Manager Peeker" is similar to the "Window Manager Peeker". It displays the information of all graphic windows built in the inspected application program.

"Error Handling/Logging Peeker" peeks at programs that use error handling and logging routines. Error handling routines check error codes, expand them to give text descriptions, and save errors encountered during execution of the program. Error messages can be written to a shared circular log file. This peeker will display the Error Handling/Logging routine call counts and many miscellaneous pieces of information.

"File Peeker" can peek at applications that use shared read/write access files. It displays numbers of function calls and errors. It also displays which file and record was last read and last written and any error associated with the access. Additionally, timing information is displayed for communications with the central file server process.

"Memory Peeker" allows you to inspect dynamic memory problems. It shows how much dynamic memory has been allocated by the program, how many memory calls have been made, whether or not memory blocks have been overwritten, etc.

"ACNET API Peeker" peeks at programs that use interprocess communication routines. This peeker will display the inter-process communication function call counts, and also show the current ACNET handle, requests, and errors.

"Process Information Peeker" will show you the inspected process' name, user name, process start up time, process ID, CPU time, memory information, date of CLIB image, type of CLIB in use, and more.

5 Example "DIO peeker"

There are hundreds of data acquisition routines in CLIB. More than half of the applications use DIO routines. "DIO Peeker" inspects accelerator device I/O problems. Click on the field 'CNSxxx Slot x' (fig. 1) to select the target application. DIO Peeker supports a lot of diagnostic information that will be categorized and displayed in four windows: 'Device Counts', 'Error Counts', 'Function Calls', and 'Status'.

The 'Device Counts' window shows users the count of: individual retrieval devices, retrieval devices in lists, individual setting devices, setting devices in lists, snapshot devices being read, retrieval lists, setting lists, table device lists, scaling information (PDB-Process Data Block) entries, devices read on event or at various frequencies, entries in the DI (Device Index) pool, settings attempted, and setting errors.

The 'Error Counts' window displays the error counts while running the application. These errors are generated by the application calling DIO routines, DBM (Data Base Manager) routines, DPM (Data Pool Manager) routines, or ACNET (Accelerator Control NETwork) routines. In addition, the number of DPM pending, database timeouts, and memory free errors are also displayed in the 'Error Counts' window.

The 'Function Calls' window displays the number of calls to the various data acquisition functions.

The 'Status' window displays which device and which property of the device were last read and last set and any errors associated with the access. Also, it shows timing statistics (in 15 Hz ticks) for the data accesses being performed, such as the longest DBPROC (the process of transmitting the database request list to the central device database server), the total time spent in DBPROCs, the longest DPPROC (the process of making new requests to and deletions from the data pool), and the total time spent in DPPROCs. The timeout for reading from or setting to the data pool and the timeout for reading from the database are also displayed in this window. Additionally, the error message logging state (on or off), the device setting flag state (enabled or disabled), the stale data notify flag (True or False), and the MIO use flag (using DIO routines to read and set from memory rather than the actual devices) are shown in this window.

If you click on the 'Command' button, a command menu pops up which allows you to read specific diagnostics or set various parameters. There are presently 10 commands in the menu:

1) 'Clear Statistics': clear DIO diagnostic values
2) 'Setting enable': enable device settings
3) 'Setting disable': disable device settings
4) 'Delete Cache': delete all entries from DIO's memory cache of central database information
5) 'Toggle Logging': enable or disable the logging of low level error messages
6) 'Set Timeout': set timeout for reading information, setting information, and accessing database information.
7) 'Enable Stale Notify': enable stale reading notification
8) 'Disable Stale Notify': disable stale reading notifica-

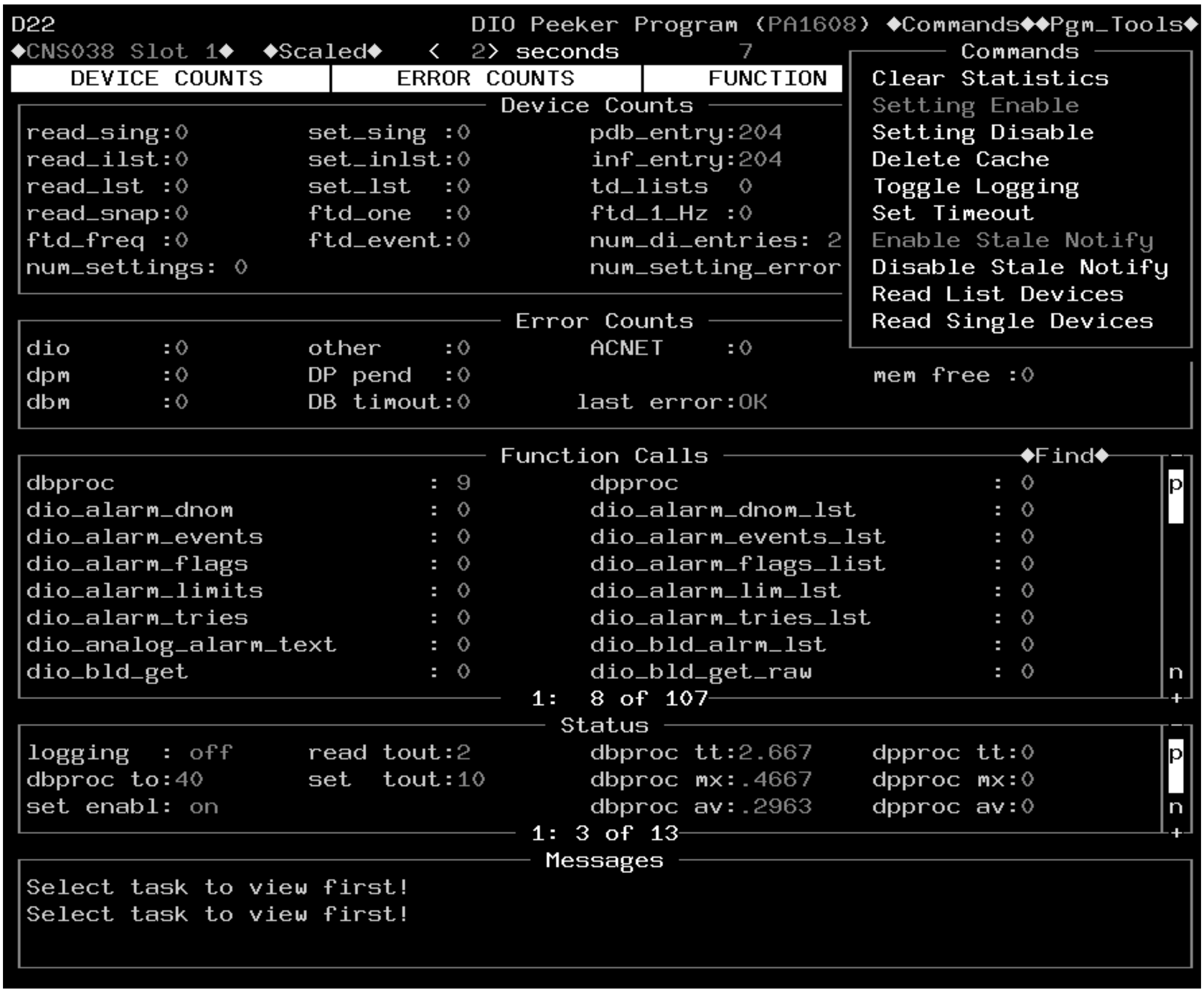
tion

9) 'Read List Devices': return the list IDs and number of devices for device reading or setting lists. If you click on a list ID, information about all of the devices in the list will be displayed including device name, device property, FTD, data size, and data offset.

10) 'Read Single Devices': returns information about individual devices that have been read or set in the inspected application. The information returned includes device name, device property, FTD, data size, and data offset.

6 Conclusions

The CLIB Peeker system continues to grow as the functionality of CLIB does. New facilities and new information for already supported facilities are continually being added. This system has repeatedly been used to investigate difficult problems during application program development as well as problems resulting from errors occurring elsewhere in the control system. The network capable nature of this system has been especially useful when programmers call control system personnel for assistance in debugging applications running in their offices.



```
D22                            DIO Peeker Program (PA1608) ◆Commands◆◆Pgm_Tools◆
◆CNS038 Slot 1◆  ◆Scaled◆   <  2> seconds        7  ┌─────── Commands ────────┐
   DEVICE COUNTS   │      ERROR COUNTS   │    FUNCTION │ Clear Statistics        │
┌──────────────────────────── Device Counts ─────────│ Setting Enable          │
│read_sing:0         set_sing :0        pdb_entry:204 │ Setting Disable         │
│read_ilst:0         set_inlst:0        inf_entry:204 │ Delete Cache            │
│read_lst :0         set_lst  :0        td_lists  0   │ Toggle Logging          │
│read_snap:0         ftd_one  :0        ftd_1_Hz :0   │ Set Timeout             │
│ftd_freq :0         ftd_event:0        num_di_entries: 2 Enable Stale Notify     │
│num_settings: 0                        num_setting_error Disable Stale Notify   │
└─────────────────────────────────────────────────────│ Read List Devices       │
┌────────────────────────────── Error Counts ─────────│ Read Single Devices     │
│dio       :0        other    :0        ACNET    :0   └─────────────────────────┘
│dpm       :0        DP pend  :0                          mem free :0         │
│dbm       :0        DB timout:0       last error:OK                           │
└──────────────────────────────────────────────────────────────────────────────┘
┌──────────────────────────── Function Calls ─────────────────────◆Find◆───┬─┐
│dbproc                     : 9        dpproc                      : 0      │p│
│dio_alarm_dnom             : 0        dio_alarm_dnom_lst          : 0      │ │
│dio_alarm_events           : 0        dio_alarm_events_lst        : 0      │ │
│dio_alarm_flags            : 0        dio_alarm_flags_list        : 0      │ │
│dio_alarm_limits           : 0        dio_alarm_lim_lst           : 0      │ │
│dio_alarm_tries            : 0        dio_alarm_tries_lst         : 0      │ │
│dio_analog_alarm_text      : 0        dio_bld_alrm_lst            : 0      │ │
│dio_bld_get                : 0        dio_bld_get_raw             : 0      │n│
└──────────────────────────────── 1:  8 of 107─────────────────────────────┴+┘
┌─────────────────────────────────── Status ───────────────────────────────┬─┐
│logging  : off      read tout:2        dbproc tt:2.667    dpproc tt:0      │p│
│dbproc to:40        set  tout:10       dbproc mx:.4667    dpproc mx:0      │ │
│set enabl: on                          dbproc av:.2963    dpproc av:0      │n│
└────────────────────────────────── 1: 3 of 13──────────────────────────────┴+┘
┌───────────────────────────────── Messages ───────────────────────────────────┐
│Select task to view first!                                                     │
│Select task to view first!                                                     │
│                                                                               │
└──────────────────────────────────────────────────────────────────────────────┘
```

Figure 1